tmxmzmeje

April 10, 2024

1 Play Store Data Analysis: Unveiling Trends and Patterns

#**Introduction:** In today's digital age, mobile applications have become an integral part of our daily lives, serving various purposes from entertainment to productivity. Google Play Store, one of the largest repositories of mobile applications, offers a vast array of apps catering to different needs and preferences of users. This project delves into the realm of Play Store data analysis, aiming to uncover insights, trends, and correlations among various attributes of apps available on the platform.

#**Objective:** The primary objective of this project is to conduct a comprehensive analysis of Google Play Store data to gain valuable insights into the characteristics, behavior, and preferences of app users. The key goals include:

1. Data Cleaning: Preprocessing the raw data to handle missing values, outliers, and inconsistencies, ensuring data integrity and reliability.
2. Exploratory Data Analysis (EDA): Exploring the dataset to identify patterns, distributions, and trends among different features such as app categories, genres, ratings, reviews, sizes, and installs.
3. Correlation Analysis: Investigating the relationships and correlations between various attributes such as app ratings, reviews, sizes, and the number of installs to uncover underlying patterns and dependencies.
4. Comparison between Free and Paid Apps: Analyzing the differences and similarities between free and paid apps in terms of their characteristics, user engagement, and popularity.
5. Version Updates Analysis: Studying the impact of version updates on app performance, user satisfaction, and ratings.
6. Genre and Category Trends: Identifying the most popular app genres and categories based on user preferences and demand.

#**Methodology:** The project will involve the following steps:

1. Data Acquisition: Gathering Play Store data using web scraping techniques or utilizing publicly available datasets.
2. Data Cleaning: Performing data cleaning and preprocessing tasks including handling missing values, removing duplicates, and standardizing data formats.
3. Exploratory Data Analysis: Conducting exploratory data analysis to visualize distributions, trends, and patterns among different attributes using statistical measures and visualization tools.

4. Correlation Analysis: Calculating correlation coefficients and conducting statistical tests to identify relationships and dependencies among various features.
5. Comparison Analysis: Comparing the characteristics and performance metrics of free and paid apps to discern any significant differences.
6. Genre and Category Trends Analysis: Analyzing the popularity and trends of different app genres and categories based on user preferences and download statistics.
7. Version Updates Analysis: Investigating the impact of version updates on app ratings, reviews, and user engagement metrics.

2 Dataset Descriptions

App: Name of the mobile application.

Category: Category or genre to which the app belongs.

Rating: User rating score of the app.

Reviews: Number of user reviews/ratings for the app.

Size: Size of the app installation package.

Installs: Number of times the app has been installed.

Type: Whether the app is free or paid.

Price: Price of the app if it's paid; otherwise, '0' or 'Free'.

Content Rating: Content rating indicating suitability for different age groups.

Genres: Specific genres or themes associated with the app.

Last Updated: Date when the app was last updated.

Current Ver: Current version of the app.

Android Ver: Minimum required Android version to run the app.

[ ]:
```
pip install tabulate
```

```
Requirement already satisfied: tabulate in /usr/local/lib/python3.10/dist-
packages (0.9.0)
```

[ ]:
```
!pip install plotly_express
```

```
Collecting plotly_express
  Downloading plotly_express-0.4.1-py2.py3-none-any.whl (2.9 kB)
Requirement already satisfied: pandas>=0.20.0 in /usr/local/lib/python3.10/dist-
packages (from plotly_express) (2.0.3)
Requirement already satisfied: plotly>=4.1.0 in /usr/local/lib/python3.10/dist-
packages (from plotly_express) (5.15.0)
Requirement already satisfied: statsmodels>=0.9.0 in
/usr/local/lib/python3.10/dist-packages (from plotly_express) (0.14.1)
Requirement already satisfied: scipy>=0.18 in /usr/local/lib/python3.10/dist-
packages (from plotly_express) (1.11.4)
```

```
Requirement already satisfied: patsy>=0.5 in /usr/local/lib/python3.10/dist-
packages (from plotly_express) (0.5.6)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.10/dist-
packages (from plotly_express) (1.25.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas>=0.20.0->plotly_express)
(2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=0.20.0->plotly_express) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=0.20.0->plotly_express) (2024.1)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5->plotly_express) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from plotly>=4.1.0->plotly_express)
(8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from plotly>=4.1.0->plotly_express) (24.0)
Installing collected packages: plotly_express
Successfully installed plotly_express-0.4.1
```

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random
from tabulate import tabulate
from wordcloud import WordCloud
import plotly.graph_objects as go
```

[ ]:
```
Playstore="/content/drive/MyDrive/Analysis/googleplaystore.csv"
```

[ ]:
```
df1=pd.read_csv(Playstore)
```

[ ]:
```
df1.head(10)
```

[ ]:

	App	Category	Rating	\
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	
1	Coloring book moana	ART_AND_DESIGN	3.9	
2	U Launcher Lite - FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	
5	Paper flowers instructions	ART_AND_DESIGN	4.4	
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	
7	Infinite Painter	ART_AND_DESIGN	4.1	
8	Garden Coloring Book	ART_AND_DESIGN	4.4	
9	Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	

```
  Reviews  Size     Installs  Type Price Content Rating  \
0     159   19M      10,000+  Free     0       Everyone
1     967   14M     500,000+  Free     0       Everyone
2   87510  8.7M   5,000,000+  Free     0       Everyone
3  215644   25M  50,000,000+  Free     0           Teen
4     967  2.8M     100,000+  Free     0       Everyone
5     167  5.6M      50,000+  Free     0       Everyone
6     178   19M      50,000+  Free     0       Everyone
7   36815   29M   1,000,000+  Free     0       Everyone
8   13791   33M   1,000,000+  Free     0       Everyone
9     121  3.1M      10,000+  Free     0       Everyone

                      Genres        Last Updated         Current Ver  \
0               Art & Design     January 7, 2018               1.0.0
1  Art & Design;Pretend Play    January 15, 2018               2.0.0
2               Art & Design      August 1, 2018               1.2.4
3               Art & Design        June 8, 2018  Varies with device
4    Art & Design;Creativity       June 20, 2018                 1.1
5               Art & Design      March 26, 2017                 1.0
6               Art & Design      April 26, 2018                 1.1
7               Art & Design       June 14, 2018           6.1.61.1
8               Art & Design  September 20, 2017               2.9.2
9    Art & Design;Creativity        July 3, 2018                 2.8

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
5    2.3 and up
6  4.0.3 and up
7    4.2 and up
8    3.0 and up
9  4.0.3 and up
```

[ ]:
```
df1.describe()
```

[ ]:
```
            Rating
count  9367.000000
mean      4.193338
std       0.537431
min       1.000000
25%       4.000000
50%       4.300000
75%       4.500000
```

```
max      19.000000
```

[ ]:
```
df1.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
```

3 Data Cleaning

[ ]:
```
null_value=df1.isnull()
```

[ ]:
```
nullvalue_count=null_value.sum()
nullvalue_count
```

[ ]:
```
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
```

[ ]:
```
#Here it shows the rows which is having the NaN value in the column called Current Ver
null_indices = df1[df1['Current Ver'].isnull()].index

# Display the row indices and corresponding rows
for index in null_indices:
    print("Row index:", index)
    print(df1.loc[index])
    print()
```

```
Row index: 15
App                Learn To Draw Kawaii Characters
Category                            ART_AND_DESIGN
Rating                                        3.2
Reviews                                        55
Size                                         2.7M
Installs                                    5,000+
Type                                          Free
Price                                            0
Content Rating                            Everyone
Genres                                 Art & Design
Last Updated                           June 6, 2018
Current Ver                                     NaN
Android Ver                              4.2 and up
Name: 15, dtype: object

Row index: 1553
App                Market Update Helper
Category             LIBRARIES_AND_DEMO
Rating                              4.1
Reviews                           20145
Size                                11k
Installs                      1,000,000+
Type                                Free
Price                                  0
Content Rating                 Everyone
Genres                  Libraries & Demo
Last Updated          February 12, 2013
Current Ver                         NaN
Android Ver                1.5 and up
Name: 1553, dtype: object

Row index: 6322
App                Virtual DJ Sound Mixer
Category                            TOOLS
Rating                                4.2
Reviews                              4010
```

```
Size                              8.7M
Installs                       500,000+
Type                              Free
Price                                0
Content Rating                 Everyone
Genres                            Tools
Last Updated                May 10, 2017
Current Ver                          NaN
Android Ver                4.0 and up
Name: 6322, dtype: object

Row index: 6803
App                      BT Master
Category                    FAMILY
Rating                         NaN
Reviews                          0
Size                          222k
Installs                      100+
Type                          Free
Price                            0
Content Rating            Everyone
Genres                   Education
Last Updated      November 6, 2016
Current Ver                    NaN
Android Ver            1.6 and up
Name: 6803, dtype: object

Row index: 7333
App                   Dots puzzle
Category                   FAMILY
Rating                        4.0
Reviews                       179
Size                          14M
Installs                  50,000+
Type                         Paid
Price                       $0.99
Content Rating           Everyone
Genres                     Puzzle
Last Updated      April 18, 2018
Current Ver                    NaN
Android Ver          4.0 and up
Name: 7333, dtype: object

Row index: 7407
App               Calculate My IQ
Category                   FAMILY
Rating                        NaN
Reviews                        44
```

```
Size                        7.2M
Installs                  10,000+
Type                         Free
Price                           0
Content Rating           Everyone
Genres              Entertainment
Last Updated        April 3, 2017
Current Ver                   NaN
Android Ver           2.3 and up
Name: 7407, dtype: object

Row index: 7730
App                      UFO-CQ
Category                  TOOLS
Rating                      NaN
Reviews                       1
Size                       237k
Installs                    10+
Type                       Paid
Price                     $0.99
Content Rating        Everyone
Genres                     Tools
Last Updated       July 4, 2016
Current Ver                 NaN
Android Ver         2.0 and up
Name: 7730, dtype: object

Row index: 10342
App                     La Fe de Jesus
Category          BOOKS_AND_REFERENCE
Rating                            NaN
Reviews                             8
Size                             658k
Installs                        1,000+
Type                             Free
Price                               0
Content Rating               Everyone
Genres              Books & Reference
Last Updated         January 31, 2017
Current Ver                       NaN
Android Ver                3.0 and up
Name: 10342, dtype: object
```

[ ]:
```
#Here it shows the rows which is having the NaN value in the column called Type
null_indices = df1[df1['Type'].isnull()].index
```

```
# Display the row indices and corresponding rows
for index in null_indices:
    print("Row index:", index)
    print(df1.loc[index])
    print()
```

```
Row index: 9148
App                Command & Conquer: Rivals
Category                            FAMILY
Rating                                 NaN
Reviews                                  0
Size                     Varies with device
Installs                                 0
Type                                   NaN
Price                                    0
Content Rating                Everyone 10+
Genres                            Strategy
Last Updated                 June 28, 2018
Current Ver              Varies with device
Android Ver              Varies with device
Name: 9148, dtype: object
```

[ ]:
```
#Here it shows the rows which is having the NaN value in the column called
 ↪Content Rating
null_indices = df1[df1['Content Rating'].isnull()].index

# Display the row indices and corresponding rows
for index in null_indices:
    print("Row index:", index)
    print(df1.loc[index])
    print()
```

```
Row index: 10472
App               Life Made WI-Fi Touchscreen Photo Frame
Category                                              1.9
Rating                                               19.0
Reviews                                              3.0M
Size                                                1,000+
Installs                                              Free
Type                                                     0
Price                                             Everyone
Content Rating                                         NaN
Genres                                    February 11, 2018
Last Updated                                         1.0.19
Current Ver                                  4.0 and up
Android Ver                                            NaN
```

```
Name: 10472, dtype: object
```

[ ]:
```
#Here it shows the rows which is having the NaN value in the column called␣
 ↪Android Ver
null_indices = df1[df1['Android Ver'].isnull()].index

# Display the row indices and corresponding rows
for index in null_indices:
    print("Row index:", index)
    print(df1.loc[index])
    print()
```

```
Row index: 4453
App                [substratum] Vacuum: P
Category                   PERSONALIZATION
Rating                                 4.4
Reviews                                230
Size                                   11M
Installs                             1,000+
Type                                   Paid
Price                                $1.49
Content Rating                    Everyone
Genres                     Personalization
Last Updated                 July 20, 2018
Current Ver                            4.4
Android Ver                            NaN
Name: 4453, dtype: object

Row index: 4490
App                 Pi Dark [substratum]
Category                 PERSONALIZATION
Rating                               4.5
Reviews                              189
Size                                2.1M
Installs                          10,000+
Type                                 Free
Price                                   0
Content Rating                  Everyone
Genres                   Personalization
Last Updated              March 27, 2018
Current Ver                          1.1
Android Ver                          NaN
Name: 4490, dtype: object

Row index: 10472
App                 Life Made WI-Fi Touchscreen Photo Frame
Category                                                1.9
```

```
Rating                                                19.0
Reviews                                               3.0M
Size                                                 1,000+
Installs                                               Free
Type                                                      0
Price                                              Everyone
Content Rating                                          NaN
Genres                                  February 11, 2018
Last Updated                                        1.0.19
Current Ver                                   4.0 and up
Android Ver                                             NaN
Name: 10472, dtype: object
```

[ ]:
```python
# Find row indices with NaN values in a Rating column
nan_indices = df1[df1['Rating'].isnull()].index

# Print the row indices horizontally
print("Row indices with NaN values in column 'Rating':")
for index in nan_indices:
    print(index, end=', ')
```

```
Row indices with NaN values in column 'Rating':
23, 113, 123, 126, 129, 130, 134, 163, 180, 185, 227, 321, 478, 479, 480, 610,
613, 617, 620, 621, 623, 624, 626, 627, 628, 629, 630, 631, 632, 635, 637, 638,
639, 641, 642, 643, 644, 647, 648, 649, 650, 651, 652, 653, 658, 659, 660, 666,
669, 683, 698, 704, 1013, 1024, 1025, 1032, 1033, 1034, 1039, 1041, 1043, 1045,
1046, 1177, 1180, 1455, 1469, 1478, 1484, 1489, 1517, 1519, 1520, 1530, 1534,
1536, 1537, 1539, 1540, 1552, 1557, 1559, 2111, 2265, 2280, 2294, 2356, 2390,
2419, 2422, 2423, 2424, 2425, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437,
2438, 2439, 2440, 2441, 2442, 2443, 2446, 2447, 2449, 2451, 2453, 2456, 2460,
2461, 2462, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2473, 2474, 2478, 2479,
2480, 2481, 2483, 2485, 2498, 2501, 2502, 2503, 2513, 2516, 2529, 2540, 3252,
3258, 3577, 3579, 3580, 3583, 3585, 3589, 3599, 3603, 3616, 3621, 3705, 3959,
3985, 4130, 4134, 4137, 4139, 4177, 4201, 4218, 4256, 4265, 4269, 4270, 4273,
4278, 4303, 4307, 4308, 4315, 4431, 4451, 4457, 4459, 4462, 4465, 4471, 4475,
4482, 4485, 4489, 4504, 4520, 4532, 4537, 4538, 4543, 4546, 4548, 4549, 4550,
4551, 4562, 4564, 4565, 4697, 4776, 4845, 4848, 4884, 4889, 4894, 5023, 5029,
5034, 5050, 5051, 5052, 5056, 5059, 5061, 5068, 5072, 5086, 5088, 5090, 5093,
5096, 5098, 5099, 5100, 5101, 5106, 5107, 5109, 5110, 5111, 5112, 5113, 5114,
5115, 5116, 5119, 5120, 5122, 5123, 5124, 5126, 5127, 5131, 5132, 5134, 5135,
5136, 5140, 5141, 5142, 5147, 5152, 5154, 5155, 5157, 5159, 5160, 5161, 5162,
5163, 5164, 5165, 5166, 5168, 5169, 5170, 5171, 5172, 5174, 5176, 5177, 5208,
5213, 5217, 5218, 5229, 5231, 5232, 5234, 5240, 5243, 5247, 5249, 5250, 5253,
5255, 5256, 5259, 5262, 5264, 5265, 5266, 5269, 5270, 5271, 5274, 5281, 5288,
5289, 5290, 5296, 5298, 5300, 5301, 5302, 5303, 5307, 5407, 5414, 5457, 5461,
5480, 5484, 5486, 5494, 5497, 5498, 5499, 5500, 5508, 5510, 5511, 5513, 5564,
5575, 5669, 5676, 5677, 5680, 5682, 5683, 5687, 5688, 5690, 5694, 5713, 5730,
```

5734, 5739, 5742, 5750, 5753, 5754, 5756, 5757, 5762, 5771, 5788, 5792, 5793,
5796, 5798, 5800, 5804, 5807, 5809, 5810, 5812, 5816, 5817, 5824, 5832, 5833,
5834, 5835, 5837, 5838, 5839, 5840, 5841, 5843, 5845, 5846, 5851, 5853, 5854,
5855, 5857, 5882, 5891, 5918, 5921, 5923, 5924, 5925, 5927, 5942, 5943, 5945,
5951, 5967, 5976, 5977, 5985, 5991, 5992, 5998, 6011, 6085, 6109, 6110, 6114,
6117, 6118, 6122, 6138, 6139, 6143, 6145, 6147, 6149, 6150, 6151, 6152, 6153,
6154, 6157, 6158, 6162, 6163, 6164, 6165, 6170, 6172, 6173, 6174, 6176, 6179,
6182, 6186, 6190, 6192, 6199, 6208, 6225, 6229, 6230, 6231, 6233, 6234, 6235,
6237, 6239, 6240, 6243, 6246, 6250, 6257, 6258, 6259, 6260, 6261, 6277, 6278,
6291, 6292, 6318, 6320, 6325, 6328, 6329, 6333, 6334, 6335, 6336, 6337, 6339,
6345, 6357, 6360, 6365, 6366, 6367, 6377, 6389, 6393, 6394, 6399, 6425, 6428,
6430, 6433, 6434, 6435, 6436, 6437, 6443, 6444, 6445, 6456, 6460, 6464, 6469,
6470, 6473, 6475, 6478, 6479, 6480, 6481, 6482, 6483, 6487, 6488, 6492, 6499,
6502, 6508, 6518, 6519, 6527, 6528, 6530, 6542, 6544, 6545, 6555, 6559, 6560,
6567, 6569, 6596, 6598, 6606, 6613, 6623, 6624, 6633, 6634, 6637, 6638, 6639,
6641, 6642, 6644, 6653, 6657, 6659, 6664, 6665, 6667, 6668, 6669, 6670, 6671,
6672, 6683, 6689, 6692, 6693, 6695, 6696, 6699, 6701, 6703, 6704, 6706, 6710,
6711, 6732, 6733, 6734, 6736, 6738, 6742, 6743, 6746, 6750, 6751, 6753, 6755,
6778, 6780, 6795, 6800, 6801, 6803, 6804, 6805, 6808, 6809, 6815, 6817, 6818,
6819, 6821, 6826, 6828, 6829, 6831, 6832, 6834, 6835, 6838, 6839, 6841, 6842,
6843, 6844, 6846, 6853, 6854, 6855, 6857, 6859, 6862, 6864, 6865, 6866, 6868,
6869, 6871, 6872, 6873, 6875, 6876, 6878, 6879, 6880, 6888, 6890, 6893, 6904,
6912, 6915, 6916, 6917, 6918, 6928, 6932, 6933, 6937, 6938, 6940, 6941, 6950,
6952, 6954, 6955, 6956, 6959, 6962, 6965, 6967, 7034, 7038, 7040, 7047, 7049,
7051, 7053, 7056, 7058, 7060, 7061, 7064, 7073, 7074, 7081, 7083, 7084, 7087,
7092, 7101, 7114, 7116, 7117, 7120, 7121, 7123, 7129, 7130, 7133, 7135, 7138,
7140, 7141, 7153, 7154, 7156, 7157, 7158, 7160, 7161, 7163, 7166, 7167, 7168,
7169, 7171, 7172, 7173, 7174, 7176, 7180, 7181, 7182, 7183, 7184, 7185, 7188,
7189, 7190, 7199, 7202, 7203, 7208, 7211, 7214, 7217, 7220, 7221, 7222, 7224,
7225, 7226, 7227, 7228, 7230, 7231, 7232, 7233, 7234, 7246, 7249, 7251, 7252,
7255, 7256, 7257, 7259, 7262, 7263, 7264, 7265, 7268, 7269, 7271, 7272, 7274,
7275, 7276, 7277, 7278, 7279, 7293, 7294, 7296, 7305, 7306, 7308, 7311, 7312,
7317, 7318, 7322, 7332, 7335, 7339, 7342, 7343, 7344, 7346, 7361, 7370, 7371,
7376, 7378, 7380, 7382, 7384, 7386, 7387, 7388, 7389, 7390, 7391, 7394, 7395,
7396, 7400, 7401, 7407, 7409, 7410, 7432, 7434, 7440, 7445, 7447, 7448, 7460,
7472, 7476, 7481, 7485, 7486, 7488, 7493, 7494, 7502, 7503, 7504, 7505, 7516,
7521, 7522, 7526, 7528, 7529, 7531, 7532, 7581, 7622, 7629, 7642, 7652, 7663,
7679, 7681, 7691, 7694, 7702, 7703, 7710, 7712, 7713, 7714, 7719, 7720, 7724,
7726, 7728, 7730, 7732, 7735, 7737, 7745, 7747, 7748, 7757, 7759, 7775, 7778,
7787, 7790, 7800, 7804, 7827, 7832, 7843, 7854, 7862, 7873, 7874, 7878, 7882,
7884, 7888, 7891, 7893, 7897, 7898, 7901, 7981, 7994, 8005, 8007, 8009, 8011,
8017, 8019, 8055, 8057, 8060, 8062, 8065, 8066, 8067, 8068, 8069, 8070, 8072,
8076, 8078, 8079, 8080, 8081, 8091, 8093, 8094, 8097, 8103, 8105, 8106, 8109,
8110, 8112, 8113, 8114, 8119, 8137, 8139, 8140, 8145, 8147, 8148, 8150, 8151,
8154, 8155, 8160, 8161, 8165, 8169, 8206, 8209, 8210, 8217, 8218, 8223, 8225,
8232, 8241, 8242, 8243, 8262, 8278, 8280, 8282, 8283, 8318, 8319, 8322, 8328,
8329, 8330, 8331, 8332, 8333, 8334, 8337, 8338, 8339, 8340, 8341, 8342, 8344,
8345, 8346, 8348, 8349, 8351, 8354, 8355, 8356, 8358, 8359, 8384, 8389, 8390,

```
8393, 8414, 8425, 8431, 8435, 8488, 8489, 8490, 8495, 8499, 8501, 8502, 8504,
8505, 8506, 8509, 8510, 8548, 8556, 8562, 8565, 8566, 8567, 8568, 8573, 8575,
8576, 8577, 8578, 8579, 8580, 8581, 8583, 8588, 8593, 8597, 8598, 8599, 8600,
8602, 8603, 8605, 8606, 8607, 8612, 8614, 8616, 8690, 8698, 8729, 8752, 8822,
8825, 8828, 8831, 8833, 8834, 8839, 8842, 8849, 8851, 8854, 8868, 8870, 8871,
8872, 8873, 8876, 8877, 8882, 8884, 8885, 8921, 8928, 8939, 8941, 8965, 8968,
8969, 8973, 8985, 8987, 8988, 8999, 9001, 9004, 9009, 9012, 9014, 9015, 9016,
9019, 9020, 9021, 9025, 9030, 9043, 9047, 9052, 9053, 9054, 9058, 9074, 9075,
9076, 9077, 9083, 9087, 9096, 9097, 9101, 9104, 9109, 9110, 9113, 9115, 9118,
9123, 9125, 9131, 9133, 9136, 9137, 9139, 9148, 9180, 9183, 9185, 9189, 9190,
9195, 9198, 9199, 9201, 9206, 9209, 9210, 9224, 9231, 9234, 9235, 9237, 9240,
9249, 9250, 9252, 9255, 9257, 9261, 9262, 9264, 9271, 9284, 9291, 9294, 9296,
9298, 9299, 9300, 9302, 9306, 9309, 9310, 9312, 9319, 9320, 9321, 9322, 9324,
9329, 9330, 9331, 9333, 9337, 9346, 9367, 9370, 9371, 9373, 9409, 9410, 9412,
9414, 9417, 9418, 9419, 9421, 9423, 9431, 9448, 9453, 9458, 9459, 9460, 9466,
9481, 9502, 9504, 9509, 9514, 9515, 9519, 9521, 9527, 9532, 9548, 9550, 9557,
9567, 9612, 9648, 9651, 9652, 9653, 9654, 9655, 9656, 9657, 9658, 9660, 9662,
9663, 9664, 9665, 9666, 9669, 9670, 9672, 9675, 9679, 9691, 9692, 9693, 9698,
9699, 9700, 9703, 9704, 9706, 9707, 9713, 9714, 9715, 9719, 9722, 9727, 9730,
9739, 9757, 9772, 9777, 9782, 9820, 9834, 9866, 9874, 9878, 9882, 9884, 9889,
9893, 9896, 9898, 9902, 9903, 9905, 9907, 9909, 9910, 9912, 9913, 9914, 9915,
9916, 9917, 9918, 9919, 9920, 9924, 9925, 9927, 9928, 9929, 9930, 9931, 9934,
9935, 9937, 9938, 9956, 9962, 9964, 9971, 9972, 9980, 9989, 9990, 9992, 9993,
9994, 9995, 9997, 9999, 10000, 10002, 10004, 10007, 10009, 10053, 10069, 10078,
10079, 10080, 10081, 10083, 10084, 10085, 10089, 10090, 10094, 10097, 10099,
10101, 10102, 10103, 10104, 10105, 10106, 10108, 10109, 10110, 10119, 10169,
10176, 10180, 10183, 10184, 10185, 10187, 10214, 10232, 10236, 10256, 10265,
10299, 10311, 10314, 10328, 10329, 10330, 10331, 10332, 10333, 10334, 10337,
10338, 10339, 10342, 10344, 10346, 10347, 10348, 10350, 10351, 10352, 10354,
10355, 10356, 10358, 10359, 10361, 10364, 10365, 10366, 10367, 10368, 10369,
10377, 10399, 10401, 10402, 10403, 10404, 10411, 10413, 10414, 10415, 10417,
10418, 10421, 10422, 10425, 10427, 10432, 10433, 10439, 10483, 10485, 10488,
10494, 10495, 10496, 10497, 10499, 10501, 10510, 10517, 10519, 10524, 10525,
10528, 10530, 10532, 10533, 10534, 10535, 10536, 10537, 10538, 10539, 10541,
10542, 10543, 10544, 10545, 10546, 10547, 10548, 10550, 10551, 10553, 10555,
10557, 10558, 10559, 10560, 10561, 10563, 10565, 10567, 10605, 10606, 10608,
10612, 10618, 10622, 10625, 10626, 10627, 10631, 10650, 10651, 10653, 10654,
10657, 10658, 10660, 10661, 10662, 10664, 10668, 10669, 10670, 10672, 10674,
10676, 10692, 10693, 10694, 10695, 10696, 10698, 10699, 10700, 10701, 10702,
10704, 10705, 10708, 10709, 10734, 10735, 10736, 10744, 10745, 10746, 10748,
10751, 10759, 10761, 10762, 10764, 10769, 10772, 10773, 10774, 10775, 10788,
10794, 10798, 10806, 10807, 10808, 10811, 10813, 10816, 10818, 10821, 10822,
10823, 10824, 10825, 10831, 10835, 10838,
```

[ ]:
```
df= df1.dropna()

# Display the cleaned DataFrame
```

```
print(df)
```

```
                                                 App             Category  \
0         Photo Editor & Candy Camera & Grid & ScrapBook       ART_AND_DESIGN
1                                    Coloring book moana       ART_AND_DESIGN
2      U Launcher Lite - FREE Live Cool Themes, Hide …       ART_AND_DESIGN
3                                    Sketch - Draw & Paint       ART_AND_DESIGN
4                  Pixel Draw - Number Art Coloring Book       ART_AND_DESIGN
…                                                    …                  …
10834                                  FR Calculator               FAMILY
10836                                Sya9a Maroc - FR              FAMILY
10837                  Fr. Mike Schmitz Audio Teachings              FAMILY
10839                     The SCP Foundation DB fr nn5n  BOOKS_AND_REFERENCE
10840      iHoroscope - 2018 Daily Horoscope & Astrology            LIFESTYLE

       Rating Reviews                Size     Installs  Type Price  \
0         4.1     159                19M      10,000+  Free     0
1         3.9     967                14M     500,000+  Free     0
2         4.7   87510                8.7M   5,000,000+  Free     0
3         4.5  215644                25M  50,000,000+  Free     0
4         4.3     967                2.8M     100,000+  Free     0
…       …     …                  …          …   …   …
10834     4.0       7                2.6M         500+  Free     0
10836     4.5      38                53M        5,000+  Free     0
10837     5.0       4                3.6M         100+  Free     0
10839     4.5     114  Varies with device        1,000+  Free     0
10840     4.5  398307                19M  10,000,000+  Free     0

      Content Rating                     Genres      Last Updated  \
0           Everyone               Art & Design   January 7, 2018
1           Everyone  Art & Design;Pretend Play  January 15, 2018
2           Everyone               Art & Design    August 1, 2018
3               Teen               Art & Design      June 8, 2018
4           Everyone    Art & Design;Creativity     June 20, 2018
…              …                          …              …
10834       Everyone                  Education     June 18, 2017
10836       Everyone                  Education     July 25, 2017
10837       Everyone                  Education      July 6, 2018
10839     Mature 17+          Books & Reference  January 19, 2015
10840       Everyone                  Lifestyle     July 25, 2018

             Current Ver         Android Ver
0                  1.0.0       4.0.3 and up
1                  2.0.0       4.0.3 and up
2                  1.2.4       4.0.3 and up
3      Varies with device         4.2 and up
4                    1.1         4.4 and up
```

```
…                    …                  …
10834               1.0.0          4.1 and up
10836                1.48          4.1 and up
10837                 1.0          4.1 and up
10839  Varies with device  Varies with device
10840  Varies with device  Varies with device

[9360 rows x 13 columns]
```

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

[ ]:
```
                                                 App        Category  Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
1                                 Coloring book moana  ART_AND_DESIGN     3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide …  ART_AND_DESIGN     4.7
3                               Sketch - Draw & Paint  ART_AND_DESIGN     4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

  Reviews  Size     Installs  Type Price Content Rating  \
0     159   19M      10,000+  Free     0       Everyone
1     967   14M     500,000+  Free     0       Everyone
2   87510  8.7M   5,000,000+  Free     0       Everyone
3  215644   25M  50,000,000+  Free     0           Teen
4     967  2.8M     100,000+  Free     0       Everyone

                      Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    Art & Design;Creativity     June 20, 2018                 1.1

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

[ ]:
```
null_varify=df.isnull()
nullverify_count=null_varify.sum()
nullverify_count
```

[ ]:
```
App             0
Category        0
Rating          0
Reviews         0
```

```
Size              0
Installs          0
Type              0
Price             0
Content Rating    0
Genres            0
Last Updated      0
Current Ver       0
Android Ver       0
dtype: int64
```

[ ]:

[ ]:
```
# Remove '+' and ',' characters from the 'Installs' column
df['Installs'] = df['Installs'].str.replace('+', '').str.replace(',', '')

# Convert the 'Installs' column to integer type
df['Installs'] = df['Installs'].astype(int)
```

```
<ipython-input-107-cfe039087dd3>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Installs'] = df['Installs'].str.replace('+', '').str.replace(',', '')
<ipython-input-107-cfe039087dd3>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Installs'] = df['Installs'].astype(int)
```

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

[ ]:
```
                                                App        Category  Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
1                                 Coloring book moana  ART_AND_DESIGN     3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide …  ART_AND_DESIGN     4.7
3                               Sketch - Draw & Paint  ART_AND_DESIGN     4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

  Reviews  Size  Installs  Type Price Content Rating  \
0     159   19M     10000  Free     0       Everyone
1     967   14M    500000  Free     0       Everyone
2   87510  8.7M   5000000  Free     0       Everyone
```

```
3  215644   25M  50000000  Free     0           Teen
4     967  2.8M    100000  Free     0       Everyone

                      Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    Art & Design;Creativity     June 20, 2018                 1.1

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

[ ]:
```
df.shape
```

[ ]:
```
(9360, 13)
```

[ ]:
```
df.duplicated()
```

[ ]:
```
0        False
1        False
2        False
3        False
4        False
         …
10834    False
10836    False
10837    False
10839    False
10840    False
Length: 9360, dtype: bool
```

Cleaning

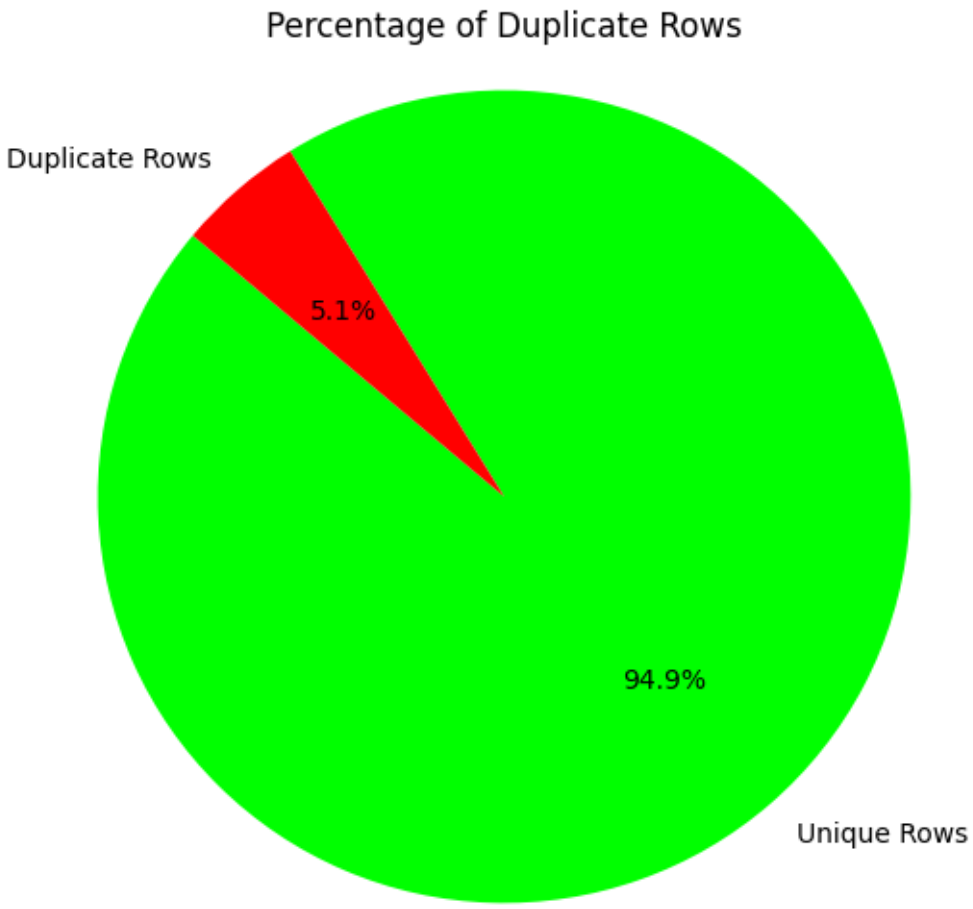
[ ]:
```
df.duplicated().value_counts()
```

[ ]:
```
False    8886
True      474
Name: count, dtype: int64
```

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

[ ]:
```
                                                 App        Category  Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
1                                Coloring book moana  ART_AND_DESIGN     3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide …  ART_AND_DESIGN     4.7
3                               Sketch - Draw & Paint  ART_AND_DESIGN     4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

  Reviews  Size  Installs  Type Price Content Rating  \
0     159   19M     10000  Free     0       Everyone
1     967   14M    500000  Free     0       Everyone
2   87510  8.7M   5000000  Free     0       Everyone
3  215644   25M  50000000  Free     0           Teen
4     967  2.8M    100000  Free     0       Everyone

                      Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    Art & Design;Creativity     June 20, 2018                 1.1

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

[ ]:
```
data = {'count': [8886, 474]}
index = ['Unique Rows', 'Duplicate Rows']
df2 = pd.Series(data['count'], index=index)

# Plot the pie chart
plt.figure(figsize=(6, 6))
plt.pie(df2, labels=df2.index, autopct='%1.1f%%', colors=['lime', 'red'],␣
 ↪startangle=140)

# Add title
plt.title('Percentage of Duplicate Rows')

# Show the plot
plt.axis('equal')
plt.show()
```



[ ]:
```
# sns.countplot(df.duplicated())
```

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

[ ]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10000	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500000	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	87510	8.7M	5000000	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50000000	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100000	Free	0	Everyone

```
                       Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    Art & Design;Creativity     June 20, 2018                 1.1

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9360 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             9360 non-null   object
 1   Category        9360 non-null   object
 2   Rating          9360 non-null   float64
 3   Reviews         9360 non-null   object
 4   Size            9360 non-null   object
 5   Installs        9360 non-null   int64
 6   Type            9360 non-null   object
 7   Price           9360 non-null   object
 8   Content Rating  9360 non-null   object
 9   Genres          9360 non-null   object
 10  Last Updated    9360 non-null   object
 11  Current Ver     9360 non-null   object
 12  Android Ver     9360 non-null   object
dtypes: float64(1), int64(1), object(11)
memory usage: 1.3+ MB
```

[ ]:
```

```

4 Exploratory Data Analysis

5 Analyzing 'Price' column

[ ]:
```
df['Price'].value_counts()
```

[ ]:
```
Price
0          8715
$2.99       114
$0.99       106
$4.99        70
$1.99        59
           …
$1.29          1
$299.99        1
$379.99        1
$37.99         1
$1.20          1
Name: count, Length: 73, dtype: int64
```

[ ]:
```
df['Price'].loc[df['Price'].str.contains('\$')].value_counts().sum()
```

[ ]:
```
645
```

[ ]:
```
df['Price'] = df['Price'].apply(lambda x : x.replace('$',"") if '$' in str(x)␣
 ↪else x)
```

```
<ipython-input-119-f37f83f0160d>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Price'] = df['Price'].apply(lambda x : x.replace('$',"") if '$' in str(x)
else x)
```

[ ]:
```
df['Price'].value_counts()
```

[ ]:
```
Price
0         8715
2.99       114
0.99       106
4.99        70
1.99        59
           …
1.29          1
299.99        1
```

```
379.99       1
37.99        1
1.20         1
Name: count, Length: 73, dtype: int64
```

[ ]:
```
df['Price'] = df['Price'].apply(lambda x: float(x))
```

```
<ipython-input-121-91aa49c83bfc>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Price'] = df['Price'].apply(lambda x: float(x))
```

6 Analyzing 'Size' column

[ ]:
```
df['Size'].unique()
```

[ ]:
```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M', '3.1M',
       '28M', '12M', '20M', '21M', '37M', '5.5M', '17M', '39M', '31M',
       '4.2M', '23M', '6.0M', '6.1M', '4.6M', '9.2M', '5.2M', '11M',
       '24M', 'Varies with device', '9.4M', '15M', '10M', '1.2M', '26M',
       '8.0M', '7.9M', '56M', '57M', '35M', '54M', '201k', '3.6M', '5.7M',
       '8.6M', '2.4M', '27M', '2.7M', '2.5M', '7.0M', '16M', '3.4M',
       '8.9M', '3.9M', '2.9M', '38M', '32M', '5.4M', '18M', '1.1M',
       '2.2M', '4.5M', '9.8M', '52M', '9.0M', '6.7M', '30M', '2.6M',
       '7.1M', '22M', '6.4M', '3.2M', '8.2M', '4.9M', '9.5M', '5.0M',
       '5.9M', '13M', '73M', '6.8M', '3.5M', '4.0M', '2.3M', '2.1M',
       '42M', '9.1M', '55M', '23k', '7.3M', '6.5M', '1.5M', '7.5M', '51M',
       '41M', '48M', '8.5M', '46M', '8.3M', '4.3M', '4.7M', '3.3M', '40M',
       '7.8M', '8.8M', '6.6M', '5.1M', '61M', '66M', '79k', '8.4M',
       '3.7M', '118k', '44M', '695k', '1.6M', '6.2M', '53M', '1.4M',
       '3.0M', '7.2M', '5.8M', '3.8M', '9.6M', '45M', '63M', '49M', '77M',
       '4.4M', '70M', '9.3M', '8.1M', '36M', '6.9M', '7.4M', '84M', '97M',
       '2.0M', '1.9M', '1.8M', '5.3M', '47M', '556k', '526k', '76M',
       '7.6M', '59M', '9.7M', '78M', '72M', '43M', '7.7M', '6.3M', '334k',
       '93M', '65M', '79M', '100M', '58M', '50M', '68M', '64M', '34M',
       '67M', '60M', '94M', '9.9M', '232k', '99M', '624k', '95M', '8.5k',
       '41k', '292k', '80M', '1.7M', '10.0M', '74M', '62M', '69M', '75M',
       '98M', '85M', '82M', '96M', '87M', '71M', '86M', '91M', '81M',
       '92M', '83M', '88M', '704k', '862k', '899k', '378k', '4.8M',
       '266k', '375k', '1.3M', '975k', '980k', '4.1M', '89M', '696k',
       '544k', '525k', '920k', '779k', '853k', '720k', '713k', '772k',
       '318k', '58k', '241k', '196k', '857k', '51k', '953k', '865k',
       '251k', '930k', '540k', '313k', '746k', '203k', '26k', '314k',
```

```
       '239k', '371k', '220k', '730k', '756k', '91k', '293k', '17k',
       '74k', '14k', '317k', '78k', '924k', '818k', '81k', '939k', '169k',
       '45k', '965k', '90M', '545k', '61k', '283k', '655k', '714k', '93k',
       '872k', '121k', '322k', '976k', '206k', '954k', '444k', '717k',
       '210k', '609k', '308k', '306k', '175k', '350k', '383k', '454k',
       '1.0M', '70k', '812k', '442k', '842k', '417k', '412k', '459k',
       '478k', '335k', '782k', '721k', '430k', '429k', '192k', '460k',
       '728k', '496k', '816k', '414k', '506k', '887k', '613k', '778k',
       '683k', '592k', '186k', '840k', '647k', '373k', '437k', '598k',
       '716k', '585k', '982k', '219k', '55k', '323k', '691k', '511k',
       '951k', '963k', '25k', '554k', '351k', '27k', '82k', '208k',
       '551k', '29k', '103k', '116k', '153k', '209k', '499k', '173k',
       '597k', '809k', '122k', '411k', '400k', '801k', '787k', '50k',
       '643k', '986k', '516k', '837k', '780k', '20k', '498k', '600k',
       '656k', '221k', '228k', '176k', '34k', '259k', '164k', '458k',
       '629k', '28k', '288k', '775k', '785k', '636k', '916k', '994k',
       '309k', '485k', '914k', '903k', '608k', '500k', '54k', '562k',
       '847k', '948k', '811k', '270k', '48k', '523k', '784k', '280k',
       '24k', '892k', '154k', '18k', '33k', '860k', '364k', '387k',
       '626k', '161k', '879k', '39k', '170k', '141k', '160k', '144k',
       '143k', '190k', '376k', '193k', '473k', '246k', '73k', '253k',
       '957k', '420k', '72k', '404k', '470k', '226k', '240k', '89k',
       '234k', '257k', '861k', '467k', '676k', '552k', '582k', '619k'],
      dtype=object)
```

Observations:

Varies with device

M

k

[ ]:
```
df[df['Size'] == 'Varies with device'].shape
```

[ ]:
```
(1637, 13)
```

[ ]:
```
df['Size'].isnull().sum()
```

[ ]:
```
0
```

[ ]:
```

```

There are no missing values in size column

Finding the values having M in them

[ ]:
```
df['Size'].loc[df['Size'].str.contains('M')].value_counts().sum()
```

[ ]:
```
7466
```

[ ]:

Finding the values having k in them

[ ]:
```
df['Size'].loc[df['Size'].str.contains('k')].value_counts().sum()
```

[ ]: 257

Finding the values having Varies with device in them

[ ]:
```
df['Size'].loc[df['Size'].str.contains('Varies with device')].value_counts().
↪sum()
```

[ ]: 1637

Converting the whole size column into bytes

[ ]:
```
def convert_size_to_bytes(size_str):
    if size_str == 'Varies with device':
        return np.nan
    elif size_str.endswith('M'):
        return float(size_str[:-1]) * 1024 * 1024
    elif size_str.endswith('k'):
        return float(size_str[:-1]) * 1024
    elif size_str.endswith('k'):
        return float(size_str[:-1])
    else:
        return np.nan
```

[ ]:
```
df['Size'] = df['Size'].apply(convert_size_to_bytes)
```

```
<ipython-input-129-67a61535978a>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Size'] = df['Size'].apply(convert_size_to_bytes)
```

[ ]:
```
df.rename(columns = {'Size': 'Size_in_Bytes'}, inplace=True)
```

```
<ipython-input-130-b5c3f86cfd53>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.rename(columns = {'Size': 'Size_in_Bytes'}, inplace=True)
```

```
[ ]: df['Size_in_Bytes'].dtype
```

```
[ ]: dtype('float64')
```

```
[ ]: df['Size_in_Bytes'].unique()
```

```
[ ]: array([1.99229440e+07, 1.46800640e+07, 9.12261120e+06, 2.62144000e+07,
            2.93601280e+06, 5.87202560e+06, 3.04087040e+07, 3.46030080e+07,
            3.25058560e+06, 2.93601280e+07, 1.25829120e+07, 2.09715200e+07,
            2.20200960e+07, 3.87973120e+07, 5.76716800e+06, 1.78257920e+07,
            4.08944640e+07, 3.25058560e+07, 4.40401920e+06, 2.41172480e+07,
            6.29145600e+06, 6.39631360e+06, 4.82344960e+06, 9.64689920e+06,
            5.45259520e+06, 1.15343360e+07, 2.51658240e+07,             nan,
            9.85661440e+06, 1.57286400e+07, 1.04857600e+07, 1.25829120e+06,
            2.72629760e+07, 8.38860800e+06, 8.28375040e+06, 5.87202560e+07,
            5.97688320e+07, 3.67001600e+07, 5.66231040e+07, 2.05824000e+05,
            3.77487360e+06, 5.97688320e+06, 9.01775360e+06, 2.51658240e+06,
            2.83115520e+07, 2.83115520e+06, 2.62144000e+06, 7.34003200e+06,
            1.67772160e+07, 3.56515840e+06, 9.33232640e+06, 4.08944640e+06,
            3.04087040e+06, 3.98458880e+07, 3.35544320e+07, 5.66231040e+06,
            1.88743680e+07, 1.15343360e+06, 2.30686720e+06, 4.71859200e+06,
            1.02760448e+07, 5.45259520e+07, 9.43718400e+06, 7.02545920e+06,
            3.14572800e+07, 2.72629760e+06, 7.44488960e+06, 2.30686720e+07,
            6.71088640e+06, 3.35544320e+06, 8.59832320e+06, 5.13802240e+06,
            9.96147200e+06, 5.24288000e+06, 6.18659840e+06, 1.36314880e+07,
            7.65460480e+07, 7.13031680e+06, 3.67001600e+06, 4.19430400e+06,
            2.41172480e+06, 2.20200960e+06, 4.40401920e+07, 9.54204160e+06,
            5.76716800e+07, 2.35520000e+04, 7.65460480e+06, 6.81574400e+06,
            1.57286400e+06, 7.86432000e+06, 5.34773760e+07, 4.29916160e+07,
            5.03316480e+07, 8.91289600e+06, 4.82344960e+07, 8.70318080e+06,
            4.50887680e+06, 4.92830720e+06, 3.46030080e+06, 4.19430400e+07,
            8.17889280e+06, 9.22746880e+06, 6.92060160e+06, 5.34773760e+06,
            6.39631360e+07, 6.92060160e+07, 8.08960000e+04, 8.80803840e+06,
            3.87973120e+06, 1.20832000e+05, 4.61373440e+07, 7.11680000e+05,
            1.67772160e+06, 6.50117120e+06, 5.55745280e+07, 1.46800640e+06,
            3.14572800e+06, 7.54974720e+06, 6.08174080e+06, 3.98458880e+06,
            1.00663296e+07, 4.71859200e+07, 6.60602880e+07, 5.13802240e+07,
            8.07403520e+07, 4.61373440e+06, 7.34003200e+07, 9.75175680e+06,
            8.49346560e+06, 3.77487360e+07, 7.23517440e+06, 7.75946240e+06,
            8.80803840e+07, 1.01711872e+08, 2.09715200e+06, 1.99229440e+06,
            1.88743680e+06, 5.55745280e+06, 4.92830720e+07, 5.69344000e+05,
            5.38624000e+05, 7.96917760e+07, 7.96917760e+06, 6.18659840e+07,
            1.01711872e+07, 8.17889280e+07, 7.54974720e+07, 4.50887680e+07,
            8.07403520e+06, 6.60602880e+06, 3.42016000e+05, 9.75175680e+07,
            6.81574400e+07, 8.28375040e+07, 1.04857600e+08, 6.08174080e+07,
            5.24288000e+07, 7.13031680e+07, 6.71088640e+07, 3.56515840e+07,
            7.02545920e+07, 6.29145600e+07, 9.85661440e+07, 1.03809024e+07,
```

2.37568000e+05, 1.03809024e+08, 6.38976000e+05, 9.96147200e+07,
8.70400000e+03, 4.19840000e+04, 2.99008000e+05, 8.38860800e+07,
1.78257920e+06, 7.75946240e+07, 6.50117120e+07, 7.23517440e+07,
7.86432000e+07, 1.02760448e+08, 8.91289600e+07, 8.59832320e+07,
1.00663296e+08, 9.12261120e+07, 7.44488960e+07, 9.01775360e+07,
9.54204160e+07, 8.49346560e+07, 9.64689920e+07, 8.70318080e+07,
9.22746880e+07, 7.20896000e+05, 8.82688000e+05, 9.20576000e+05,
3.87072000e+05, 5.03316480e+06, 2.72384000e+05, 3.84000000e+05,
1.36314880e+06, 9.98400000e+05, 1.00352000e+06, 4.29916160e+06,
9.33232640e+07, 7.12704000e+05, 5.57056000e+05, 5.37600000e+05,
9.42080000e+05, 7.97696000e+05, 8.73472000e+05, 7.37280000e+05,
7.30112000e+05, 7.90528000e+05, 3.25632000e+05, 5.93920000e+04,
2.46784000e+05, 2.00704000e+05, 8.77568000e+05, 5.22240000e+04,
9.75872000e+05, 8.85760000e+05, 2.57024000e+05, 9.52320000e+05,
5.52960000e+05, 3.20512000e+05, 7.63904000e+05, 2.07872000e+05,
2.66240000e+04, 3.21536000e+05, 2.44736000e+05, 3.79904000e+05,
2.25280000e+05, 7.47520000e+05, 7.74144000e+05, 9.31840000e+04,
3.00032000e+05, 1.74080000e+04, 7.57760000e+04, 1.43360000e+04,
3.24608000e+05, 7.98720000e+04, 9.46176000e+05, 8.37632000e+05,
8.29440000e+04, 9.61536000e+05, 1.73056000e+05, 4.60800000e+04,
9.88160000e+05, 9.43718400e+07, 5.58080000e+05, 6.24640000e+04,
2.89792000e+05, 6.70720000e+05, 7.31136000e+05, 9.52320000e+04,
8.92928000e+05, 1.23904000e+05, 3.29728000e+05, 9.99424000e+05,
2.10944000e+05, 9.76896000e+05, 4.54656000e+05, 7.34208000e+05,
2.15040000e+05, 6.23616000e+05, 3.15392000e+05, 3.13344000e+05,
1.79200000e+05, 3.58400000e+05, 3.92192000e+05, 4.64896000e+05,
1.04857600e+06, 7.16800000e+04, 8.31488000e+05, 4.52608000e+05,
8.62208000e+05, 4.27008000e+05, 4.21888000e+05, 4.70016000e+05,
4.89472000e+05, 3.43040000e+05, 8.00768000e+05, 7.38304000e+05,
4.40320000e+05, 4.39296000e+05, 1.96608000e+05, 4.71040000e+05,
7.45472000e+05, 5.07904000e+05, 8.35584000e+05, 4.23936000e+05,
5.18144000e+05, 9.08288000e+05, 6.27712000e+05, 7.96672000e+05,
6.99392000e+05, 6.06208000e+05, 1.90464000e+05, 8.60160000e+05,
6.62528000e+05, 3.81952000e+05, 4.47488000e+05, 6.12352000e+05,
7.33184000e+05, 5.99040000e+05, 1.00556800e+06, 2.24256000e+05,
5.63200000e+04, 3.30752000e+05, 7.07584000e+05, 5.23264000e+05,
9.73824000e+05, 9.86112000e+05, 2.56000000e+04, 5.67296000e+05,
3.59424000e+05, 2.76480000e+04, 8.39680000e+04, 2.12992000e+05,
5.64224000e+05, 2.96960000e+04, 1.05472000e+05, 1.18784000e+05,
1.56672000e+05, 2.14016000e+05, 5.10976000e+05, 1.77152000e+05,
6.11328000e+05, 8.28416000e+05, 1.24928000e+05, 4.20864000e+05,
4.09600000e+05, 8.20224000e+05, 8.05888000e+05, 5.12000000e+04,
6.58432000e+05, 1.00966400e+06, 5.28384000e+05, 8.57088000e+05,
7.98720000e+05, 2.04800000e+04, 5.09952000e+05, 6.14400000e+05,
6.71744000e+05, 2.26304000e+05, 2.33472000e+05, 1.80224000e+05,
3.48160000e+04, 2.65216000e+05, 1.67936000e+05, 4.68992000e+05,
6.44096000e+05, 2.86720000e+04, 2.94912000e+05, 7.93600000e+05,

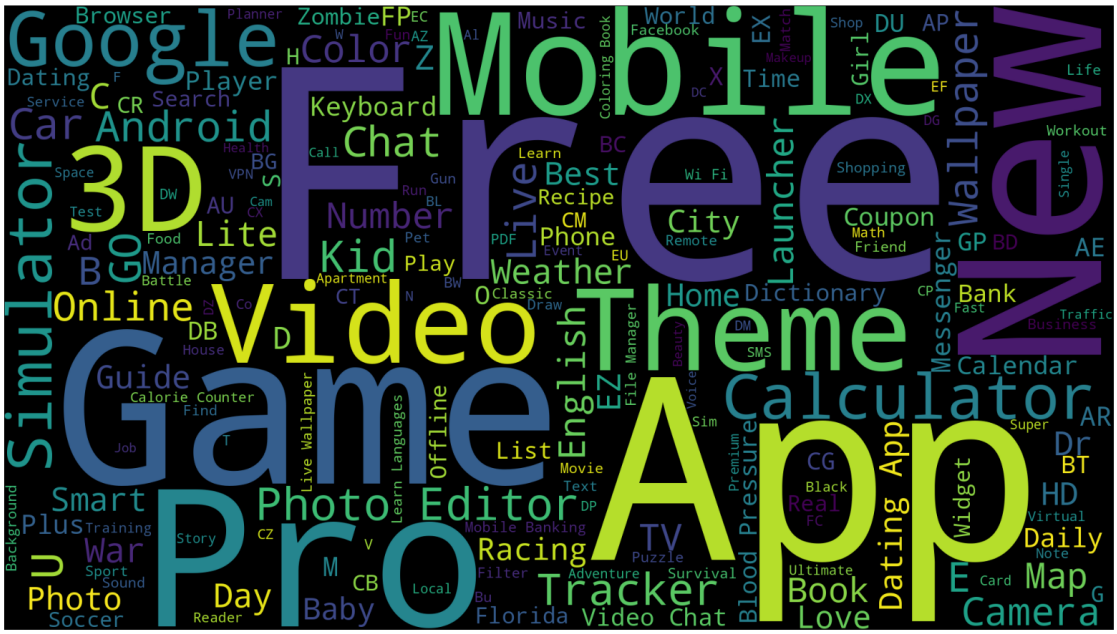
```
       8.03840000e+05, 6.51264000e+05, 9.37984000e+05, 1.01785600e+06,
       3.16416000e+05, 4.96640000e+05, 9.35936000e+05, 9.24672000e+05,
       6.22592000e+05, 5.12000000e+05, 5.52960000e+04, 5.75488000e+05,
       8.67328000e+05, 9.70752000e+05, 8.30464000e+05, 2.76480000e+05,
       4.91520000e+04, 5.35552000e+05, 8.02816000e+05, 2.86720000e+05,
       2.45760000e+04, 9.13408000e+05, 1.57696000e+05, 1.84320000e+04,
       3.37920000e+04, 8.80640000e+05, 3.72736000e+05, 3.96288000e+05,
       6.41024000e+05, 1.64864000e+05, 9.00096000e+05, 3.99360000e+04,
       1.74080000e+05, 1.44384000e+05, 1.63840000e+05, 1.47456000e+05,
       1.46432000e+05, 1.94560000e+05, 3.85024000e+05, 1.97632000e+05,
       4.84352000e+05, 2.51904000e+05, 7.47520000e+04, 2.59072000e+05,
       9.79968000e+05, 4.30080000e+05, 7.37280000e+04, 4.13696000e+05,
       4.81280000e+05, 2.31424000e+05, 2.45760000e+05, 9.11360000e+04,
       2.39616000e+05, 2.63168000e+05, 8.81664000e+05, 4.78208000e+05,
       6.92224000e+05, 5.65248000e+05, 5.95968000e+05, 6.33856000e+05])
```

[ ]:
```
df.head(10)
```

[ ]:
```
                                                 App        Category  Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
1                                 Coloring book moana  ART_AND_DESIGN     3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide …  ART_AND_DESIGN     4.7
3                               Sketch - Draw & Paint  ART_AND_DESIGN     4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3
5                         Paper flowers instructions  ART_AND_DESIGN     4.4
6            Smoke Effect Photo Maker - Smoke Editor  ART_AND_DESIGN     3.8
7                                   Infinite Painter  ART_AND_DESIGN     4.1
8                               Garden Coloring Book  ART_AND_DESIGN     4.4
9                     Kids Paint Free - Drawing Fun  ART_AND_DESIGN     4.7

  Reviews  Size_in_Bytes  Installs  Type  Price Content Rating  \
0     159     19922944.0     10000  Free    0.0       Everyone
1     967     14680064.0    500000  Free    0.0       Everyone
2   87510      9122611.2   5000000  Free    0.0       Everyone
3  215644     26214400.0  50000000  Free    0.0           Teen
4     967      2936012.8    100000  Free    0.0       Everyone
5     167      5872025.6     50000  Free    0.0       Everyone
6     178     19922944.0     50000  Free    0.0       Everyone
7   36815     30408704.0   1000000  Free    0.0       Everyone
8   13791     34603008.0   1000000  Free    0.0       Everyone
9     121      3250585.6     10000  Free    0.0       Everyone

                     Genres        Last Updated         Current Ver  \
0              Art & Design     January 7, 2018               1.0.0
1  Art & Design;Pretend Play    January 15, 2018               2.0.0
2              Art & Design      August 1, 2018               1.2.4
3              Art & Design        June 8, 2018  Varies with device
```

```
4    Art & Design;Creativity       June 20, 2018                1.1
5                Art & Design      March 26, 2017                1.0
6                Art & Design      April 26, 2018                1.1
7                Art & Design       June 14, 2018            6.1.61.1
8                Art & Design  September 20, 2017                2.9.2
9    Art & Design;Creativity        July 3, 2018                2.8

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
5    2.3 and up
6  4.0.3 and up
7    4.2 and up
8    3.0 and up
9  4.0.3 and up
```

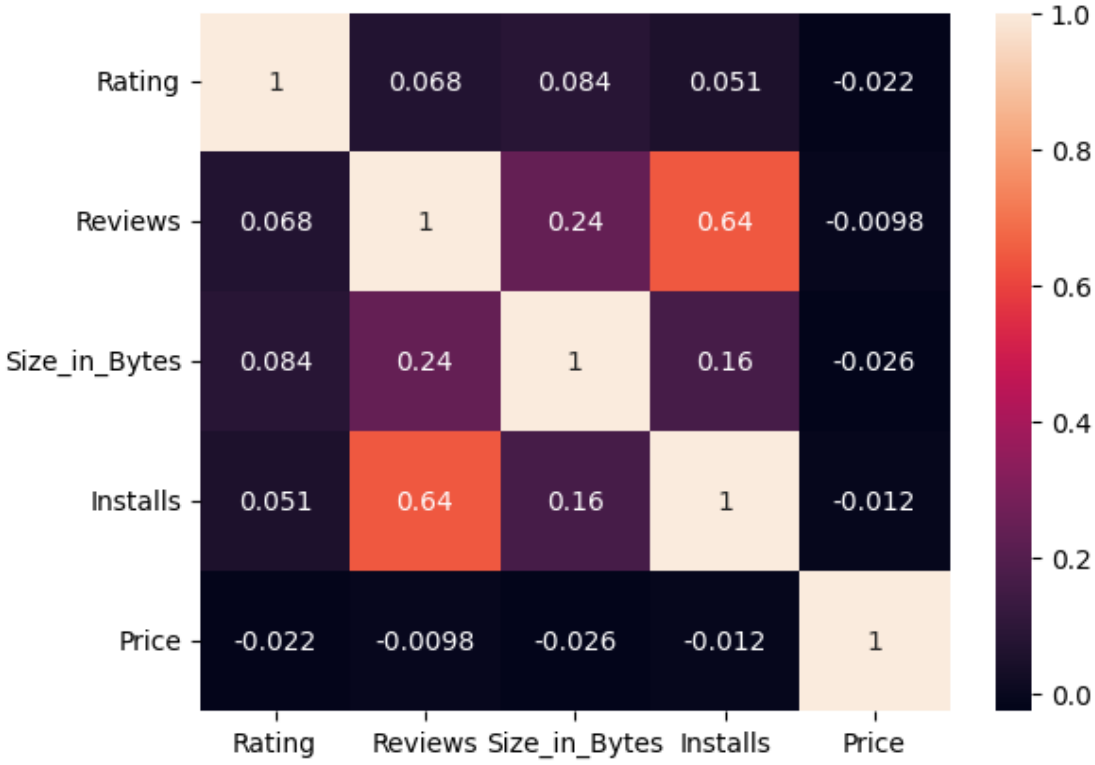
[ ]:
```
plt.subplots(figsize=(20,12))
wordcloud = WordCloud(
                          background_color='black',
                          width=1920,
                          height=1080
                         ).generate(" ".join(df.App))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



7 Correlation Analysis

[ ]:
```
cols = ['Rating','Reviews','Size_in_Bytes','Installs','Price']
sns.heatmap(df[cols].corr(), annot = True)
```

[ ]: <Axes: >

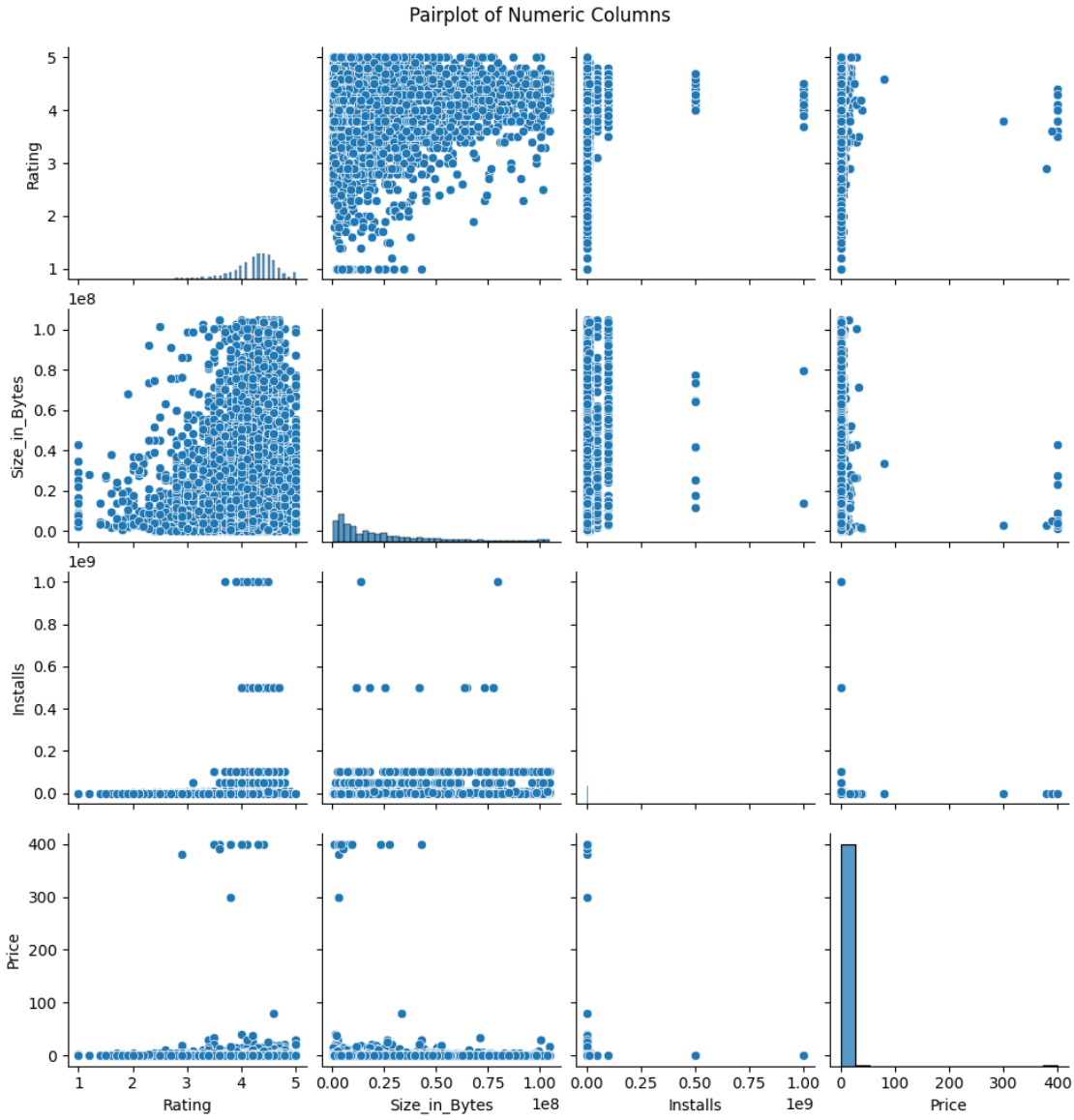


From the above Matrix, We can see that the reviews column has the higher correlation with Installs

[ ]:
```
# Convert columns to numeric if applicable
numeric_columns = df.select_dtypes(include=['int64', 'float64']).columns
df[numeric_columns] = df[numeric_columns].apply(pd.to_numeric, errors='coerce')

# Remove non-numeric columns
df_numeric = df[numeric_columns]

# Plot pairplot
sns.pairplot(df_numeric)
plt.suptitle('Pairplot of Numeric Columns', y=1.02)
```

```
plt.show()
```

```
<ipython-input-170-54627e8e7e85>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[numeric_columns] = df[numeric_columns].apply(pd.to_numeric,
errors='coerce')
```



8 1) How many different kind of apps are there in the playstore ?

[ ]:
```
Category = df['Category'].value_counts()
print(Category)
```

```
Category
FAMILY                 1746
GAME                   1097
TOOLS                   733
PRODUCTIVITY            351
MEDICAL                 350
COMMUNICATION           328
FINANCE                 323
SPORTS                  319
PHOTOGRAPHY             317
LIFESTYLE               314
PERSONALIZATION         312
BUSINESS                303
HEALTH_AND_FITNESS      297
SOCIAL                  259
SHOPPING                238
NEWS_AND_MAGAZINES      233
TRAVEL_AND_LOCAL        226
DATING                  195
BOOKS_AND_REFERENCE     178
VIDEO_PLAYERS           160
EDUCATION               155
ENTERTAINMENT           149
MAPS_AND_NAVIGATION     124
FOOD_AND_DRINK          109
HOUSE_AND_HOME           76
WEATHER                  75
AUTO_AND_VEHICLES        73
LIBRARIES_AND_DEMO       64
ART_AND_DESIGN           61
COMICS                   58
PARENTING                50
EVENTS                   45
BEAUTY                   42
Name: count, dtype: int64
```

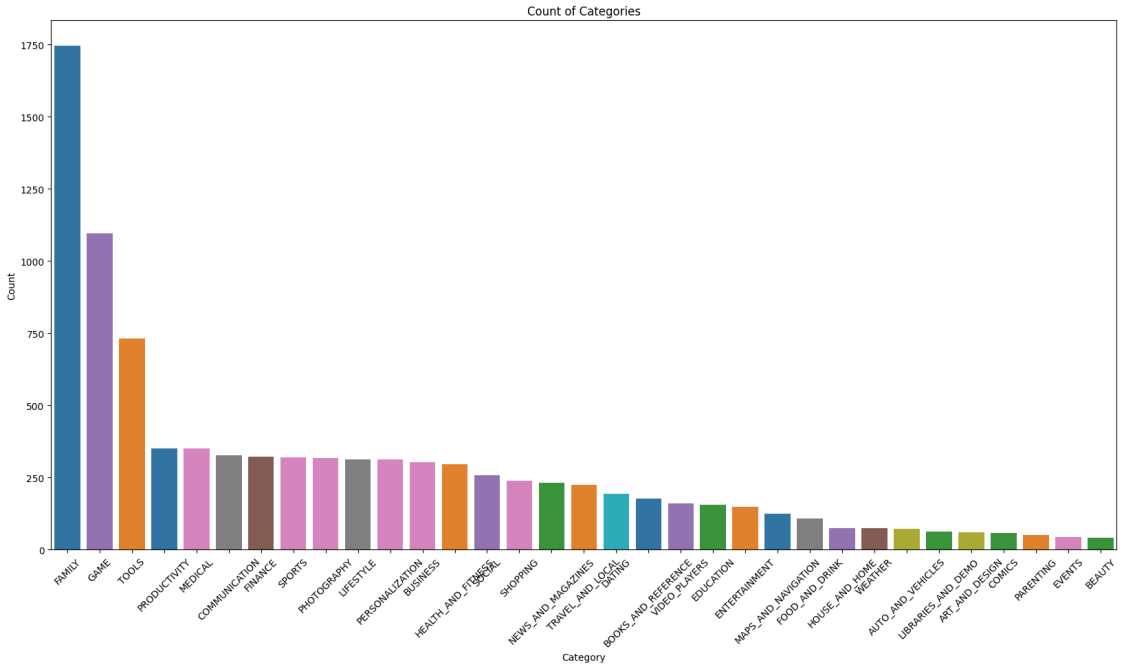
Plotting the number of different categories of application of the Google play store.

Multicolored plotting using Random

[ ]:
```
Category_plotting = df['Category'].value_counts()

# Generate random colors for each category
```

```
num_categories = len(Category_plotting)
colors = [plt.cm.tab10(random.randint(0, 9)) for _ in range(num_categories)]

# Plotting
plt.figure(figsize=(20, 10))  # Adjust the figure size as needed
sns.barplot(x=Category_plotting.index, y=Category_plotting.values,␣
 ↪palette=colors)
plt.title('Count of Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```

```
<ipython-input-135-a7c216e8aef5>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=Category_plotting.index, y=Category_plotting.values,
palette=colors)
```

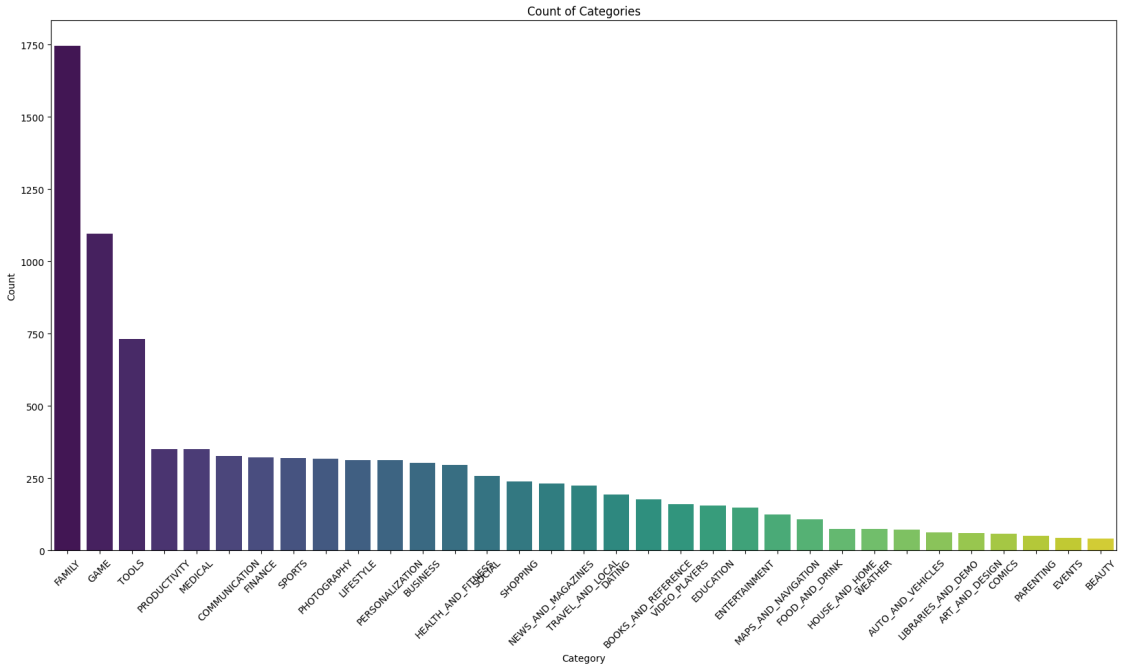


Muticolored plotting using Seaborn.

[ ]:
```
Category_plotting = df['Category'].value_counts()
```

```
# Generate a list of unique colors based on the number of categories
num_categories = len(Category_plotting)
colors = sns.color_palette('viridis', n_colors=num_categories)

# Plotting
plt.figure(figsize=(20, 10))  # Adjust the figure size as needed
sns.barplot(x=Category_plotting.index, y=Category_plotting.values,␣
 ↪palette=colors)
plt.title('Count of Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```

```
<ipython-input-136-c96ae39bcb66>:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=Category_plotting.index, y=Category_plotting.values,
palette=colors)
```



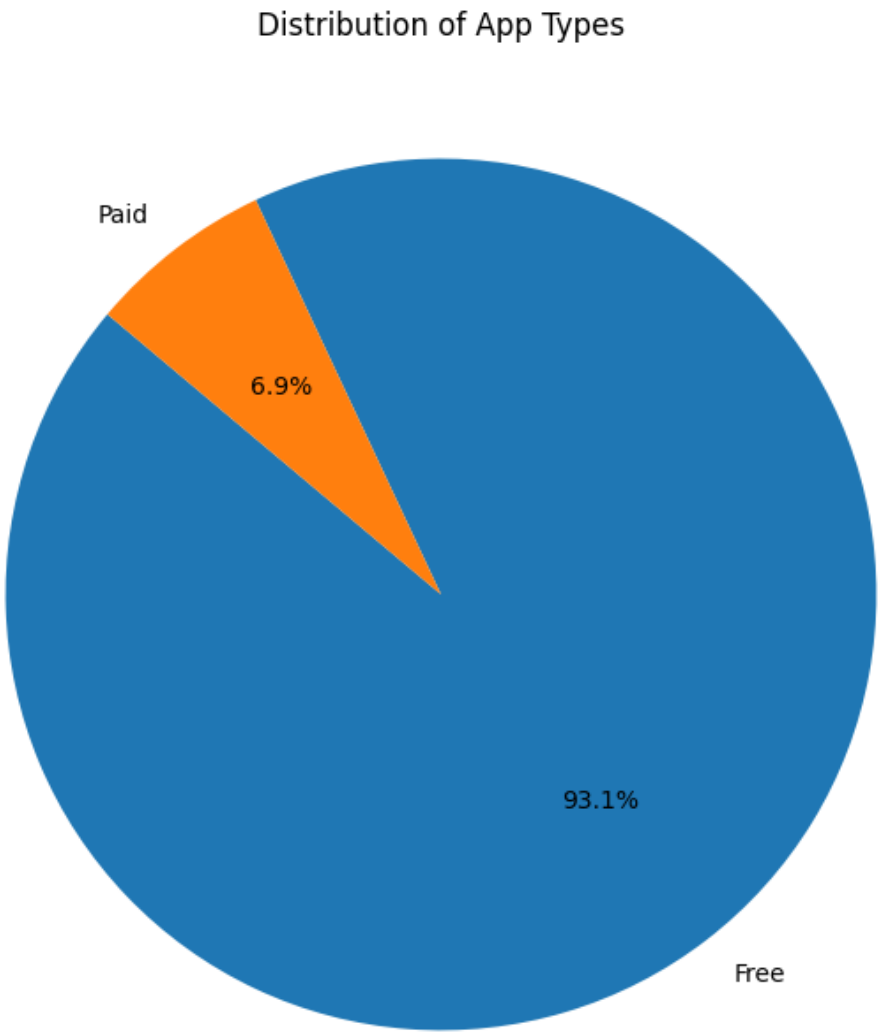
[ ]:

9 2) How many apps are free and how many apps are pain on Playstore

[ ]:
```
Type_plotting = df['Type'].value_counts()
print(Type_plotting)
```

```
Type
Free    8715
Paid     645
Name: count, dtype: int64
```

[ ]:
```
Type_plotting = df['Type'].value_counts()
# Plotting
plt.figure(figsize=(8, 8))  # Adjust the figure size as needed
plt.pie(Type_plotting.values, labels=Type_plotting.index, autopct='%1.1f%%',␣
 ↪startangle=140)
plt.title('Distribution of App Types')
plt.show()
```

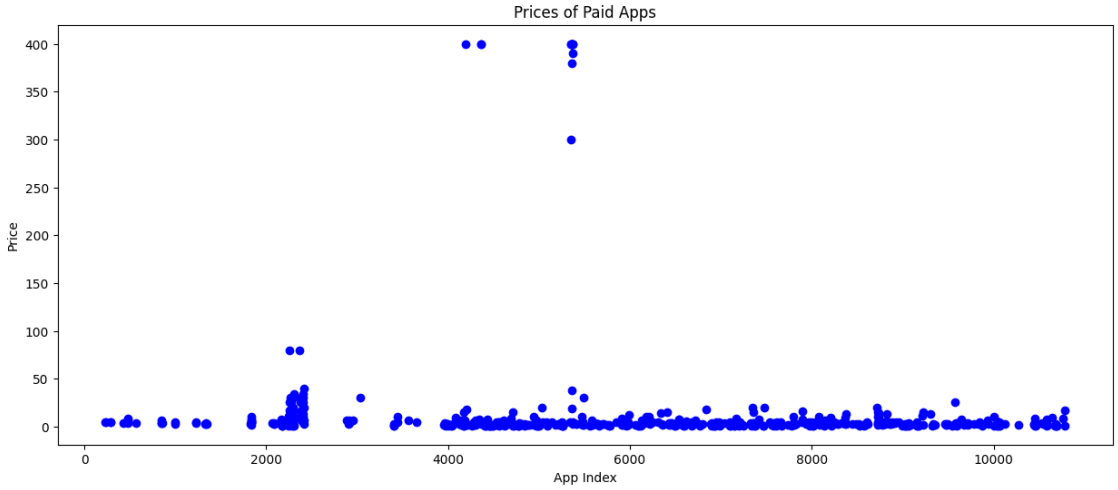


[ ]:
```
# Filter out only the paid apps
paid_apps = df[df['Type'] == 'Paid']

# Display the filtered DataFrame with the prices of paid apps
paid_apps_with_prices = paid_apps[['App', 'Price']]
print(paid_apps_with_prices)
```

```
                                                App  Price
234    TurboScan: scan documents and receipts in PDF   4.99
235                  Tiny Scanner Pro: PDF Doc Scan    4.99
290    TurboScan: scan documents and receipts in PDF   4.99
291                  Tiny Scanner Pro: PDF Doc Scan    4.99
```

```
427                                Puffin Browser Pro   3.99
…                                                …    …
10690                                      FO Bixby   0.99
10697                                       Mu.F.O.   0.99
10760                                 Fast Tract Diet   7.99
10782                   Trine 2: Complete Story  16.99
10785                                 sugar, sugar   1.20

[645 rows x 2 columns]
```

[ ]:
```
plt.figure(figsize=(15, 6))
plt.scatter(paid_apps_with_prices.index, paid_apps_with_prices['Price'],␣
↪color='blue')
plt.title('Prices of Paid Apps')
plt.xlabel('App Index')
plt.ylabel('Price')
plt.show()
```



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

[ ]:
```
                                                 App        Category  Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
1                                 Coloring book moana  ART_AND_DESIGN     3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide …  ART_AND_DESIGN     4.7
3                               Sketch - Draw & Paint  ART_AND_DESIGN     4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

  Reviews  Size_in_Bytes  Installs  Type  Price Content Rating  \
0     159     19922944.0     10000  Free    0.0       Everyone
1     967     14680064.0    500000  Free    0.0       Everyone
```

```
2   87510      9122611.2   5000000  Free    0.0       Everyone
3  215644     26214400.0  50000000  Free    0.0           Teen
4     967      2936012.8    100000  Free    0.0       Everyone

                      Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    Art & Design;Creativity     June 20, 2018                 1.1

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

10 3) Define the Genre of different application available on the Google Playstore

[ ]:
```
Genres=df['Genres'].unique()
Genres
```

[ ]:
```
array(['Art & Design', 'Art & Design;Pretend Play',
       'Art & Design;Creativity', 'Auto & Vehicles', 'Beauty',
       'Books & Reference', 'Business', 'Comics', 'Comics;Creativity',
       'Communication', 'Dating', 'Education;Education', 'Education',
       'Education;Creativity', 'Education;Music & Video',
       'Education;Action & Adventure', 'Education;Pretend Play',
       'Education;Brain Games', 'Entertainment',
       'Entertainment;Music & Video', 'Entertainment;Brain Games',
       'Entertainment;Creativity', 'Events', 'Finance', 'Food & Drink',
       'Health & Fitness', 'House & Home', 'Libraries & Demo',
       'Lifestyle', 'Lifestyle;Pretend Play',
       'Adventure;Action & Adventure', 'Arcade', 'Casual', 'Card',
       'Casual;Pretend Play', 'Action', 'Strategy', 'Puzzle', 'Sports',
       'Music', 'Word', 'Racing', 'Casual;Creativity',
       'Casual;Action & Adventure', 'Simulation', 'Adventure', 'Board',
       'Trivia', 'Role Playing', 'Simulation;Education',
       'Action;Action & Adventure', 'Casual;Brain Games',
       'Simulation;Action & Adventure', 'Educational;Creativity',
       'Puzzle;Brain Games', 'Educational;Education', 'Card;Brain Games',
       'Educational;Brain Games', 'Educational;Pretend Play',
       'Entertainment;Education', 'Casual;Education',
       'Music;Music & Video', 'Racing;Action & Adventure',
```

```
        'Arcade;Pretend Play', 'Role Playing;Action & Adventure',
        'Simulation;Pretend Play', 'Puzzle;Creativity',
        'Sports;Action & Adventure', 'Educational;Action & Adventure',
        'Arcade;Action & Adventure', 'Entertainment;Action & Adventure',
        'Puzzle;Action & Adventure', 'Strategy;Action & Adventure',
        'Music & Audio;Music & Video', 'Health & Fitness;Education',
        'Adventure;Education', 'Board;Brain Games',
        'Board;Action & Adventure', 'Board;Pretend Play',
        'Casual;Music & Video', 'Role Playing;Pretend Play',
        'Entertainment;Pretend Play', 'Video Players & Editors;Creativity',
        'Card;Action & Adventure', 'Medical', 'Social', 'Shopping',
        'Photography', 'Travel & Local',
        'Travel & Local;Action & Adventure', 'Tools', 'Tools;Education',
        'Personalization', 'Productivity', 'Parenting',
        'Parenting;Music & Video', 'Parenting;Brain Games',
        'Parenting;Education', 'Weather', 'Video Players & Editors',
        'Video Players & Editors;Music & Video', 'News & Magazines',
        'Maps & Navigation', 'Health & Fitness;Action & Adventure',
        'Educational', 'Casino', 'Adventure;Brain Games',
        'Lifestyle;Education', 'Books & Reference;Education',
        'Puzzle;Education', 'Role Playing;Brain Games',
        'Strategy;Education', 'Racing;Pretend Play',
        'Communication;Creativity', 'Strategy;Creativity'], dtype=object)
```

11 4) Display the app count of each Genre .

[ ]:
```
# Assuming df is your DataFrame and 'Genres' is the name of the column

# Group by 'Genres' column and count occurrences
genre_count = df.groupby('Genres')['Genres'].agg({'count'}).reset_index()

# Display the counts of each genre
print(genre_count)
```

```
                                  Genres  count
0                                 Action    358
1              Action;Action & Adventure     17
2                              Adventure     73
3           Adventure;Action & Adventure     13
4                 Adventure;Brain Games      1
..                                  …    …
110              Video Players & Editors    158
111    Video Players & Editors;Creativity      2
112  Video Players & Editors;Music & Video      3
113                             Weather     75
114                                Word     28
```

[115 rows x 2 columns]

[ ]:
```
# Assuming df is your DataFrame and 'Genres' is the name of the column

# Group by 'Genres' column and count occurrences
genre_count = df.groupby('Genres')['Genres'].agg({'count'}).reset_index()

# Add an index starting from 1
genre_count.index = genre_count.index + 1
genre_count.index.name = 'Sl.No'

# Split the DataFrame into two equal halves
half_length = len(genre_count) // 2
first_half = genre_count.iloc[:half_length]
second_half = genre_count.iloc[half_length:]

# Convert halves to tabular format
first_half_tabular = tabulate(first_half, headers='keys', tablefmt='psql')
second_half_tabular = tabulate(second_half, headers='keys', tablefmt='psql')

# Print the aligned output
print(tabulate([[first_half_tabular, '', second_half_tabular]], headers=['1st␣
 ↪Half', '', '2nd Half'], tablefmt='psql'))
```

```
+-----------------------------------------------------------+----+--------------
-----------------------------------------------+
| 1st Half                                                  |    | 2nd Half
|
|-----------------------------------------------------------+----+--------------
-----------------------------------------------|
| +---------+-----------------------------------+---------+ |    |
+---------+-----------------------------------------+---------+ |
| |   Sl.No | Genres                            |   count | |    | |   Sl.No |
Genres                                |   count | |
| |---------+-----------------------------------+---------| |    |
|---------+-----------------------------------------+---------| |
| |       1 | Action                            |     358 | |    | |      58 |
Events                                |      45 | |
| |       2 | Action;Action & Adventure         |      17 | |    | |      59 |
Finance                               |     323 | |
| |       3 | Adventure                         |      73 | |    | |      60 |
Food & Drink                          |     109 | |
| |       4 | Adventure;Action & Adventure      |      13 | |    | |      61 |
Health & Fitness                      |     297 | |
| |       5 | Adventure;Brain Games             |       1 | |    | |      62 |
Health & Fitness;Action & Adventure   |       1 | |
| |       6 | Adventure;Education               |       2 | |    | |      63 |
```

```
Health & Fitness;Education            |       1 | |
| |       7 | Arcade                    |     207 | |    | |      64 |
House & Home                          |      76 | |
| |       8 | Arcade;Action & Adventure |      15 | |    | |      65 |
Libraries & Demo                      |      64 | |
| |       9 | Arcade;Pretend Play       |       1 | |    | |      66 |
Lifestyle                             |     313 | |
| |      10 | Art & Design              |      55 | |    | |      67 |
Lifestyle;Education                   |       1 | |
| |      11 | Art & Design;Creativity   |       7 | |    | |      68 |
Lifestyle;Pretend Play                |       1 | |
| |      12 | Art & Design;Pretend Play |       2 | |    | |      69 |
Maps & Navigation                     |     124 | |
| |      13 | Auto & Vehicles           |      73 | |    | |      70 |
Medical                               |     350 | |
| |      14 | Beauty                    |      42 | |    | |      71 |
Music                                 |      21 | |
| |      15 | Board                     |      41 | |    | |      72 |
Music & Audio;Music & Video           |       1 | |
| |      16 | Board;Action & Adventure  |       3 | |    | |      73 |
Music;Music & Video                   |       3 | |
| |      17 | Board;Brain Games         |      15 | |    | |      74 |
News & Magazines                      |     233 | |
| |      18 | Board;Pretend Play        |       1 | |    | |      75 |
Parenting                             |      40 | |
| |      19 | Books & Reference         |     178 | |    | |      76 |
Parenting;Brain Games                 |       1 | |
| |      20 | Books & Reference;Education |     2 | |    | |      77 |
Parenting;Education                   |       3 | |
| |      21 | Business                  |     303 | |    | |      78 |
Parenting;Music & Video               |       6 | |
| |      22 | Card                      |      45 | |    | |      79 |
Personalization                       |     312 | |
| |      23 | Card;Action & Adventure   |       2 | |    | |      80 |
Photography                           |     317 | |
| |      24 | Card;Brain Games          |       1 | |    | |      81 |
Productivity                          |     351 | |
| |      25 | Casino                    |      37 | |    | |      82 |
Puzzle                                |     120 | |
| |      26 | Casual                    |     185 | |    | |      83 |
Puzzle;Action & Adventure             |       5 | |
| |      27 | Casual;Action & Adventure |      21 | |    | |      84 |
Puzzle;Brain Games                    |      19 | |
| |      28 | Casual;Brain Games        |      13 | |    | |      85 |
Puzzle;Creativity                     |       2 | |
| |      29 | Casual;Creativity         |       7 | |    | |      86 |
Puzzle;Education                      |       1 | |
| |      30 | Casual;Education          |       3 | |    | |      87 |
```

```
Racing                                |      93 | |
| |      31 | Casual;Music & Video              |       2 | |    | |      88 |
Racing;Action & Adventure             |      20 | |
| |      32 | Casual;Pretend Play               |      31 | |    | |      89 |
Racing;Pretend Play                   |       1 | |
| |      33 | Comics                            |      57 | |    | |      90 |
Role Playing                          |     106 | |
| |      34 | Comics;Creativity                 |       1 | |    | |      91 |
Role Playing;Action & Adventure       |       7 | |
| |      35 | Communication                     |     328 | |    | |      92 |
Role Playing;Brain Games              |       1 | |
| |      36 | Communication;Creativity          |       1 | |    | |      93 |
Role Playing;Pretend Play             |       5 | |
| |      37 | Dating                            |     195 | |    | |      94 |
Shopping                              |     238 | |
| |      38 | Education                         |     468 | |    | |      95 |
Simulation                            |     194 | |
| |      39 | Education;Action & Adventure      |       6 | |    | |      96 |
Simulation;Action & Adventure         |      11 | |
| |      40 | Education;Brain Games             |       4 | |    | |      97 |
Simulation;Education                  |       3 | |
| |      41 | Education;Creativity              |       7 | |    | |      98 |
Simulation;Pretend Play               |       4 | |
| |      42 | Education;Education               |      50 | |    | |      99 |
Social                                |     259 | |
| |      43 | Education;Music & Video           |       5 | |    | |     100 |
Sports                                |     333 | |
| |      44 | Education;Pretend Play            |      23 | |    | |     101 |
Sports;Action & Adventure             |       4 | |
| |      45 | Educational                       |      32 | |    | |     102 |
Strategy                              |     103 | |
| |      46 | Educational;Action & Adventure    |       4 | |    | |     103 |
Strategy;Action & Adventure           |       2 | |
| |      47 | Educational;Brain Games           |       6 | |    | |     104 |
Strategy;Creativity                   |       1 | |
| |      48 | Educational;Creativity            |       5 | |    | |     105 |
Strategy;Education                    |       1 | |
| |      49 | Educational;Education             |      38 | |    | |     106 |
Tools                                 |     732 | |
| |      50 | Educational;Pretend Play          |      18 | |    | |     107 |
Tools;Education                       |       1 | |
| |      51 | Entertainment                     |     533 | |    | |     108 |
Travel & Local                        |     225 | |
| |      52 | Entertainment;Action & Adventure  |       3 | |    | |     109 |
Travel & Local;Action & Adventure     |       1 | |
| |      53 | Entertainment;Brain Games         |       8 | |    | |     110 |
Trivia                                |      28 | |
| |      54 | Entertainment;Creativity          |       3 | |    | |     111 |
```

```
Video Players & Editors                |     158 | |
| |      55 | Entertainment;Education          |       1 | |    | |     112 |
Video Players & Editors;Creativity     |       2 | |
| |      56 | Entertainment;Music & Video      |      27 | |    | |     113 |
Video Players & Editors;Music & Video |       3 | |
| |      57 | Entertainment;Pretend Play       |       2 | |    | |     114 |
Weather                                |      75 | |
| +---------+----------------------------------+---------+ |    | |     115 |
Word                                   |      28 | |
|                                                          |    |
+---------+------------------------------------------+---------+ |
+-----------------------------------------------------------+----+---------------
-----------------------------------------------+
```

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

[ ]:
```
                                                 App        Category  Rating  \
0     Photo Editor & Candy Camera & Grid & ScrapBook  ART_AND_DESIGN     4.1
1                                 Coloring book moana  ART_AND_DESIGN     3.9
2  U Launcher Lite - FREE Live Cool Themes, Hide …  ART_AND_DESIGN     4.7
3                               Sketch - Draw & Paint  ART_AND_DESIGN     4.5
4              Pixel Draw - Number Art Coloring Book  ART_AND_DESIGN     4.3

  Reviews  Size_in_Bytes  Installs  Type  Price Content Rating  \
0     159     19922944.0     10000  Free    0.0       Everyone
1     967     14680064.0    500000  Free    0.0       Everyone
2   87510      9122611.2   5000000  Free    0.0       Everyone
3  215644     26214400.0  50000000  Free    0.0           Teen
4     967      2936012.8    100000  Free    0.0       Everyone

                      Genres      Last Updated         Current Ver  \
0               Art & Design   January 7, 2018               1.0.0
1  Art & Design;Pretend Play  January 15, 2018               2.0.0
2               Art & Design    August 1, 2018               1.2.4
3               Art & Design      June 8, 2018  Varies with device
4    Art & Design;Creativity     June 20, 2018                 1.1

    Android Ver
0  4.0.3 and up
1  4.0.3 and up
2  4.0.3 and up
3    4.2 and up
4    4.4 and up
```

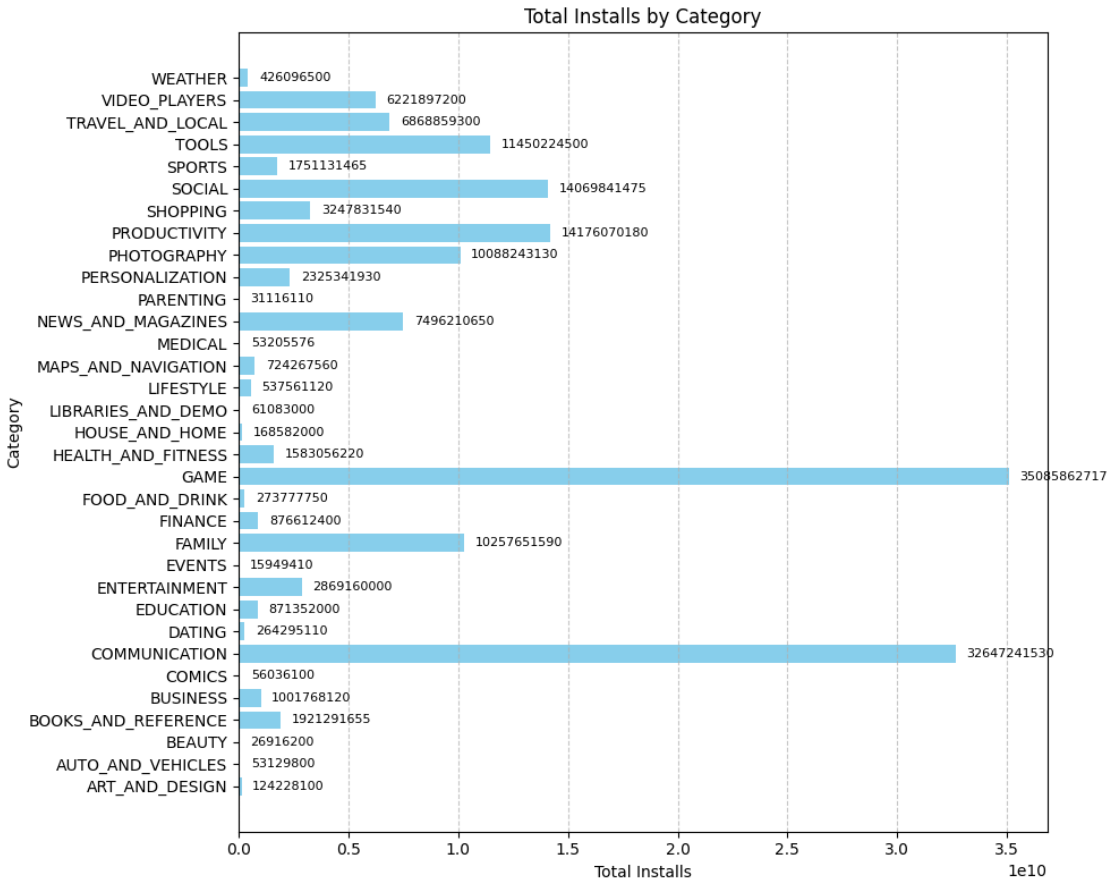
12 5) Which category have how many downloads ?

[ ]:
```
Categorywise_download = df.groupby('Category')['Installs'].agg('sum').
  ↪reset_index()
print(Categorywise_download)
```

```
              Category     Installs
0        ART_AND_DESIGN    124228100
1     AUTO_AND_VEHICLES     53129800
2               BEAUTY     26916200
3   BOOKS_AND_REFERENCE   1921291655
4             BUSINESS   1001768120
5               COMICS     56036100
6        COMMUNICATION  32647241530
7               DATING    264295110
8            EDUCATION    871352000
9        ENTERTAINMENT   2869160000
10              EVENTS     15949410
11              FAMILY  10257651590
12             FINANCE    876612400
13       FOOD_AND_DRINK    273777750
14                GAME  35085862717
15   HEALTH_AND_FITNESS   1583056220
16       HOUSE_AND_HOME    168582000
17   LIBRARIES_AND_DEMO     61083000
18            LIFESTYLE    537561120
19  MAPS_AND_NAVIGATION    724267560
20              MEDICAL     53205576
21   NEWS_AND_MAGAZINES   7496210650
22            PARENTING     31116110
23      PERSONALIZATION   2325341930
24          PHOTOGRAPHY  10088243130
25         PRODUCTIVITY  14176070180
26             SHOPPING   3247831540
27               SOCIAL  14069841475
28               SPORTS   1751131465
29                TOOLS  11450224500
30     TRAVEL_AND_LOCAL   6868859300
31        VIDEO_PLAYERS   6221897200
32              WEATHER    426096500
```

[ ]:
```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 8))
bars = plt.barh(Categorywise_download['Category'],␣
  ↪Categorywise_download['Installs'], color='skyblue')
```

```
# Add install values directly on the bars
for bar, installs in zip(bars, Categorywise_download['Installs']):
    plt.text(bar.get_width() + 0.5e9, bar.get_y() + bar.get_height()/2,␣
 ↪f'{installs}', va='center', fontsize=8)

plt.title('Total Installs by Category')
plt.xlabel('Total Installs')
plt.ylabel('Category')
plt.grid(axis='x', linestyle='--', alpha=0.7)  # Add grid lines for better␣
 ↪readability
plt.tight_layout()  # Ensure labels do not overlap
plt.show()
```



[ ]:

Dropping the unwanted Columns

[ ]:
```
df.drop("Current Ver", axis=1, inplace=True)
df.drop("Android Ver", axis=1, inplace=True)
```

```
<ipython-input-149-d278cd5e372c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop("Current Ver", axis=1, inplace=True)
<ipython-input-149-d278cd5e372c>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop("Android Ver", axis=1, inplace=True)
```

[ ]:
```
df.drop("Last Updated", axis=1, inplace=True)
```

```
<ipython-input-150-864a03fee9a5>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop("Last Updated", axis=1, inplace=True)
```

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

[ ]:

	App	Category	Rating	Reviews	Size_in_Bytes	Installs	Type	Price	Content Rating	Genres
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19922944.0	10000	Free	0.0	Everyone	Art & Design
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14680064.0	500000	Free	0.0	Everyone	Art & Design;Pretend Play
2	U Launcher Lite - FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	87510	9122611.2	5000000	Free	0.0	Everyone	Art & Design
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	26214400.0	50000000	Free	0.0	Teen	Art & Design
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2936012.8	100000	Free	0.0	Everyone	Art & Design;Creativity

13 6) Find the of the row with the maximum number of reviews

[ ]:
```
# Convert the "Reviews" column to numeric data type
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')

# Find the index of the row with the maximum number of reviews
max_review_index = df['Rating'].idxmax()

# Get the corresponding row
app_with_max_reviews = df.loc[max_review_index]

print("App with Maximum Rating:")
print(app_with_max_reviews)
```

```
App with Maximum Rating:
App                Hojiboy Tojiboyev Life Hacks
Category                                COMICS
Rating                                     5.0
Reviews                                     15
Size_in_Bytes                         38797312.0
Installs                                  1000
Type                                      Free
Price                                      0.0
Content Rating                         Everyone
Genres                                   Comics
Name: 329, dtype: object
```

```
<ipython-input-152-bca905836e19>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')
```

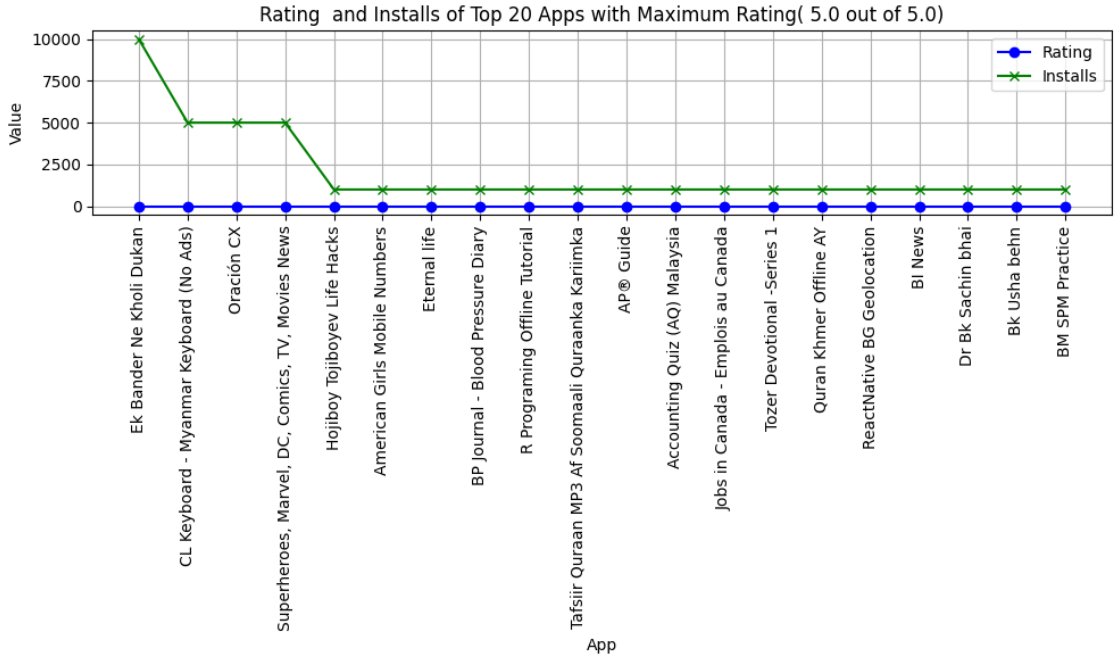
14 7) Find top 20 max rated apps which have maximum number of installation

[ ]:
```
max_rating = df['Rating'].max()
max_rating_apps = df[df['Rating'] == max_rating]

top_20_max_rating_apps = max_rating_apps.nlargest(20, 'Installs')

top_20_max_rating_apps.reset_index(drop=True, inplace=True)
top_20_max_rating_apps.index += 1

selected_columns = top_20_max_rating_apps[['App', 'Rating', 'Installs']]
```

```
apps_list = selected_columns.to_dict(orient='records')

print(tabulate(apps_list, headers='keys', tablefmt='psql'))
```

```
+---------------------------------------------------+----------+------------+
| App                                               |   Rating |   Installs |
|---------------------------------------------------+----------+------------|
| Ek Bander Ne Kholi Dukan                          |        5 |      10000 |
| CL Keyboard - Myanmar Keyboard (No Ads)           |        5 |       5000 |
| Oración CX                                        |        5 |       5000 |
| Superheroes, Marvel, DC, Comics, TV, Movies News  |        5 |       5000 |
| Hojiboy Tojiboyev Life Hacks                      |        5 |       1000 |
| American Girls Mobile Numbers                     |        5 |       1000 |
| Eternal life                                      |        5 |       1000 |
| BP Journal - Blood Pressure Diary                 |        5 |       1000 |
| R Programing Offline Tutorial                     |        5 |       1000 |
| Tafsiir Quraan MP3 Af Soomaali Quraanka Kariimka  |        5 |       1000 |
| AP® Guide                                         |        5 |       1000 |
| Accounting Quiz (AQ) Malaysia                     |        5 |       1000 |
| Jobs in Canada - Emplois au Canada                |        5 |       1000 |
| Tozer Devotional -Series 1                        |        5 |       1000 |
| Quran Khmer Offline AY                            |        5 |       1000 |
| ReactNative BG Geolocation                        |        5 |       1000 |
| BI News                                           |        5 |       1000 |
| Dr Bk Sachin bhai                                 |        5 |       1000 |
| Bk Usha behn                                      |        5 |       1000 |
| BM SPM Practice                                   |        5 |       1000 |
+---------------------------------------------------+----------+------------+
```

[ ]:
```
app_names = top_20_max_rating_apps['App']
ratings = top_20_max_rating_apps['Rating']
installs = top_20_max_rating_apps['Installs']

plt.figure(figsize=(10, 6))
plt.plot(app_names, ratings, marker='o', label='Rating', color='blue')
plt.plot(app_names, installs, marker='x', label='Installs', color='green')
plt.xticks(rotation=90)
plt.xlabel('App')
plt.ylabel('Value')
plt.title('Rating  and Installs of Top 20 Apps with Maximum Rating( 5.0 out of␣
↪5.0)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



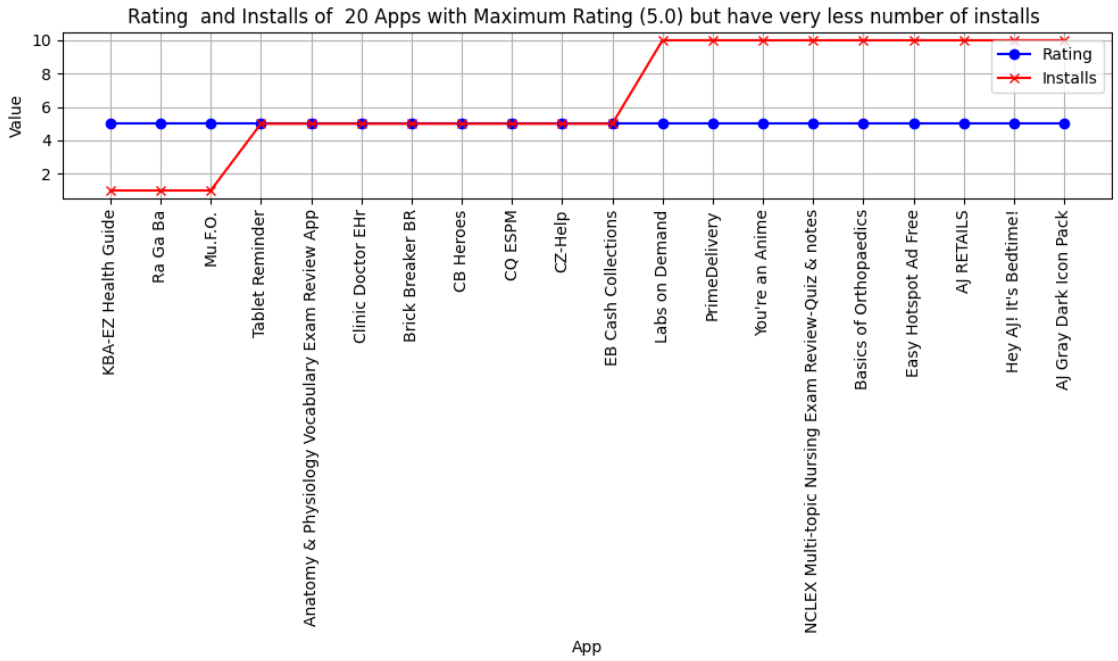
15 8) Find 20 max rated apps which have minimum number of installation

[ ]:
```
max_rating = df['Rating'].max()
max_rating_apps = df[df['Rating'] == max_rating]

top_20_max_rating_apps = max_rating_apps.nsmallest(20, 'Installs')

top_20_max_rating_apps.reset_index(drop=True, inplace=True)
top_20_max_rating_apps.index += 1

selected_columns = top_20_max_rating_apps[['App', 'Rating', 'Installs']]

apps_list = selected_columns.to_dict(orient='records')

print(tabulate(apps_list, headers='keys', tablefmt='psql'))
```

```
+-----------------------------------------------------+----------+------------+
| App                                                 |   Rating |   Installs |
|-----------------------------------------------------+----------+------------|
| KBA-EZ Health Guide                                 |        5 |          1 |
| Ra Ga Ba                                            |        5 |          1 |
| Mu.F.O.                                             |        5 |          1 |
| Tablet Reminder                                     |        5 |          5 |
| Anatomy & Physiology Vocabulary Exam Review App     |        5 |          5 |
```

```
| Clinic Doctor EHr                                    |         5 |           5 |
| Brick Breaker BR                                     |         5 |           5 |
| CB Heroes                                            |         5 |           5 |
| CQ ESPM                                              |         5 |           5 |
| CZ-Help                                              |         5 |           5 |
| EB Cash Collections                                  |         5 |           5 |
| Labs on Demand                                       |         5 |          10 |
| PrimeDelivery                                        |         5 |          10 |
| You're an Anime                                      |         5 |          10 |
| NCLEX Multi-topic Nursing Exam Review-Quiz & notes |         5 |          10 |
| Basics of Orthopaedics                               |         5 |          10 |
| Easy Hotspot Ad Free                                 |         5 |          10 |
| AJ RETAILS                                           |         5 |          10 |
| Hey AJ! It's Bedtime!                                |         5 |          10 |
| AJ Gray Dark Icon Pack                               |         5 |          10 |
+------------------------------------------------------+-----------+-------------+
```

[ ]:
```
app_names = top_20_max_rating_apps['App']
ratings = top_20_max_rating_apps['Rating']
installs = top_20_max_rating_apps['Installs']

plt.figure(figsize=(10, 6))
plt.plot(app_names, ratings, marker='o', label='Rating', color='blue')
plt.plot(app_names, installs, marker='x', label='Installs', color='Red')
plt.xticks(rotation=90)
plt.xlabel('App')
plt.ylabel('Value')
plt.title('Rating  and Installs of  20 Apps with Maximum Rating (5.0) but have␣
 ↪very less number of installs ')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



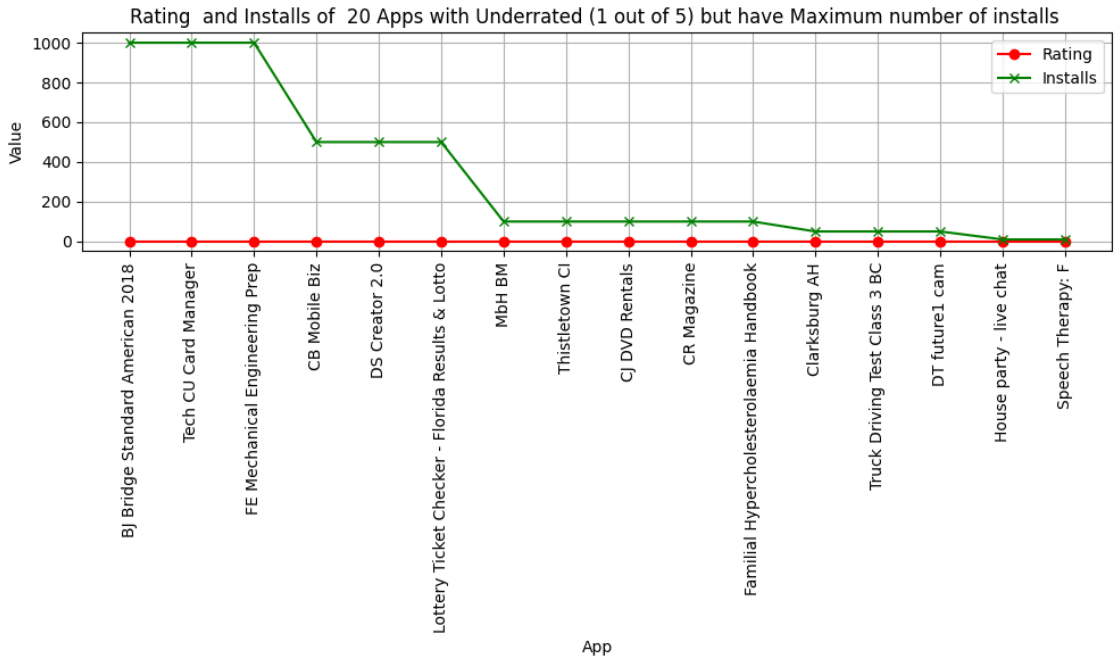
16 9) Find 20 underrated apps which have Maximum number of installation

[ ]:
```
min_rating = df['Rating'].min()
min_rating_apps = df[df['Rating'] == min_rating]

top_20_min_rating_apps = min_rating_apps.nlargest(20, 'Installs')

top_20_min_rating_apps.reset_index(drop=True, inplace=True)
top_20_min_rating_apps.index += 1

selected_columns = top_20_min_rating_apps[['App', 'Rating', 'Installs']]

apps_list = selected_columns.to_dict(orient='records')

print(tabulate(apps_list, headers='keys', tablefmt='psql'))
```

```
+-------------------------------------------------+----------+-------------+
| App                                             |   Rating |   Installs |
|-------------------------------------------------+----------+-------------|
| BJ Bridge Standard American 2018                |        1 |       1000 |
| Tech CU Card Manager                            |        1 |       1000 |
| FE Mechanical Engineering Prep                  |        1 |       1000 |
| CB Mobile Biz                                   |        1 |        500 |
| DS Creator 2.0                                  |        1 |        500 |
```

```
| Lottery Ticket Checker - Florida Results & Lotto |         1 |         500 |
| MbH BM                                           |         1 |         100 |
| Thistletown CI                                   |         1 |         100 |
| CJ DVD Rentals                                   |         1 |         100 |
| CR Magazine                                      |         1 |         100 |
| Familial Hypercholesterolaemia Handbook          |         1 |         100 |
| Clarksburg AH                                    |         1 |          50 |
| Truck Driving Test Class 3 BC                    |         1 |          50 |
| DT future1 cam                                   |         1 |          50 |
| House party - live chat                          |         1 |          10 |
| Speech Therapy: F                                |         1 |          10 |
+--------------------------------------------------+-----------+-------------+
```

[ ]:
```
app_names = top_20_min_rating_apps['App']
ratings = top_20_min_rating_apps['Rating']
installs = top_20_min_rating_apps['Installs']

plt.figure(figsize=(10, 6))
plt.plot(app_names, ratings, marker='o', label='Rating', color='red')
plt.plot(app_names, installs, marker='x', label='Installs', color='Green')
plt.xticks(rotation=90)
plt.xlabel('App')
plt.ylabel('Value')
plt.title('Rating  and Installs of  20 Apps with Underrated (1 out of 5) but␣
 ↪have Maximum number of installs ')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



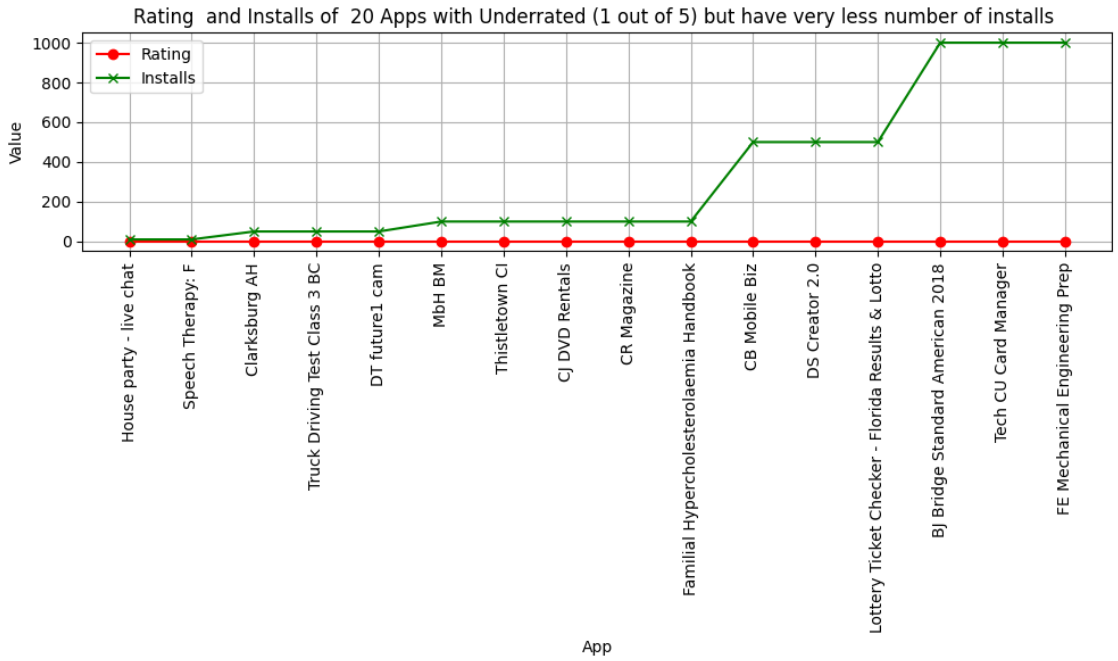
17 10) Find 20 underrated apps which have very less number of installation

[ ]:
```
min_rating = df['Rating'].min()
min_rating_apps = df[df['Rating'] == min_rating]

top_20_min_rating_apps = min_rating_apps.nsmallest(20, 'Installs')

top_20_min_rating_apps.reset_index(drop=True, inplace=True)
top_20_min_rating_apps.index += 1

selected_columns = top_20_min_rating_apps[['App', 'Rating', 'Installs']]

apps_list = selected_columns.to_dict(orient='records')

print(tabulate(apps_list, headers='keys', tablefmt='psql'))
```

```
+-----------------------------------------------+----------+------------+
| App                                           |   Rating |   Installs |
|-----------------------------------------------+----------+------------|
| House party - live chat                       |        1 |         10 |
| Speech Therapy: F                             |        1 |         10 |
| Clarksburg AH                                 |        1 |         50 |
| Truck Driving Test Class 3 BC                 |        1 |         50 |
| DT future1 cam                                |        1 |         50 |
```

```
| MbH BM                                              |         1 |         100 |
| Thistletown CI                                      |         1 |         100 |
| CJ DVD Rentals                                      |         1 |         100 |
| CR Magazine                                         |         1 |         100 |
| Familial Hypercholesterolaemia Handbook             |         1 |         100 |
| CB Mobile Biz                                       |         1 |         500 |
| DS Creator 2.0                                      |         1 |         500 |
| Lottery Ticket Checker - Florida Results & Lotto    |         1 |         500 |
| BJ Bridge Standard American 2018                    |         1 |        1000 |
| Tech CU Card Manager                                |         1 |        1000 |
| FE Mechanical Engineering Prep                      |         1 |        1000 |
+-----------------------------------------------------+-----------+-------------+
```

[ ]:
```
app_names = top_20_min_rating_apps['App']
ratings = top_20_min_rating_apps['Rating']
installs = top_20_min_rating_apps['Installs']

plt.figure(figsize=(10, 6))
plt.plot(app_names, ratings, marker='o', label='Rating', color='red')
plt.plot(app_names, installs, marker='x', label='Installs', color='Green')
plt.xticks(rotation=90)
plt.xlabel('App')
plt.ylabel('Value')
plt.title('Rating  and Installs of  20 Apps with Underrated (1 out of 5) but␣
 ↪have very less number of installs ')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



[ ]:
```
df.head(10)
```

[ ]:

	App	Category	Rating	Reviews	Size_in_Bytes	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19922944.0	10000	Free	0.0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14680064.0	500000	Free	0.0	Everyone
2	U Launcher Lite - FREE Live Cool Themes, Hide …	ART_AND_DESIGN	4.7	87510	9122611.2	5000000	Free	0.0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	26214400.0	50000000	Free	0.0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2936012.8	100000	Free	0.0	Everyone
5	Paper flowers instructions	ART_AND_DESIGN	4.4	167	5872025.6	50000	Free	0.0	Everyone
6	Smoke Effect Photo Maker - Smoke Editor	ART_AND_DESIGN	3.8	178	19922944.0	50000	Free	0.0	Everyone
7	Infinite Painter	ART_AND_DESIGN	4.1	36815	30408704.0	1000000	Free	0.0	Everyone
8	Garden Coloring Book	ART_AND_DESIGN	4.4	13791	34603008.0	1000000	Free	0.0	Everyone
9	Kids Paint Free - Drawing Fun	ART_AND_DESIGN	4.7	121	3250585.6	10000	Free	0.0	Everyone

```
                      Genres
0                Art & Design
1  Art & Design;Pretend Play
2                Art & Design
3                Art & Design
4     Art & Design;Creativity
5                Art & Design
6                Art & Design
7                Art & Design
8                Art & Design
9     Art & Design;Creativity
```

[ ]:
```
total_rows = len(df)
print("Total number of rows in the DataFrame:", total_rows)
```

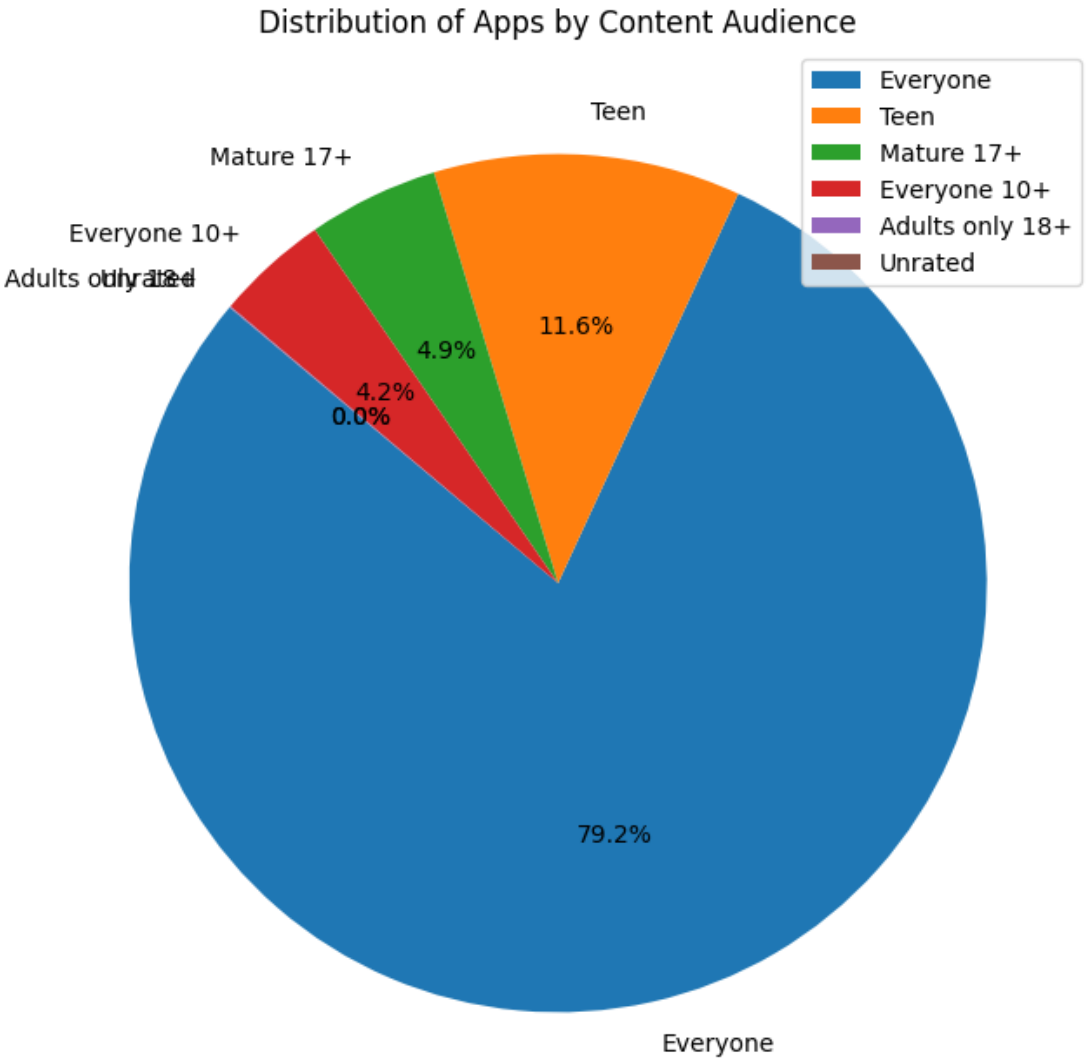
```
Total number of rows in the DataFrame: 9360
```

18 11) Differentiate based on Content

[ ]:
```
content_rating_count = df['Content Rating'].value_counts().reset_index()
content_rating_count.columns = ['Content Audience', 'Number of Apps']

content_rating_count
```

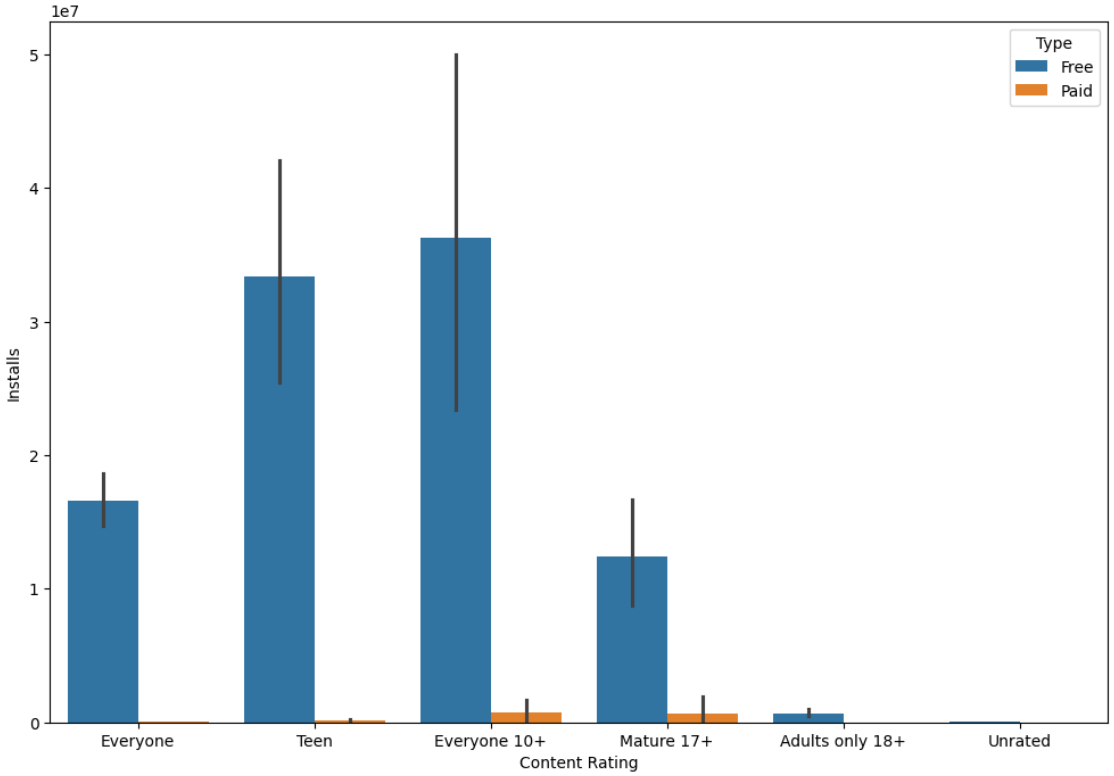
[ ]:
```
  Content Audience  Number of Apps
0         Everyone            7414
1             Teen            1084
2       Mature 17+             461
3     Everyone 10+             397
4  Adults only 18+               3
5          Unrated               1
```

[ ]:
```
plt.figure(figsize=(8, 8))
plt.pie(content_rating_count['Number of Apps'],␣
 ↪labels=content_rating_count['Content Audience'], autopct='%1.1f%%',␣
 ↪startangle=140)
plt.title('Distribution of Apps by Content Audience')
plt.legend(content_rating_count['Content Audience'], loc="best")
plt.show()
```



[ ]:
```
plt.figure(figsize=(12,8))
sns.barplot(x="Content Rating", y="Installs", hue="Type", data=df)
```

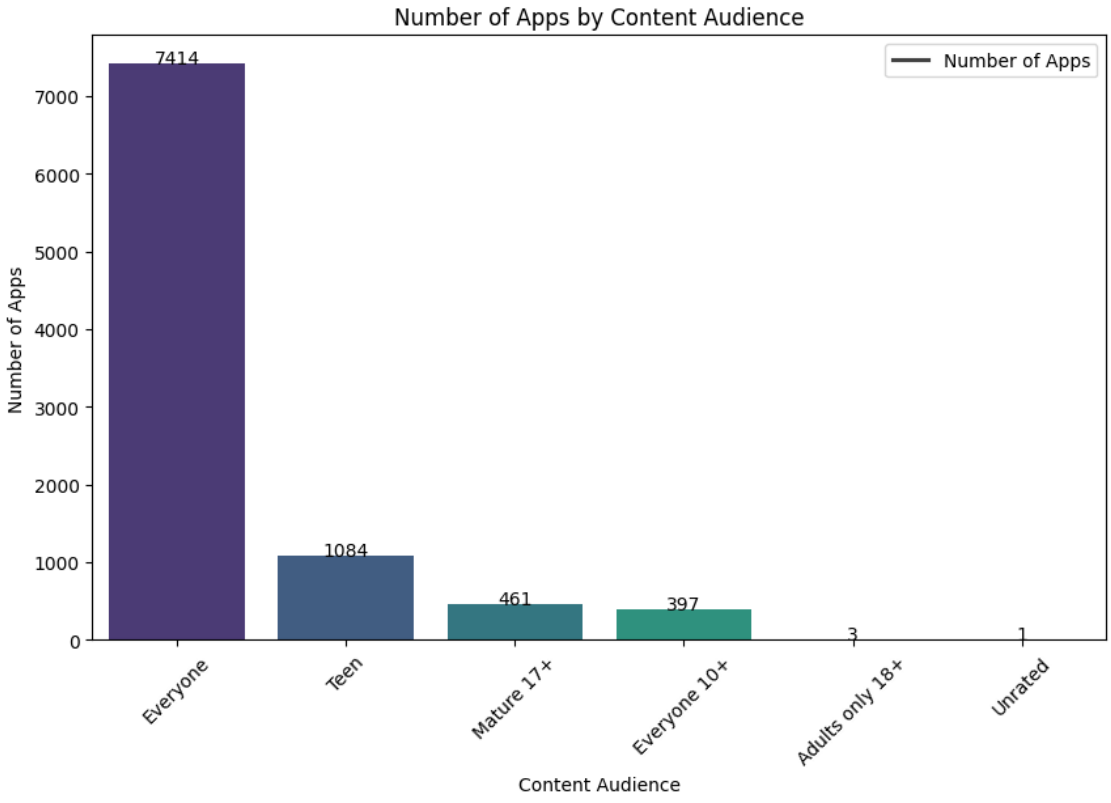
[ ]:
```
<Axes: xlabel='Content Rating', ylabel='Installs'>
```



[ ]:
```
# Create a bar plot
plt.figure(figsize=(10, 6))
bar_plot = sns.barplot(x='Content Audience', y='Number of Apps',␣
 ↪data=content_rating_count, palette='viridis')

# Add counts on top of each bar
for index, row in content_rating_count.iterrows():
    bar_plot.text(row.name, row['Number of Apps'], f'{row["Number of Apps"]}',␣
 ↪color='black', ha="center")

plt.title('Number of Apps by Content Audience')
plt.xlabel('Content Audience')
plt.ylabel('Number of Apps')
plt.xticks(rotation=45)

# Add legend
plt.legend(['Number of Apps'])

plt.show()
```

```
<ipython-input-165-28e87cbeef4c>:3: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  bar_plot = sns.barplot(x='Content Audience', y='Number of Apps',
data=content_rating_count, palette='viridis')
```

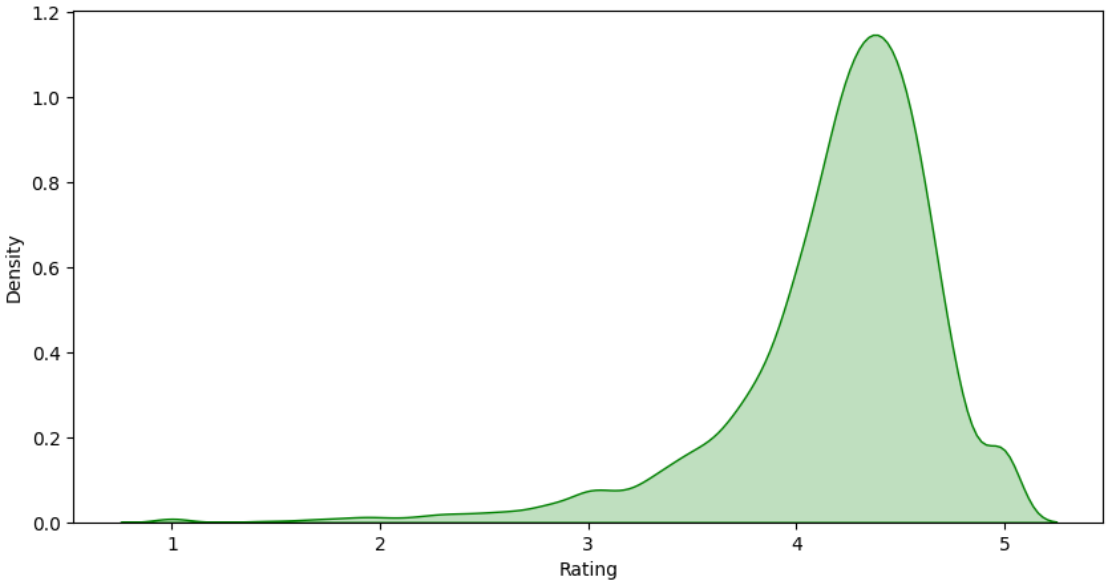


19 12) How are the ratings distributed across different apps?

[ ]:
```
# How are the ratings distributed across different apps?
plt.figure(figsize=(10, 5))
sns.kdeplot(df['Rating'], color="green", shade=True)
```

```
<ipython-input-174-3b88650a8b3a>:3: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df['Rating'], color="green", shade=True)
```

[ ]:
```
<Axes: xlabel='Rating', ylabel='Density'>
```



#13) Top 20 Biggest apps by size

[ ]:
```
# top 20 Biggest apps by size
df.nlargest(20, 'Size_in_Bytes')[['App', 'Size_in_Bytes']]
```

[ ]:

	App	Size_in_Bytes
1080	Post Bank	104857600.0
1565	Talking Babsy Baby: Baby Games	104857600.0
1758	Hungry Shark Evolution	104857600.0
1793	Mini Golf King - Multiplayer Game	104857600.0
1988	Hungry Shark Evolution	104857600.0
2299	Navi Radiography Pro	104857600.0
3973	Hungry Shark Evolution	104857600.0
4690	Vi Trainer	104857600.0
5427	Ultimate Tennis	104857600.0
5530	The Walking Dead: Our World	104857600.0
5862	Miami crime simulator	104857600.0
5865	Gangster Town: Vice District	104857600.0
7404	SimCity BuildIt	104857600.0
8409	Car Crash III Beam DH Real Damage Simulator 2018	104857600.0
8847	Draft Simulator for FUT 18	104857600.0
9170	Stickman Legends: Shadow Wars	104857600.0
1522	Chakra Cleansing	103809024.0
1697	My Talking Angela	103809024.0
1710	My Talking Angela	103809024.0
1713	Miraculous Ladybug & Cat Noir - The Official Game	103809024.0

#14) Top 20 most expensive apps

[ ]:
```
# Top 20 most expensive apps
df.nlargest(20, 'Price')[['App', 'Price']]
```

[ ]:

	App	Price
4367	I'm Rich - Trump Edition	400.00
4197	most expensive app (H)	399.99
4362	I'm rich	399.99
5351	I am rich	399.99
5354	I am Rich Plus	399.99
5356	I Am Rich Premium	399.99
5358	I am Rich!	399.99
5359	I am rich(premium)	399.99
5362	I Am Rich Pro	399.99
5364	I am rich (Most expensive app)	399.99
5369	I am Rich	399.99
5373	I AM RICH PRO PLUS	399.99
5366	I Am Rich	389.99
5357	I am extremely Rich	379.99
5355	I am rich VIP	299.99
2253	Vargo Anesthesia Mega App	79.99
2365	Vargo Anesthesia Mega App	79.99
2414	LTC AS Legal	39.99
5360	I am Rich Person	37.99
2301	A Manual of Acupuncture	33.99

20 15) Top 10 reviewed apps

[ ]:
```
# top 10 reviewed apps
df.nlargest(10, 'Reviews')[['App', 'Reviews']]
```

[ ]:

	App	Reviews
2544	Facebook	78158306
3943	Facebook	78128208
336	WhatsApp Messenger	69119316
381	WhatsApp Messenger	69119316
3904	WhatsApp Messenger	69109672
2604	Instagram	66577446
2545	Instagram	66577313
2611	Instagram	66577313
3909	Instagram	66509917
382	Messenger - Text and Video Chat for Free	56646578

21 16) Distribution of content rating and rating

[ ]:
```
data = go.Box(
    x=df['Content Rating'],
    y=df['Rating'],
    marker_color='mediumorchid',
)
layout = go.Layout(height=800,
                   width=1200,
                   title={
                       'text': "Distribution Of Content Rating",
                       'x': 0.4,
                       'y': 0.93,
                       'xanchor': 'center',
                       'yanchor': 'top'
                   },
                   xaxis={'title': 'Content Rating'},
                   yaxis=dict(title='Rating'),
                   template='simple_white')
fig = go.Figure(data=data, layout=layout)
fig.show()
```

22 observations and conclusions

#Based on the provided data, here is a summary

Top Apps:

The most popular app is ROBLOX with a rating of 9. Other notable apps include 8 Ball Pool, Bubble Shooter, Helix Jump, Zombie Catchers, Bowmasters, Candy Crush Saga, and Temple Run 2.

App Categories:

The majority of apps fall into the FAMILY category (1717), followed by GAME (1074) and TOOLS (733).

App Ratings:

The most common ratings are 4.4, 4.3, and 4.5, indicating a generally positive user sentiment.

App Sizes:

A significant number of apps have a size that varies with the device (1637), with other common sizes around 14M, 12M, 15M, and 11M.

Number of Installs:

A large portion of apps has been installed over 1,000,000 times (1576), with other popular install ranges being 10,000,000+ and 100,000+.

App Types:

The majority of apps are Free (8275), with only a small portion being Paid (611).

App Genres:

Top genres include Tools, Entertainment, Education, Action, and Productivity.

Content Ratings:

The majority of apps are rated Everyone (7089), followed by Teen, Mature 17+, and Everyone 10+.

23 Key Conclusions:

Market Diversity:

There appears to be significant diversity in the app market, with a notable emphasis on family-oriented and gaming applications.

Positive Ratings:

The distribution of user ratings suggests a general preference and liking, with a high proportion of apps receiving favorable ratings.

High Installations:

A large number of apps have been installed over a million times, indicating widespread popularity among users.

Regular Updates:

The presence of a substantial number of updates, especially around August 2018, indicates ongoing developer interest in enhancing and updating the applications.

Trend Towards Free Apps:

It appears that the majority of apps are offered for free, reflecting the prevalent trend of providing free services to users.

Broad Android Version Support:

Developers seem to target a wide range of Android operating system versions to ensure app compatibility with a diverse array of devices.

User Engagement:

Positive ratings and high installation numbers suggest strong user engagement with these applications.