Exploratory Data Analysis(EDA) with PySpark on Google Playstore

-- Akshitha Kukudala

Introduction

This project is a journey of analyzing various apps found on the Google play store with the help of PySpark.

Motivation

- Being an everyday phone user, it is interesting to take the real time application information and drive the insights based on installations.

Why Android when most of us are iPhone users?

Smartphones running the **Android operating system hold an 87 percent share of the global market in 2019** and this is expected to increase over the forthcoming years. The mobile operating system developed by Apple (iOS) has a 13 percent share of the market.

Data Content

Dataset: Dataset is downloaded from Kaggle
It consists of 28 columns as mentioned below and 450796 rows.

App Name, App Id, Category, Rating, Rating Count, Installs, Minimum Installs, Free, Price, Currency, Size, Minimum Android, Developer Id, Developer Website Developer Email, Released, Last update, Privacy Policy, Content Rating, Ad Supported, In app purchases, Editor Choice, Summary, Reviews, Android version Text, Developer, Developer Address, Developer Internal ID, Version

Data Preparation

- Multiline records were handled.
- Unwanted columns are dropped.
- Removed unwanted data and casted the columns.
- Removed unwanted special characters from the required columns to compare and analyze.
- Converted the required records to Pandas to plot.

Prerequisites

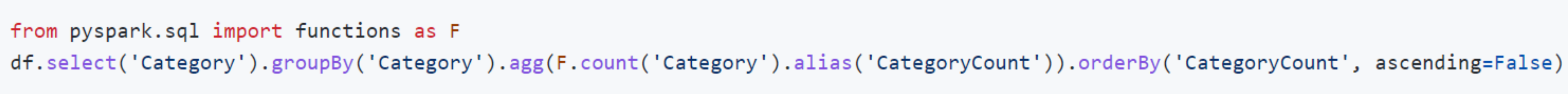
- Dataset from Kaggle.
- Google colab or Jupyter notebook.

```
Install pyspark  #project is based on pyspark
import pandas as pd #converted pyspark df to pandas df
import plotly.express as px #For plotting
import plotly #Used plotly templates for px charts
from pyspark.sql import functions as F
import matplotlib.pyplot as plt #generated stacked chart
```

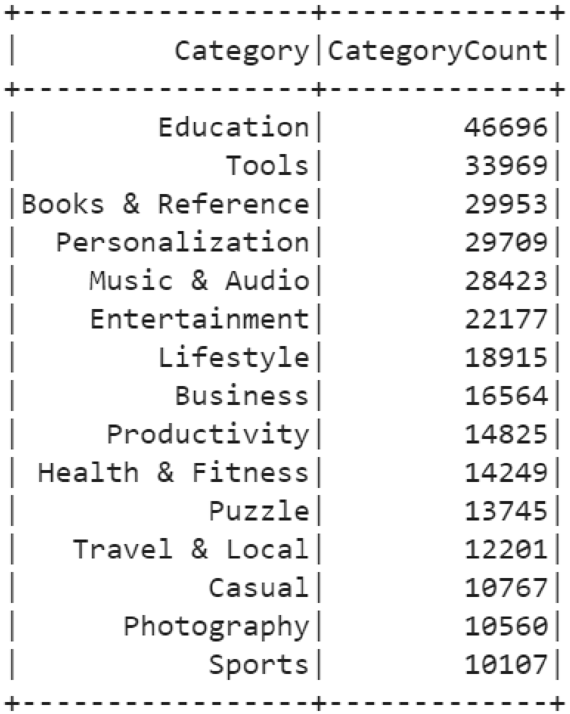
Goal with EDA

- Top categories in the play store?
- Free Vs Paid Apps in Each Category?
- Free Vs Paid Apps?
- Distribution of Ratings?
- Top Apps which has Installations greater than a Billion?
- Top Apps which has highest number of Reviews?

Top categories in the play store?

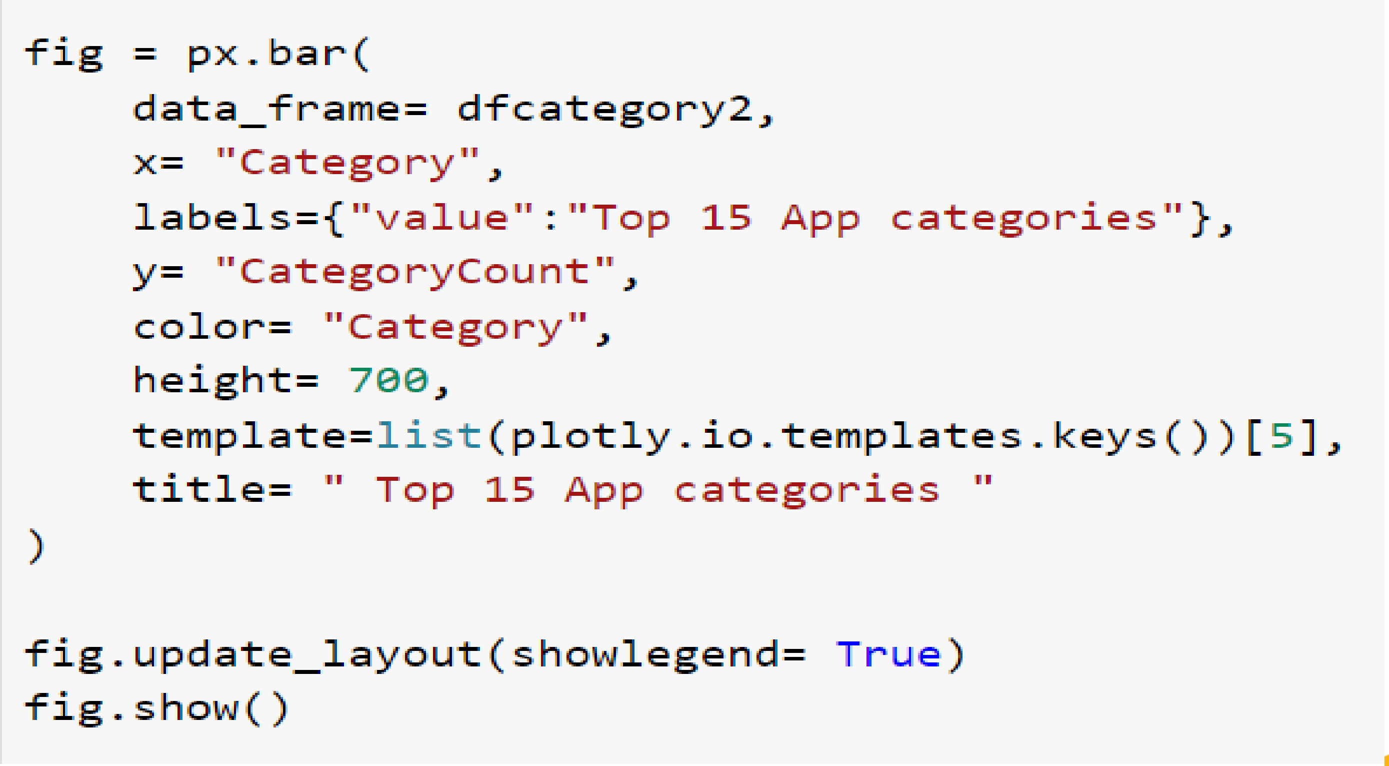


```
from pyspark.sql import functions as F
df.select('Category').groupBy('Category').agg(F.count('Category').alias('CategoryCount')).orderBy('CategoryCount', ascending=False)
```

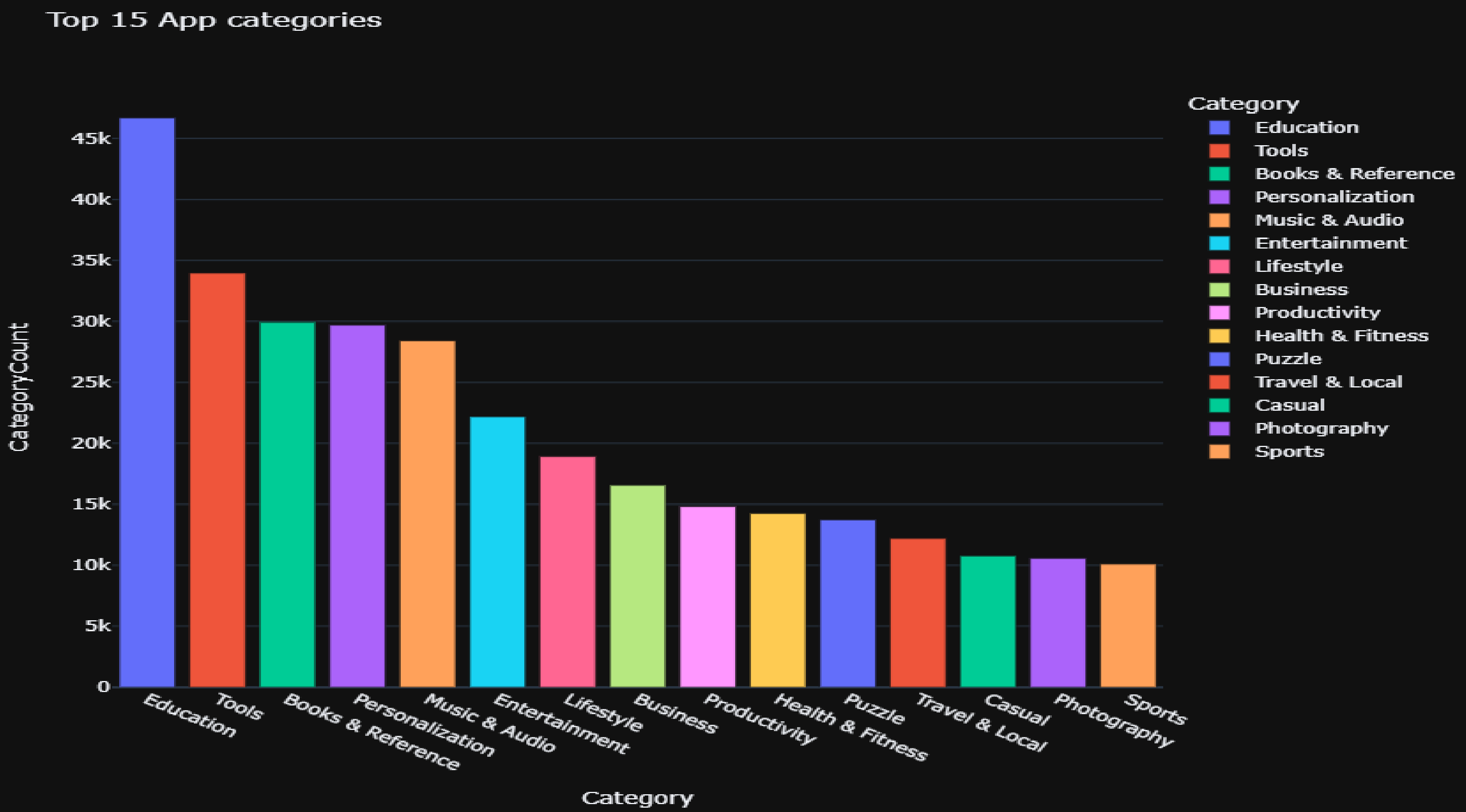


```
+----------------+-------------+
|        Category|CategoryCount|
+----------------+-------------+
|       Education|        46696|
|           Tools|        33969|
|Books & Reference|        29953|
|  Personalization|        29709|
|    Music & Audio|        28423|
|    Entertainment|        22177|
|        Lifestyle|        18915|
|         Business|        16564|
|     Productivity|        14825|
|  Health & Fitness|        14249|
|           Puzzle|        13745|
|    Travel & Local|        12201|
|           Casual|        10767|
|      Photography|        10560|
|           Sports|        10107|
+----------------+-------------+
```

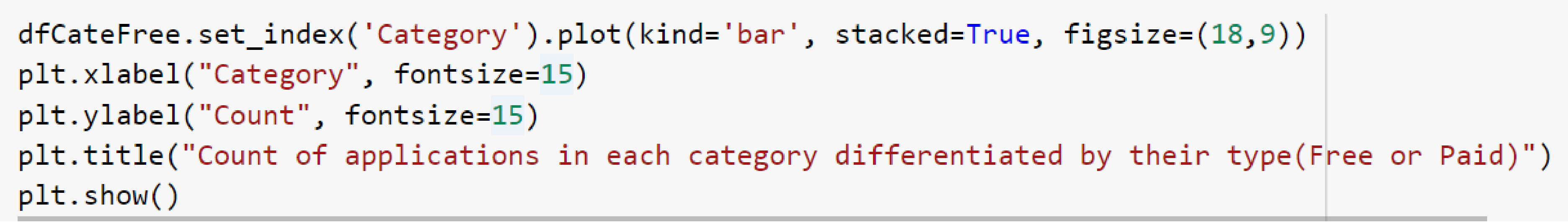
Plotly Express library



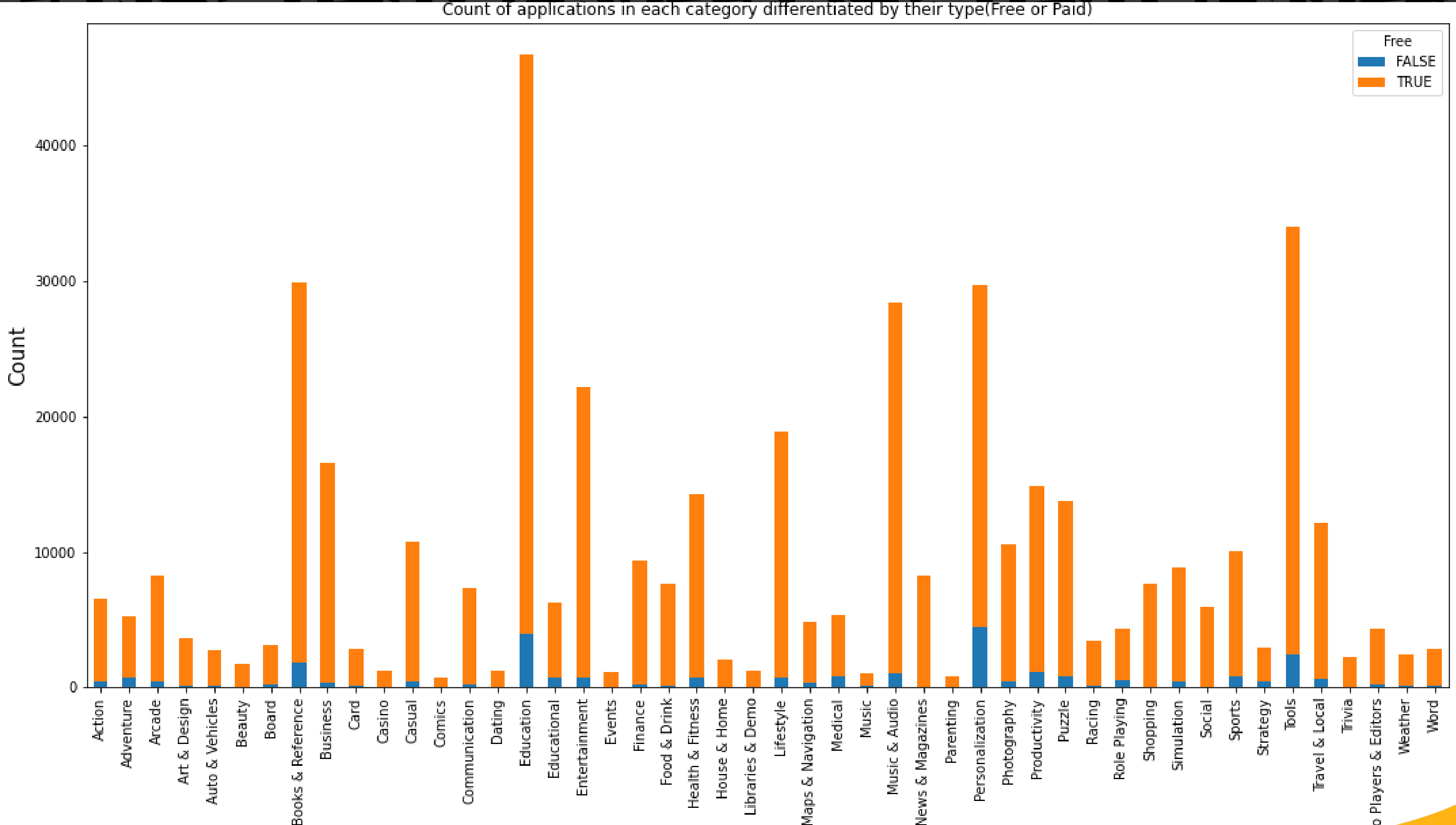
```
fig = px.bar(
    data_frame= dfcategory2,
    x= "Category",
    labels={"value":"Top 15 App categories"},
    y= "CategoryCount",
    color= "Category",
    height= 700,
    template=list(plotly.io.templates.keys())[5],
    title= " Top 15 App categories "
)

fig.update_layout(showlegend= True)
fig.show()
```


Top 15 App categories
CategoryCount
0
5k
10k
15k
20k
25k
30k
35k
40k
45k
Education
Tools
Books & Reference
Personalization
Music & Audio
Entertainment
Lifestyle
Business
Productivity
Health & Fitness
Puzzle
Travel & Local
Casual
Photography
Sports
Category
Category
Education
Tools
Books & Reference
Personalization
Music & Audio
Entertainment
Lifestyle
Business
Productivity
Health & Fitness
Puzzle
Travel & Local
Casual
Photography
Sports

Free Vs Paid Apps in Each Category

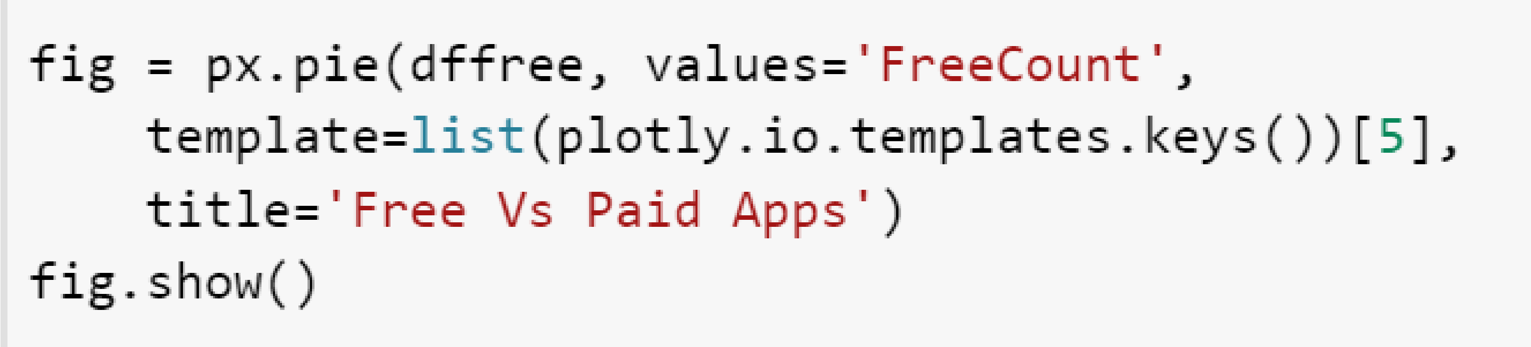


```
dfCateFree.set_index('Category').plot(kind='bar', stacked=True, figsize=(18,9))
plt.xlabel("Category", fontsize=15)
plt.ylabel("Count", fontsize=15)
plt.title("Count of applications in each category differentiated by their type(Free or Paid)")
plt.show()
```

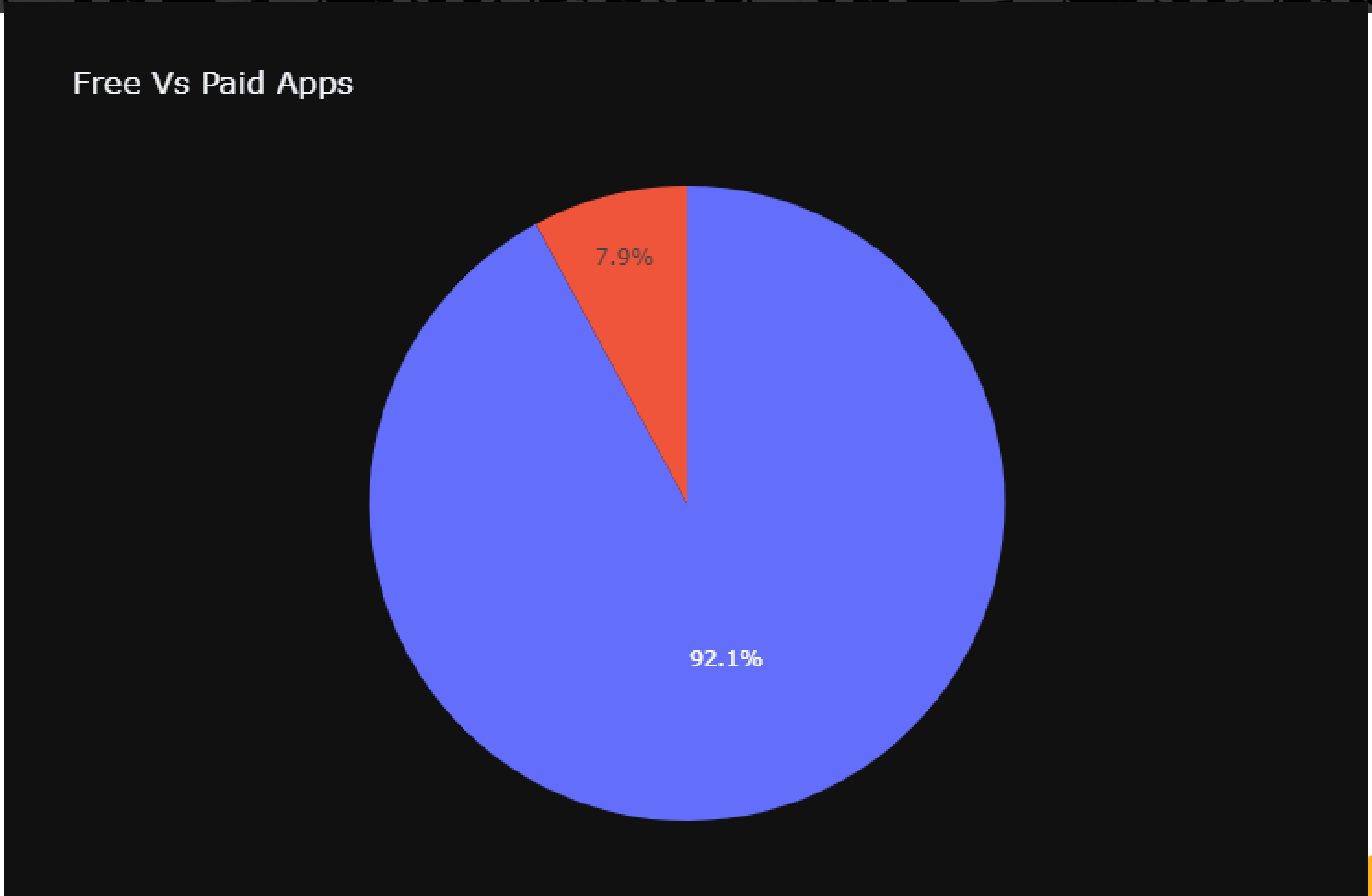


Count of applications in each category differentiated by their type(Free or Paid)

Free Vs Paid Apps

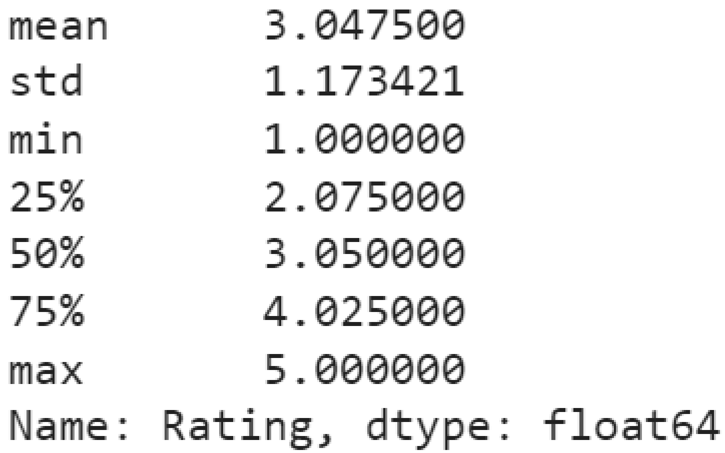


```
fig = px.pie(dffree, values='FreeCount',
    template=list(plotly.io.templates.keys())[5],
    title='Free Vs Paid Apps')
fig.show()
```

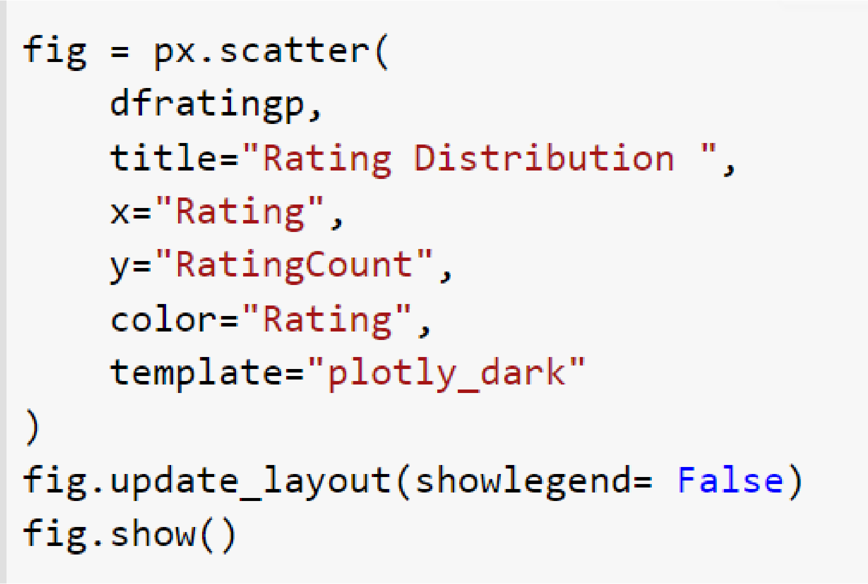


Free Vs Paid Apps

Distribution of Rating?

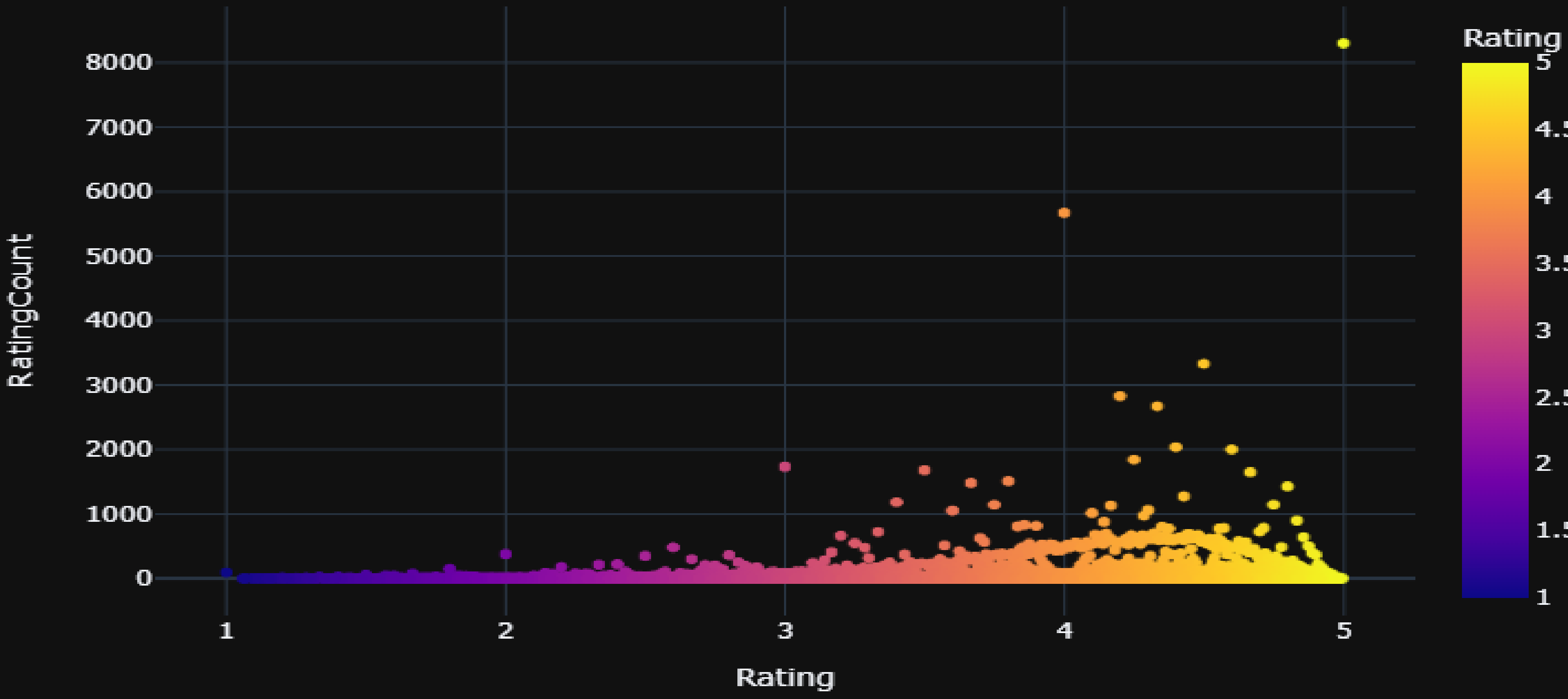


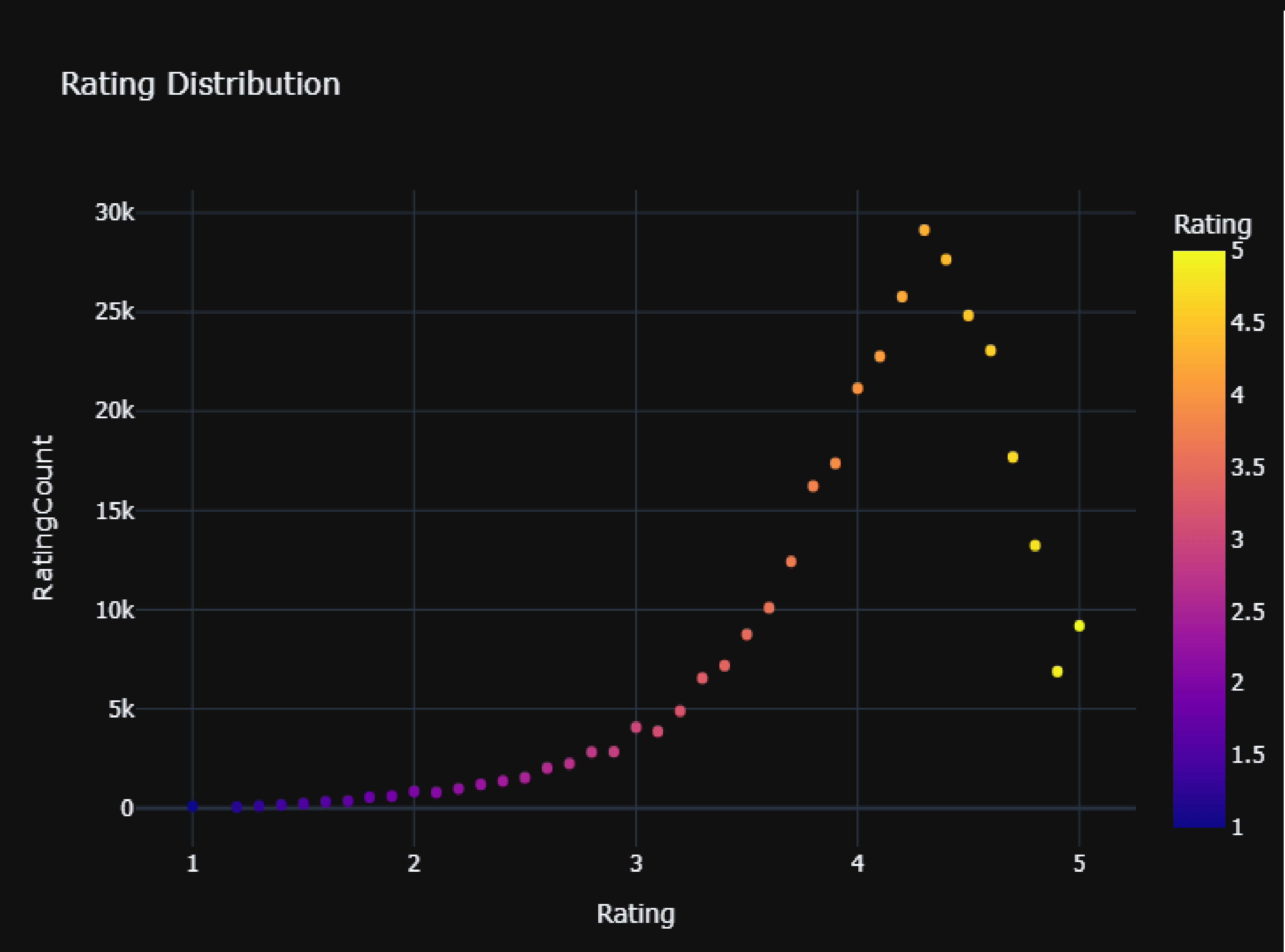
```
mean       3.047500
std        1.173421
min        1.000000
25%        2.075000
50%        3.050000
75%        4.025000
max        5.000000
Name: Rating, dtype: float64
```

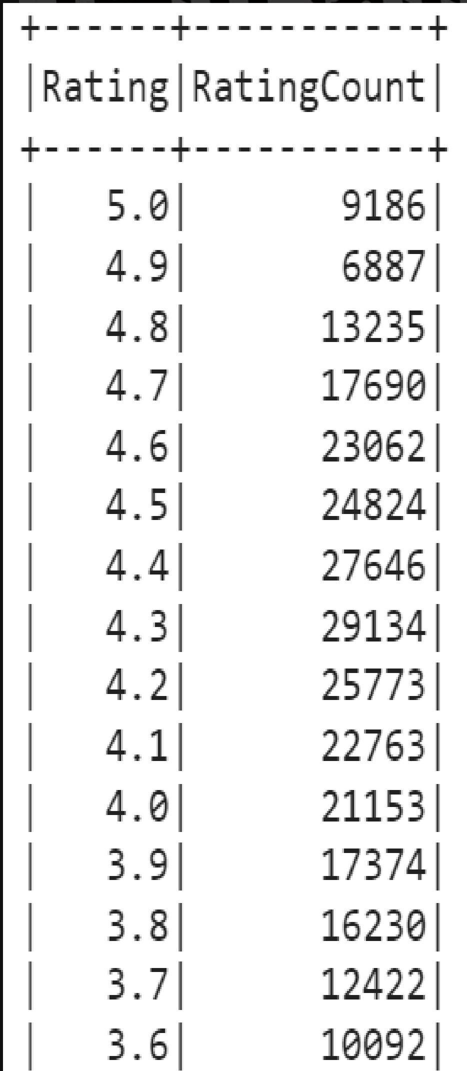


```
fig = px.scatter(
    dfratingp,
    title="Rating Distribution ",
    x="Rating",
    y="RatingCount",
    color="Rating",
    template="plotly_dark"
)
fig.update_layout(showlegend= False)
fig.show()
```

Rating Distribution

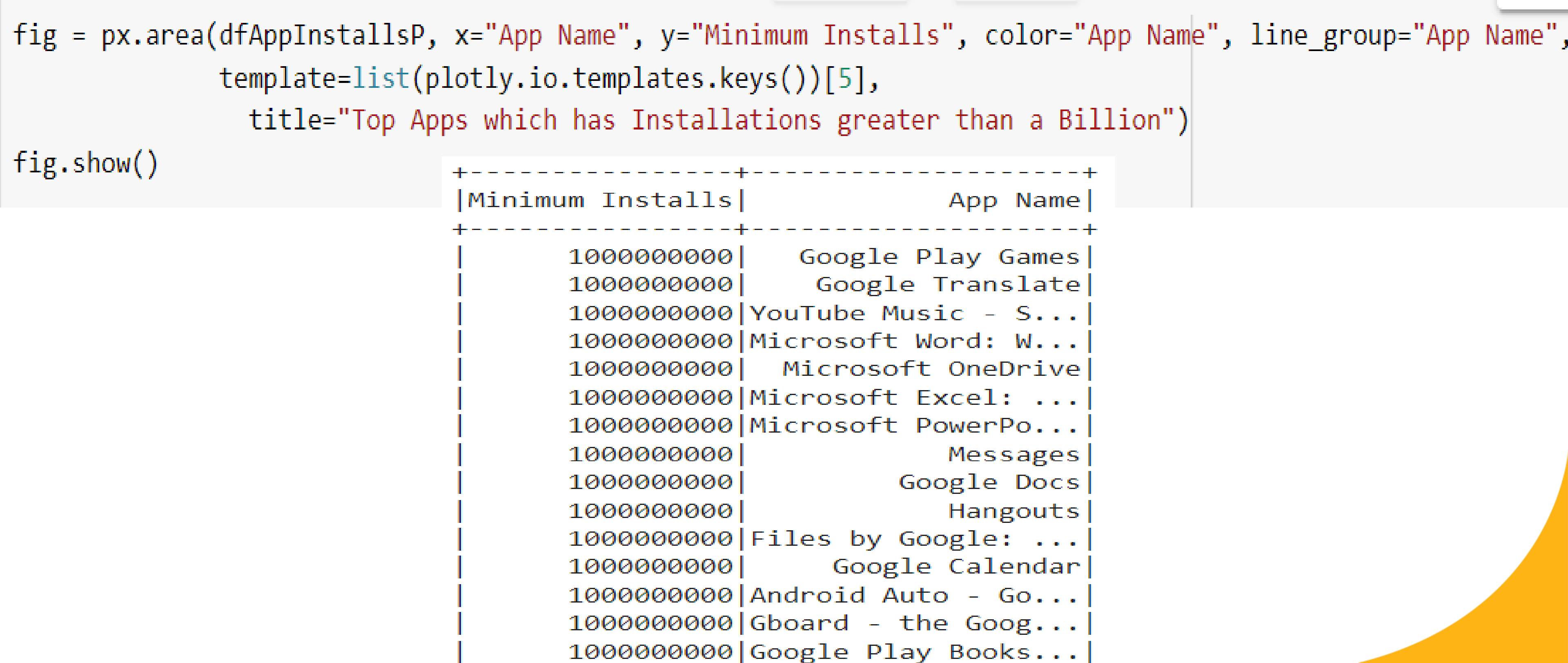






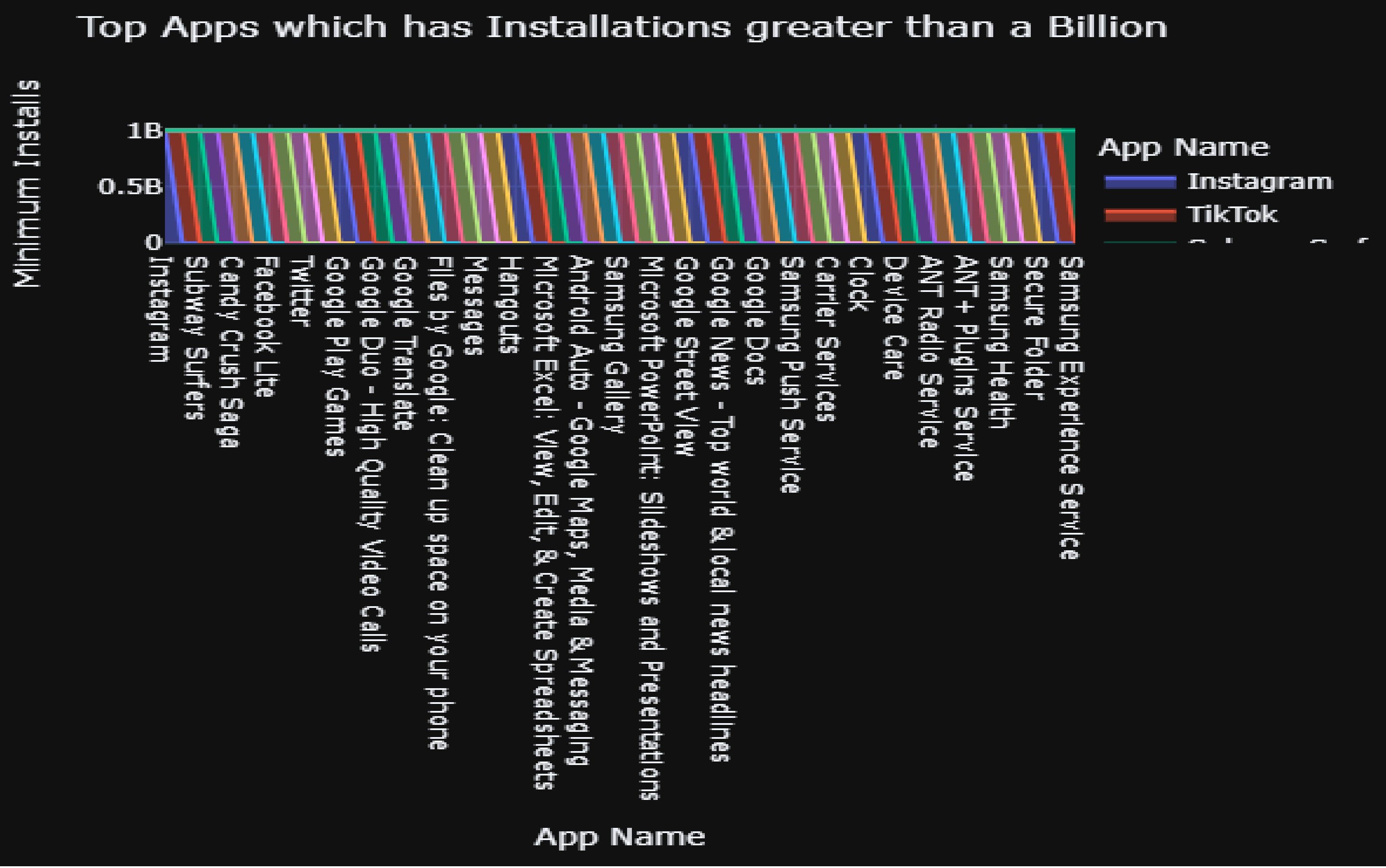
```
+------+-----------+
|Rating|RatingCount|
+------+-----------+
|   5.0|       9186|
|   4.9|       6887|
|   4.8|      13235|
|   4.7|      17690|
|   4.6|      23062|
|   4.5|      24824|
|   4.4|      27646|
|   4.3|      29134|
|   4.2|      25773|
|   4.1|      22763|
|   4.0|      21153|
|   3.9|      17374|
|   3.8|      16230|
|   3.7|      12422|
|   3.6|      10092|
```

Top Apps which has Installations greater than a Billion?



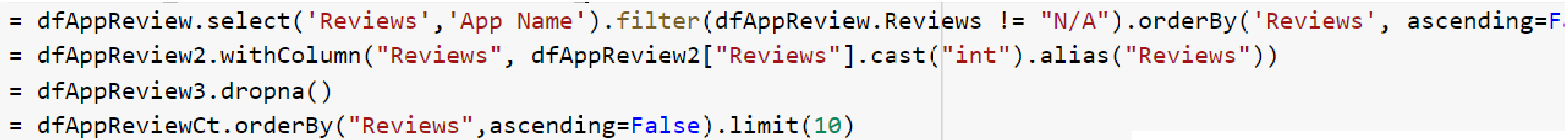
```
fig = px.area(dfAppInstallsP, x="App Name", y="Minimum Installs", color="App Name", line_group="App Name",
              template=list(plotly.io.templates.keys())[5],
                title="Top Apps which has Installations greater than a Billion")
fig.show()
```

```
+----------------+----------------------+
|Minimum Installs|              App Name|
+----------------+----------------------+
|      1000000000|     Google Play Games|
|      1000000000|      Google Translate|
|      1000000000|YouTube Music - S...|
|      1000000000|Microsoft Word: W...|
|      1000000000|    Microsoft OneDrive|
|      1000000000|Microsoft Excel: ...|
|      1000000000|Microsoft PowerPo...|
|      1000000000|              Messages|
|      1000000000|           Google Docs|
|      1000000000|              Hangouts|
|      1000000000|Files by Google: ...|
|      1000000000|       Google Calendar|
|      1000000000|Android Auto - Go...|
|      1000000000|Gboard - the Goog...|
|      1000000000|Google Play Books...|
```

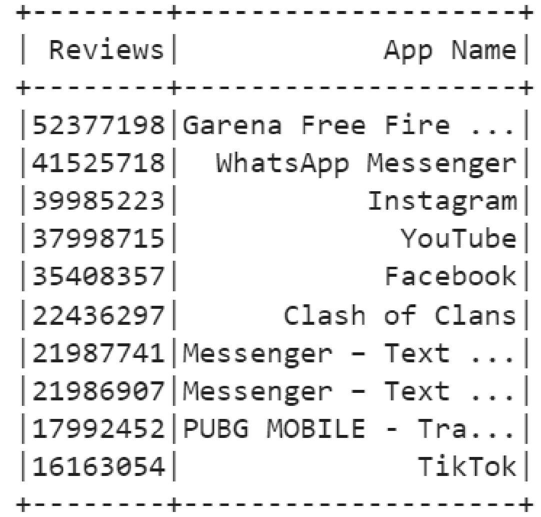


Top Apps which has Installations greater than a Billion

Top Apps which has highest number of Reviews?

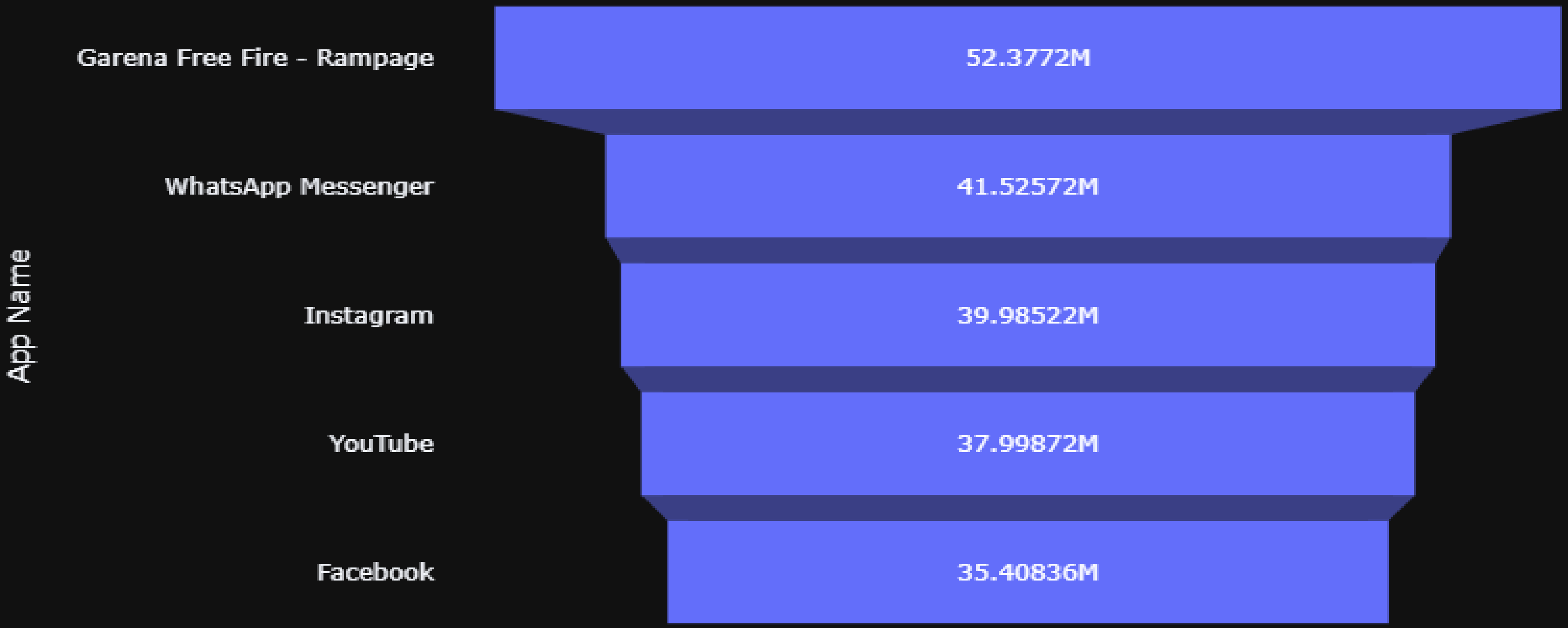


```
= dfAppReview.select('Reviews','App Name').filter(dfAppReview.Reviews != "N/A").orderBy('Reviews', ascending=F
= dfAppReview2.withColumn("Reviews", dfAppReview2["Reviews"].cast("int").alias("Reviews"))
= dfAppReview3.dropna()
= dfAppReviewCt.orderBy("Reviews",ascending=False).limit(10)
```



```
+--------+--------------------+
| Reviews|            App Name|
+--------+--------------------+
|52377198|Garena Free Fire ...|
|41525718|  WhatsApp Messenger|
|39985223|           Instagram|
|37998715|             YouTube|
|35408357|            Facebook|
|22436297|      Clash of Clans|
|21987741|Messenger – Text ...|
|21986907|Messenger – Text ...|
|17992452|PUBG MOBILE - Tra...|
|16163054|              TikTok|
+--------+--------------------+
```

Top 5 Apps which has highest number of Reviews



Conclusion

The dataset contains possibilities to deliver insights to understand customer demands better and thus help developers to popularize the product.

References:

https://depositphotos.com/87537316/stock-illustration-education-and-learning-icon.html

https://www.vectorstock.com/royalty-free-vector/girl-with-cell-phone-vector-995611

https://www.xda-developers.com/fix-common-problems-play-store-app/

Thank You!