GOOGLE APP STORE ANALYSIS

Name:Preethi kotaru

```
!pip install jovian opendatasets --upgrade --quiet
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.6/68.6 kB 3.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
  Building wheel for uuid (setup.py) ... done
```

```
!pip install plotly==3.10.0
from chart_studio import plotly
```

```
Collecting plotly==3.10.0
  Downloading plotly-3.10.0-py2.py3-none-any.whl (41.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.5/41.5 MB 12.8 MB/s eta 0:00:00
Requirement already satisfied: decorator>=4.0.6 in /usr/local/lib/python3.10/dist-packages (from plo
Requirement already satisfied: nbformat>=4.2 in /usr/local/lib/python3.10/dist-packages (from plotly
Requirement already satisfied: pytz in /usr/local/lib/python3.10/dist-packages (from plotly==3.10.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from plotly==3.1
Collecting retrying>=1.3.3 (from plotly==3.10.0)
  Downloading retrying-1.3.4-py3-none-any.whl (11 kB)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from plotly==3.10.0)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.10/dist-packages (from nbfor
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.10/dist-packages (from nbfo
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.10/dist-packages (from nbforma
Requirement already satisfied: traitlets>=5.1 in /usr/local/lib/python3.10/dist-packages (from nbfor
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from request
Requirement already satisfied: attrs>=17.4.0 in /usr/local/lib/python3.10/dist-packages (from jsonsc
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in /usr/local/lib/pytho
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from ju
Installing collected packages: retrying, plotly
  Attempting uninstall: plotly
    Found existing installation: plotly 5.13.1
    Uninstalling plotly-5.13.1:
      Successfully uninstalled plotly-5.13.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are inst
cufflinks 0.17.3 requires plotly>=4.1.1, but you have plotly 3.10.0 which is incompatible.
Successfully installed plotly-3.10.0 retrying-1.3.4
```

```
from chart_studio import plotly
```

```
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import numpy as np # linear algebra
import matplotlib.pyplot as plt
import seaborn as sns  # visualization tool
# plotly
import plotly.plotly as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go

# word cloud library
from wordcloud import WordCloud
```

```
data = pd.read_csv("/content/Google Apps data.csv")
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8276 entries, 0 to 8275
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Unnamed: 0.1         8276 non-null   int64  
 1   Unnamed: 0           8276 non-null   int64  
 2   App                  8276 non-null   object 
 3   Category             8276 non-null   object 
 4   Rating               8276 non-null   float64
 5   Reviews              8276 non-null   int64  
 6   Size                 8276 non-null   float64
 7   Installs             8276 non-null   int64  
 8   Type                 8276 non-null   object 
 9   Price                8276 non-null   float64
 10  Content Rating       7915 non-null   object 
 11  Last Updated         8276 non-null   object 
 12  Current Ver          8276 non-null   object 
 13  Minimum Android Ver  8276 non-null   object 
 14  Genres               8276 non-null   object 
dtypes: float64(3), int64(4), object(8)
memory usage: 970.0+ KB
```

```
data.shape
```

```
(8276, 15)
```

```
data.head()
```

```
data['Category'].unique()
```

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

```
data['Rating'].unique()
```

```
array([4.1, 3.9, 4.7, 4.5, 4.3, 4.4, 3.8, 4.2, 4.6, 4. , 4.8, 4.9, 3.6,
       3.7, 3.2, 3.3, 3.4, 3.5, 3.1, 5. , 2.6, 3. , 1.9, 2.5, 2.8, 2.7,
       1. , 2.9, 2.3, 2.2, 1.7, 2. , 1.8, 2.4, 1.6, 2.1, 1.4, 1.5, 1.2])
```

```
data['Rating'] = pd.to_numeric(data['Rating'], errors='coerce')
data['Rating'].dtype
```

```
dtype('float64')
```

```
data['Reviews'].unique()
```

```
array([   159,    967,  87510, ...,    603,   1195, 398307])
```

```
data[data['Reviews'] =='3.0M']
```

```
data['Reviews'] = data.Reviews.replace("0.0",0)
data['Reviews'] = data.Reviews.replace("3.0M",3000000.0)
data['Reviews'] = data['Reviews'].astype(float)
data['Reviews'].dtype
```

```
dtype('float64')
```

```
data['Size'].unique()
```

```
array([1.9000e+01, 1.4000e+01, 8.7000e+00, 2.5000e+01, 2.8000e+00,
       5.6000e+00, 2.9000e+01, 3.3000e+01, 3.1000e+00, 2.8000e+01,
       1.2000e+01, 2.0000e+01, 2.1000e+01, 3.7000e+01, 5.5000e+00,
       1.7000e+01, 3.9000e+01, 3.1000e+01, 4.2000e+00, 2.3000e+01,
       6.0000e+00, 6.1000e+00, 4.6000e+00, 9.2000e+00, 5.2000e+00,
       1.1000e+01, 2.4000e+01, 1.0000e+00, 9.4000e+00, 1.5000e+01,
       1.0000e+01, 1.2000e+00, 2.6000e+01, 8.0000e+00, 7.9000e+00,
       5.6000e+01, 5.7000e+01, 3.5000e+01, 5.4000e+01, 1.9629e-01,
       3.6000e+00, 5.7000e+00, 8.6000e+00, 2.4000e+00, 2.7000e+01,
       2.7000e+00, 2.5000e+00, 7.0000e+00, 1.6000e+01, 3.4000e+00,
       8.9000e+00, 3.9000e+00, 2.9000e+00, 3.8000e+01, 3.2000e+01,
       5.4000e+00, 1.8000e+01, 1.1000e+00, 2.2000e+00, 4.5000e+00,
       9.8000e+00, 5.2000e+01, 9.0000e+00, 6.7000e+00, 3.0000e+01,
       2.6000e+00, 7.1000e+00, 2.2000e+01, 6.4000e+00, 3.2000e+00,
       8.2000e+00, 4.9000e+00, 9.5000e+00, 5.0000e+00, 5.9000e+00,
       1.3000e+01, 7.3000e+01, 6.8000e+00, 3.5000e+00, 4.0000e+00,
       2.3000e+00, 2.1000e+00, 4.2000e+01, 9.1000e+00, 5.5000e+01,
       2.2460e-02, 7.3000e+00, 6.5000e+00, 1.5000e+00, 7.5000e+00,
       5.1000e+01, 4.1000e+01, 4.8000e+01, 8.5000e+00, 4.6000e+01,
       8.3000e+00, 4.3000e+00, 4.7000e+00, 3.3000e+00, 4.0000e+01,
       7.8000e+00, 8.8000e+00, 6.6000e+00, 5.1000e+00, 6.1000e+01,
       6.6000e+01, 7.7150e-02, 8.4000e+00, 3.7000e+00, 1.1523e-01,
       4.4000e+01, 6.7871e-01, 1.6000e+00, 6.2000e+00, 5.3000e+01,
       1.4000e+00, 3.0000e+00, 7.2000e+00, 5.8000e+00, 3.8000e+00,
       9.6000e+00, 4.5000e+01, 6.3000e+01, 4.9000e+01, 7.7000e+01,
       4.4000e+00, 7.0000e+01, 9.3000e+00, 8.1000e+00, 3.6000e+01,
       6.9000e+00, 7.4000e+00, 8.4000e+01, 9.7000e+01, 2.0000e+00,
       1.9000e+00, 1.8000e+00, 5.3000e+00, 4.7000e+01, 5.4297e-01,
       5.1367e-01, 7.6000e+01, 7.6000e+00, 5.9000e+01, 9.7000e+00,
       7.8000e+01, 7.2000e+01, 4.3000e+01, 7.7000e+00, 6.3000e+00,
       3.2617e-01, 9.3000e+01, 6.5000e+01, 7.9000e+01, 1.0000e+02,
       5.8000e+01, 5.0000e+01, 6.8000e+01, 6.4000e+01, 3.4000e+01,
       6.7000e+01, 6.0000e+01, 9.4000e+01, 9.9000e+00, 2.2656e-01,
       9.9000e+01, 6.0938e-01, 9.5000e+01, 8.3000e-03, 4.0040e-02,
       2.8516e-01, 8.0000e+01, 1.7000e+00, 7.4000e+01, 6.2000e+01,
       6.9000e+01, 7.5000e+01, 9.8000e+01, 8.5000e+01, 8.2000e+01,
       9.6000e+01, 8.7000e+01, 7.1000e+01, 8.6000e+01, 9.1000e+01,
       8.1000e+01, 9.2000e+01, 8.3000e+01, 8.8000e+01, 6.8750e-01,
       8.4180e-01, 8.7793e-01, 3.6914e-01, 4.8000e+00, 2.5977e-01,
       3.6621e-01, 1.3000e+00, 9.5215e-01, 9.5703e-01, 4.1000e+00,
       8.9000e+01, 6.7969e-01, 5.3125e-01, 5.1270e-01, 8.9844e-01,
       7.6074e-01, 8.3301e-01, 7.0312e-01, 6.9629e-01, 7.5391e-01,
```

```
3.1055e-01, 5.6640e-02, 2.3535e-01, 1.9141e-01, 8.3691e-01,
4.9800e-02, 9.3066e-01, 8.4473e-01, 2.4512e-01, 9.0820e-01,
5.2734e-01, 3.0566e-01, 7.2852e-01, 1.9824e-01, 2.5390e-02,
3.0664e-01, 2.3340e-01, 3.6230e-01, 2.1484e-01, 7.1289e-01,
7.3828e-01, 8.8870e-02, 2.8613e-01, 1.6600e-02, 7.2270e-02,
1.3670e-02, 3.0957e-01, 7.6170e-02, 9.0234e-01, 7.9883e-01,
7.9100e-02, 9.1699e-01, 1.6504e-01, 4.3950e-02, 9.4238e-01,
9.0000e+01, 5.3223e-01, 5.9570e-02, 2.7637e-01, 6.3965e-01,
6.9727e-01, 9.0820e-02, 8.5156e-01, 1.1816e-01, 3.1445e-01,
9.5312e-01, 2.0117e-01, 9.3164e-01, 4.3359e-01, 7.0020e-01,
2.0508e-01, 5.9473e-01, 3.0078e-01, 2.9883e-01, 1.7090e-01,
3.4180e-01, 3.7402e-01, 4.4336e-01, 6.8360e-02, 7.9297e-01,
4.3164e-01, 8.2227e-01, 4.0723e-01, 4.0234e-01, 4.4824e-01,
4.6680e-01, 3.2715e-01, 7.6367e-01, 7.0410e-01, 4.1992e-01,
4.1895e-01, 1.8750e-01, 4.4922e-01, 7.1094e-01, 4.8438e-01,
```

```
data['Size'].dtype
```

```
dtype('float64')
```

```
data['Installs'].unique()
```

```
array([     10000,     500000,    5000000,   50000000,     100000,
            50000,    1000000,   10000000,       5000,  100000000,
       1000000000,       1000,  500000000,        100,        500,
               10,          5,         50,          1])
```

```
data['Installs'] = data['Installs'].astype(float)
data['Installs'].dtype
```

```
dtype('float64')
```

```
data['Price'].unique()
```

```
array([  0.  ,   4.99,   3.99,   6.99,   7.99,   5.99,   2.99,   3.49,
         1.99,   9.99,   7.49,   0.99,   9.  ,   5.49,  10.  ,  24.99,
        11.99,  79.99,  16.99,  14.99,  29.99,  12.99,   2.49,  10.99,
         1.5 ,  19.99,  15.99,  33.99,  39.99,   3.95,   4.49,   1.7 ,
         8.99,   1.49,   3.88, 399.99,  17.99, 400.  ,   3.02,   1.76,
         4.84,   4.77,   1.61,   2.5 ,   1.59,   6.49,   1.29, 299.99,
       379.99,  37.99,  18.99, 389.99,   8.49,   1.75,  14.  ,   2.  ,
         3.08,   2.59,  19.4 ,   3.9 ,   4.59,  15.46,   3.04,  13.99,
         4.29,   3.28,   4.6 ,   1.  ,   2.95,   2.9 ,   1.97,   2.56,
         1.2 ])
```

```
data['Price'] = data['Price'].astype(float)
data['Price'].dtype
```

```
dtype('float64')
```

```
data['Last Updated'].unique()
```

```
array(['January 7, 2018', 'January 15, 2018', 'August 1, 2018', ...,
       'January 20, 2014', 'February 16, 2014', 'March 23, 2014'],
      dtype=object)
```

```
data['Last Updated'] = pd.to_datetime(data['Last Updated'])
data['Last Updated']
```

```
0      2018-01-07
1      2018-01-15
2      2018-08-01
3      2018-06-08
4      2018-06-20
          ...
8271   2017-06-18
8272   2017-07-25
8273   2018-07-06
8274   2015-01-19
```

```
8275   2018-07-25
Name: Last Updated, Length: 8276, dtype: datetime64[ns]
```

```
data.corr()
```

```
f,ax = plt.subplots(figsize=(12, 12))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```

```
data.describe()
```

```
category_list = list(data['Category'].unique())
category_review = []
for i in category_list:
    x = data[data['Category'] == i]
    if(len(x)!=0):
        review = sum(x.Reviews)/len(x)
        category_review.append(review)
    else:
        review = sum(x.Reviews)
        category_review.append(review)
#sorting
data_category_reviews = pd.DataFrame({'category': category_list,'review':category_review})
new_index = (data_category_reviews['review'].sort_values(ascending=False)).index.values
sorted_data =data_category_reviews.reindex(new_index)

# visualization
plt.figure(figsize=(15,10))
sns.barplot(x=sorted_data['category'], y=sorted_data['review'])
plt.xticks(rotation=80)
plt.xlabel("Category")
plt.ylabel("Reviews")
plt.title("Category and Reviews")
plt.show()
```

```
category_list = list(data['Category'].unique())
category_install = []
for i in category_list:
    x = data[data['Category'] == i]
    if(len(x)!=0):
        install = sum(x.Installs)/len(x)
        category_install.append(install)
    else:
        install = sum(x.Installs)
        category_install.append(install)

#sorting
data_category_install = pd.DataFrame({'category': category_list,'install':category_install})
new_index = (data_category_install['install'].sort_values(ascending=False)).index.values
sorted_data =data_category_install.reindex(new_index)

# visualization
plt.figure(figsize=(15,10))
sns.barplot(x=sorted_data['category'], y=sorted_data['install'])
plt.xticks(rotation=80)
plt.xlabel("Category")
plt.ylabel("Install")
plt.title("Category and Install")
plt.show()
```

```
plt.figure(figsize=(10,7))
sns.countplot(data=data, x='Content Rating')
plt.xticks(rotation=80)
plt.title('Content Rating',color = 'blue',fontsize=15)
plt.show()
```