Image Source

Importing Necessary Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='darkgrid')
%matplotlib inline
from datetime import datetime
import re
```

```
! pip install wordcloud
```

```
Requirement already satisfied: wordcloud in
/opt/conda/lib/python3.7/site-packages (1.8.2.2)
Requirement already satisfied: pillow in
/opt/conda/lib/python3.7/site-packages (from wordcloud) (9.1.1)
Requirement already satisfied: numpy>=1.6.1 in
/opt/conda/lib/python3.7/site-packages (from wordcloud) (1.21.6)
Requirement already satisfied: matplotlib in
/opt/conda/lib/python3.7/site-packages (from wordcloud) (3.5.3)
Requirement already satisfied: pyparsing>=2.2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->wordcloud)
(3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->wordcloud)
(1.4.3)
Requirement already satisfied: cycler>=0.10 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->wordcloud)
(0.11.0)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->wordcloud)
(21.3)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->wordcloud)
(2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.7/site-packages (from matplotlib->wordcloud)
(4.33.3)
Requirement already satisfied: typing-extensions in
/opt/conda/lib/python3.7/site-packages (from kiwisolver>=1.0.1-
>matplotlib->wordcloud) (4.4.0)
Requirement already satisfied: six>=1.5 in
/opt/conda/lib/python3.7/site-packages (from python-dateutil>=2.7-
>matplotlib->wordcloud) (1.15.0)
WARNING: Running pip as the 'root' user can result in broken
```

```
permissions and conflicting behaviour with the system package manager.
It is recommended to use a virtual environment instead:
https://pip.pypa.io/warnings/venv

from wordcloud import WordCloud, STOPWORDS
```

Loading Data into the DataFrame

```
data =
pd.read_csv("/kaggle/input/play-store-apps/googleplaystore.csv")
data.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

data.shape

(10841, 13)
```

Getting info about the data

```
data.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
```

Removing NaN values from data

```
data.isnull().any()
```

```
App                  False
Category             False
Rating                True
Reviews              False
Size                 False
Installs             False
Type                  True
Price                False
Content Rating        True
Genres               False
Last Updated         False
Current Ver           True
Android Ver           True
dtype: bool
```

```
data.isnull().any().sum()
```

```
5
```

```
data[data.isnull().any(axis=1)].head()
```

```
                                 App        Category  Rating Reviews
Size  \
15   Learn To Draw Kawaii Characters  ART_AND_DESIGN     3.2      55
2.7M
23            Mcqueen Coloring pages  ART_AND_DESIGN     NaN      61
7.0M
113        Wrinkles and rejuvenation          BEAUTY     NaN     182
5.7M
123           Manicure - nail design          BEAUTY     NaN     119
3.7M
126     Skin Care and Natural Beauty          BEAUTY     NaN     654
7.4M

     Installs  Type Price Content Rating
Genres  \
15     5,000+  Free     0       Everyone                         Art &
Design
23   100,000+  Free     0       Everyone  Art & Design;Action &
Adventure
113  100,000+  Free     0   Everyone 10+
Beauty
123   50,000+  Free     0       Everyone
Beauty
126  100,000+  Free     0           Teen
Beauty

           Last Updated Current Ver Android Ver
15         June 6, 2018         NaN  4.2 and up
23        March 7, 2018       1.0.0  4.1 and up
113  September 20, 2017         8.0  3.0 and up
123       July 23, 2018         1.3  4.1 and up
126       July 17, 2018        1.15  4.1 and up
```

```
data.dropna(axis = 0, inplace = True)
data.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  
```

```
data.isnull().any()
```

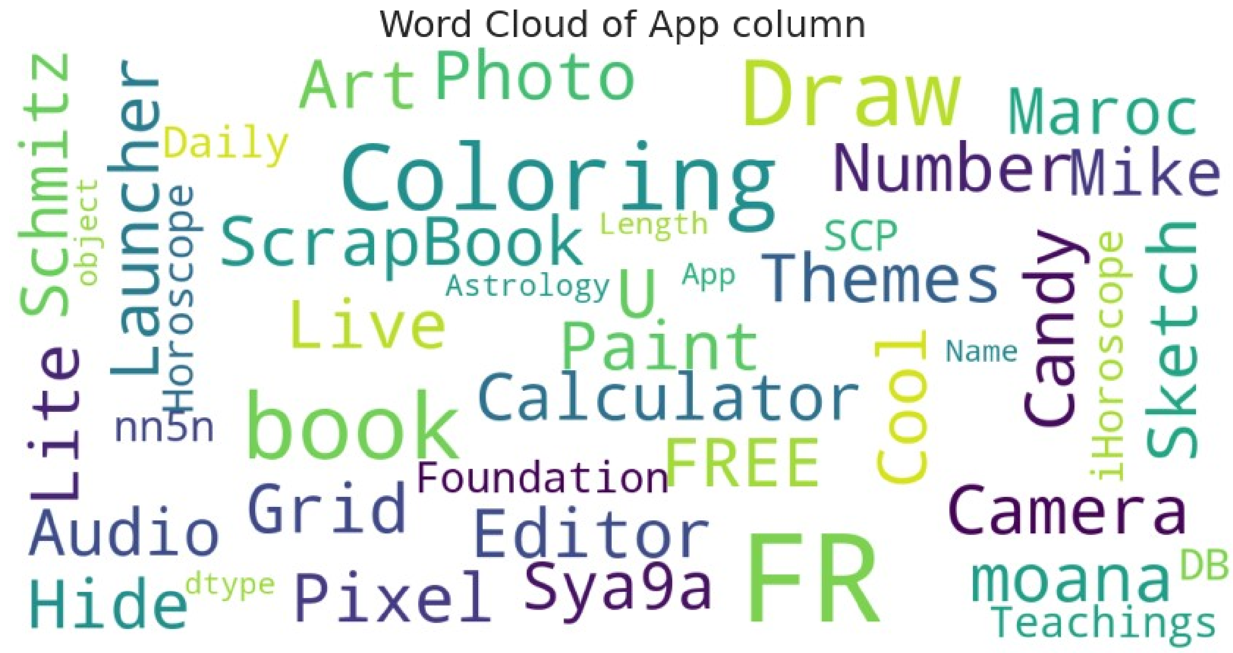
```
App               False
Category          False
Rating            False
Reviews           False
Size              False
Installs          False
Type              False
Price             False
Content Rating    False
Genres            False
Last Updated      False
Current Ver       False
Android Ver       False
dtype: bool
```

```
data.isnull().any().sum()
```

```
0
```

Creating Word-Cloud of App Column

```
stopwords = set(STOPWORDS)

wordcloud = WordCloud(background_color = 'white',
                      stopwords = stopwords,
                      max_words = 300,
                      max_font_size = 40,
                      scale = 3,
                      random_state = 1).generate(str(data["App"]))

fig = plt.figure(1, figsize = (15, 15))
plt.axis('off')
plt.title("Word Cloud of App column", fontsize = 25)
plt.imshow(wordcloud)
plt.show()
```



Converting strings to categories

```
# before conversion
data.dtypes
```

```
App                 object
Category            object
Rating             float64
Reviews             object
Size                object
Installs            object
Type                object
Price               object
Content Rating      object
Genres              object
Last Updated        object
Current Ver         object
```

```
Android Ver          object
dtype: object
```

Category Column

```
data["Category"] = data["Category"].astype('category')
data["Category"].dtype
```

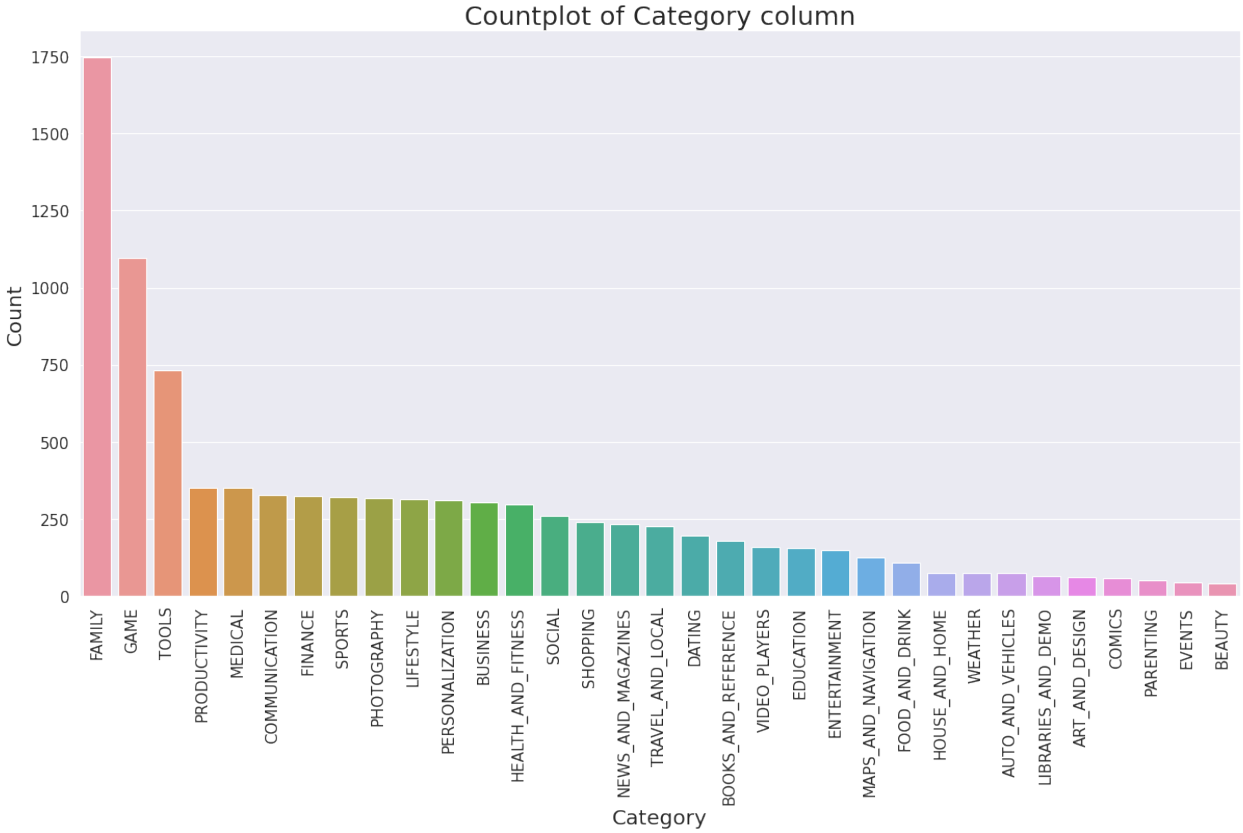
```
CategoricalDtype(categories=['ART_AND_DESIGN', 'AUTO_AND_VEHICLES',
'BEAUTY',
                  'BOOKS_AND_REFERENCE', 'BUSINESS', 'COMICS',
'COMMUNICATION',
                  'DATING', 'EDUCATION', 'ENTERTAINMENT', 'EVENTS',
'FAMILY',
                  'FINANCE', 'FOOD_AND_DRINK', 'GAME',
'HEALTH_AND_FITNESS',
                  'HOUSE_AND_HOME', 'LIBRARIES_AND_DEMO', 'LIFESTYLE',
                  'MAPS_AND_NAVIGATION', 'MEDICAL',
'NEWS_AND_MAGAZINES',
                  'PARENTING', 'PERSONALIZATION', 'PHOTOGRAPHY',
                  'PRODUCTIVITY', 'SHOPPING', 'SOCIAL', 'SPORTS',
'TOOLS',
                  'TRAVEL_AND_LOCAL', 'VIDEO_PLAYERS', 'WEATHER'],
, ordered=False)
```

```
data["Category"].value_counts()
```

```
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: Category, dtype: int64
```

```
plt.figure(figsize = [20, 10])
sns.countplot(x = "Category", data = data, order =
data["Category"].value_counts().index)
plt.xticks(rotation = 90, fontsize = 15)
plt.xlabel("Category", fontsize = 20)
plt.yticks(fontsize = 15);
plt.ylabel("Count", fontsize = 20)
plt.title("Countplot of Category column", fontsize = 25)
plt.show()
```



Type Column

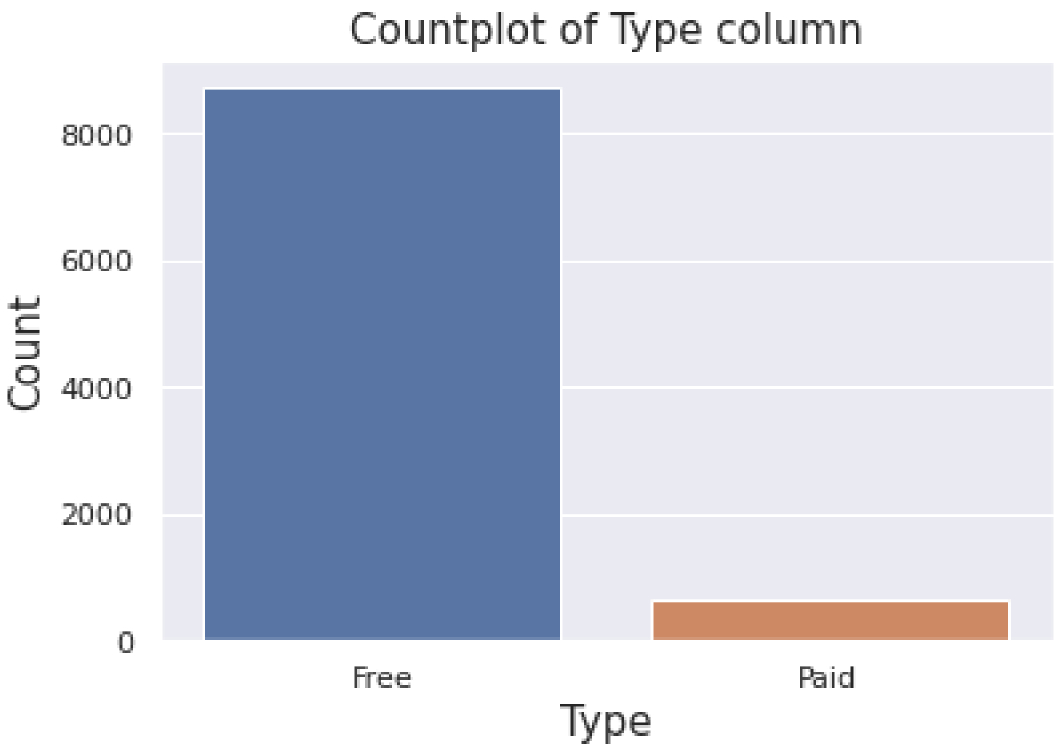
```
data["Type"] = data["Type"].astype("category")
data["Type"].dtype
```

```
CategoricalDtype(categories=['Free', 'Paid'], ordered=False)
```

```
data["Type"].value_counts()
```

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

```
sns.countplot(x = "Type", data = data)
plt.xlabel("Type", fontsize = 15)
plt.ylabel("Count", fontsize = 15)
plt.title("Countplot of Type column", fontsize = 15)
plt.show()
```

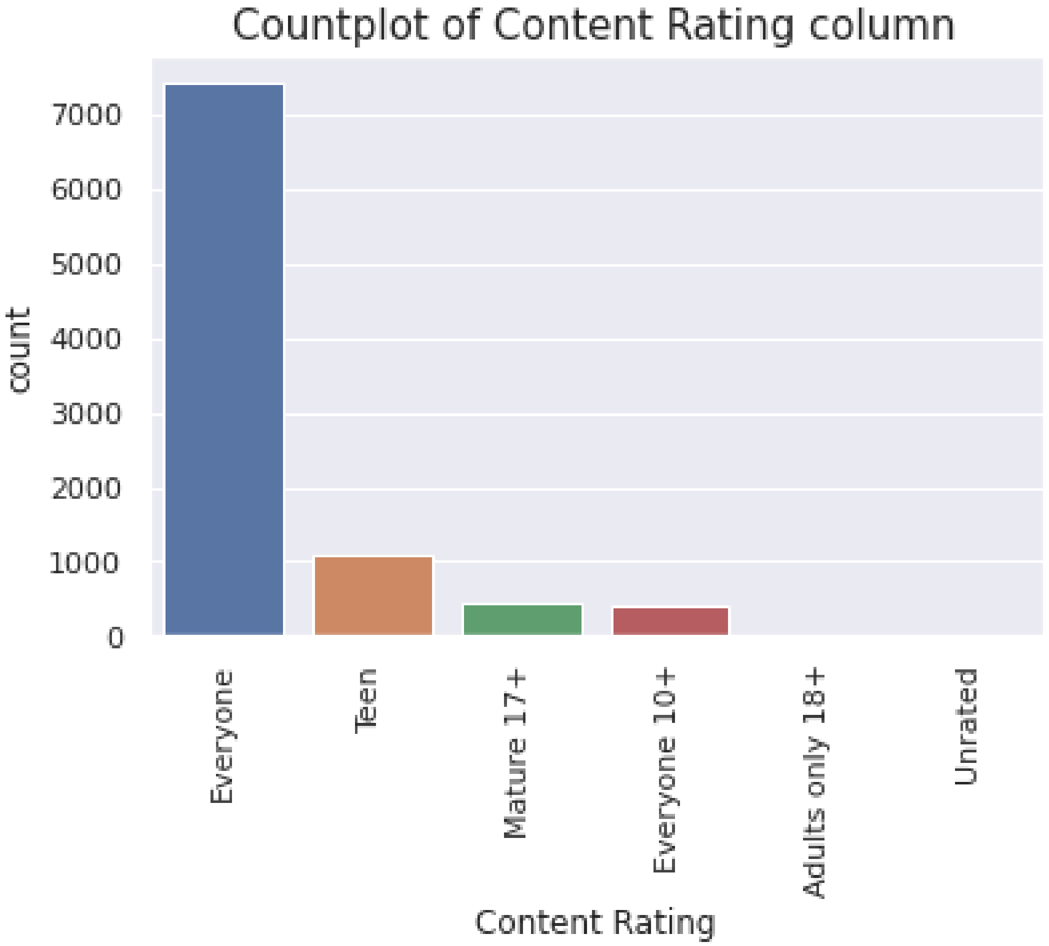


Content Rating Column

```
data["Content Rating"] = data["Content Rating"].astype("category")
data["Content Rating"].dtype
```

```
CategoricalDtype(categories=['Adults only 18+', 'Everyone', 'Everyone
10+', 'Mature 17+',
                  'Teen', 'Unrated'],
, ordered=False)
```

```
data["Content Rating"].value_counts()

Everyone           7414
Teen               1084
Mature 17+          461
Everyone 10+        397
Adults only 18+       3
Unrated               1
Name: Content Rating, dtype: int64

sns.countplot(x = "Content Rating", data = data, order = data["Content
Rating"].value_counts().index)
plt.title("Countplot of Content Rating column", fontsize = 15)
plt.xticks(rotation = 90)
plt.show()
```



Genres Column

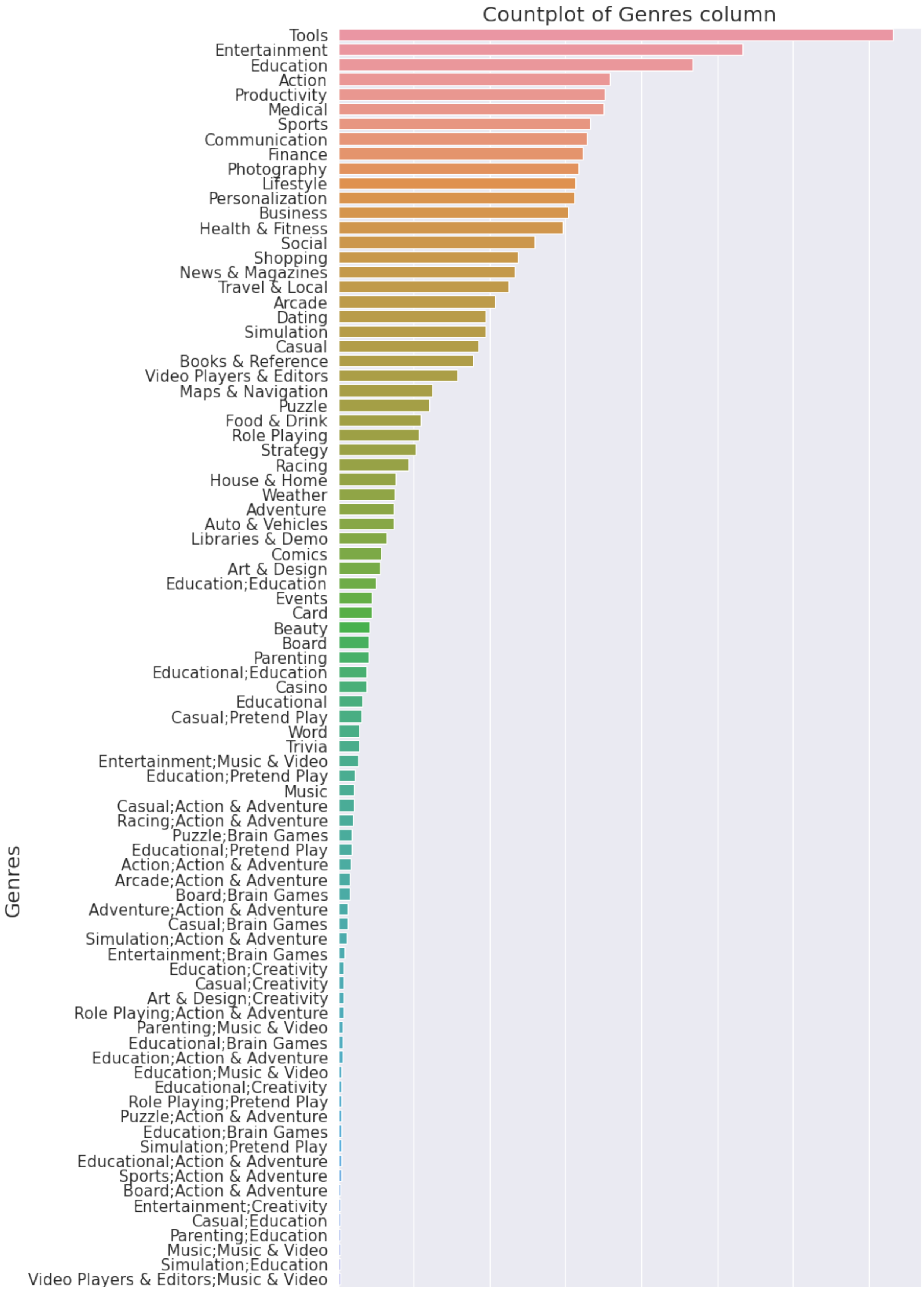
```
data["Genres"] = data["Genres"].astype("category")
data["Genres"].dtype
```

```
CategoricalDtype(categories=['Action', 'Action;Action & Adventure',
'Adventure',
                  'Adventure;Action & Adventure', 'Adventure;Brain
Games',
                  'Adventure;Education', 'Arcade', 'Arcade;Action &
Adventure',
                  'Arcade;Pretend Play', 'Art & Design',
                  ...
                  'Tools', 'Tools;Education', 'Travel & Local',
                  'Travel & Local;Action & Adventure', 'Trivia',
                  'Video Players & Editors',
                  'Video Players & Editors;Creativity',
                  'Video Players & Editors;Music & Video', 'Weather',
'Word'],
, ordered=False)
```

```
data["Genres"].value_counts()
```

```
Tools                          732
Entertainment                  533
Education                      468
Action                         358
Productivity                   351
                              ...
Parenting;Brain Games            1
Health & Fitness;Education       1
Card;Brain Games                 1
Music & Audio;Music & Video      1
Role Playing;Brain Games         1
Name: Genres, Length: 115, dtype: int64
```

```
fig = plt.figure(figsize = [10, 30])
sns.countplot(y = "Genres", data = data, order =
data["Genres"].value_counts().index)
plt.xticks(fontsize = 20)
plt.xlabel("Count", fontsize = 20)
plt.yticks(fontsize = 15);
plt.ylabel("Genres", fontsize = 20)
plt.title("Countplot of Genres column", fontsize = 20)
plt.show()
```


Countplot of Genres column
Genres
Tools
Entertainment
Education
Action
Productivity
Medical
Sports
Communication
Finance
Photography
Lifestyle
Personalization
Business
Health & Fitness
Social
Shopping
News & Magazines
Travel & Local
Arcade
Dating
Simulation
Casual
Books & Reference
Video Players & Editors
Maps & Navigation
Puzzle
Food & Drink
Role Playing
Strategy
Racing
House & Home
Weather
Adventure
Auto & Vehicles
Libraries & Demo
Comics
Art & Design
Education;Education
Events
Card
Beauty
Board
Parenting
Educational;Education
Casino
Educational
Casual;Pretend Play
Word
Trivia
Entertainment;Music & Video
Education;Pretend Play
Music
Casual;Action & Adventure
Racing;Action & Adventure
Puzzle;Brain Games
Educational;Pretend Play
Action;Action & Adventure
Arcade;Action & Adventure
Board;Brain Games
Adventure;Action & Adventure
Casual;Brain Games
Simulation;Action & Adventure
Entertainment;Brain Games
Education;Creativity
Casual;Creativity
Art & Design;Creativity
Role Playing;Action & Adventure
Parenting;Music & Video
Educational;Brain Games
Education;Action & Adventure
Education;Music & Video
Educational;Creativity
Role Playing;Pretend Play
Puzzle;Action & Adventure
Education;Brain Games
Simulation;Pretend Play
Educational;Action & Adventure
Sports;Action & Adventure
Board;Action & Adventure
Entertainment;Creativity
Casual;Education
Parenting;Education
Music;Music & Video
Simulation;Education
Video Players & Editors;Music & Video

Convertnig string to date-time

Before Conversion

```
data[["Last Updated"]].head()
```

```
       Last Updated
0   January 7, 2018
1  January 15, 2018
2    August 1, 2018
3      June 8, 2018
4     June 20, 2018
```

```
data["Last Updated"] = pd.to_datetime(data["Last Updated"], format =
"%B %d, %Y")
data["Last Updated"].dtype
```

```
dtype('<M8[ns]')
```

After Conversion

```
data[["Last Updated"]].head()
```

```
  Last Updated
0   2018-01-07
1   2018-01-15
2   2018-08-01
3   2018-06-08
4   2018-06-20
```

Separating Date Column

```
data["Year"] = data["Last Updated"].dt.year

data["Month"] = data["Last Updated"].dt.month

data["Day"] = data["Last Updated"].dt.day

data.drop("Last Updated", axis = 1, inplace = True)

data[["Year", "Month", "Day"]].head()
```

```
   Year  Month  Day
0  2018      1    7
1  2018      1   15
2  2018      8    1
3  2018      6    8
4  2018      6   20
```

Strings to numbers

Reviews Column

Before Conversion

```
data[["Reviews"]].head()
```

```
  Reviews
0     159
1     967
2   87510
3  215644
4     967
```

```
data["Reviews"] = data["Reviews"].astype("int64")
data["Reviews"].dtype
```

```
dtype('int64')
```

After Conversion

```
data[["Reviews"]].head()
```

```
   Reviews
0      159
1      967
2    87510
3   215644
4      967
```

Size Column

Before Conversion

```
data[["Size"]].head()
```

```
   Size
0   19M
1   14M
2  8.7M
3   25M
4  2.8M
```

```
data["Size"] = data["Size"].replace('Varies with device', "5.0M")

data["Size in Numbers"] = data["Size"].apply(lambda x: re.findall(r"\
d+\.\d+|\d+", x))

data["Size in Numbers"] = data["Size in Numbers"].apply(lambda x:
"".join(x))
```

```
data["Size in Numbers"] = data["Size in Numbers"].apply(lambda x:
float(x))
data["Size in Numbers"].dtype

dtype('float64')

data["Size in M or K"] = data["Size"].apply(lambda x: re.findall(r"[A-
Za-z]", x))
data[["Size in M or K"]].head()

  Size in M or K
0            [M]
1            [M]
2            [M]
3            [M]
4            [M]

data["Size in M or K"] = data["Size in M or K"].apply(lambda x:
"".join(x))
data["Size in M or K"].dtype

dtype('O')

# converting Size in M or K column to categorical dtype
data["Size in M or K"] = data["Size in M or K"].astype("category")
data["Size in M or K"].dtype

CategoricalDtype(categories=['M', 'k'], ordered=False)
```

After Conversion

```
data[["Size in Numbers", "Size in M or K"]].head()

   Size in Numbers Size in M or K
0             19.0              M
1             14.0              M
2              8.7              M
3             25.0              M
4              2.8              M

# dropping the original Size column
data.drop("Size", inplace = True, axis = 1)
```

Price Column

Before Conversion

```
data[["Price"]].head()

  Price
0     0
1     0
```

```
2      0
3      0
4      0
```

```
data["Price in $"] = data["Price"].apply(lambda x: re.findall(r"\d+\.\d+|\d+", x))
```

```
data["Price in $"] = data["Price in $"].apply(lambda x: "".join(x))
```

```
data["Price in $"] = data["Price in $"].astype("float64")
```

```
data["Price in $"].dtype
```

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

After Conversion

```
data[["Price in $"]].head()
```

```
   Price in $
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
```

```
data.drop("Price", inplace = True, axis = 1)
```

Installs Column

Before Conversion

```
data[["Installs"]].head()
```

```
      Installs
0      10,000+
1     500,000+
2   5,000,000+
3  50,000,000+
4     100,000+
```

```
data["Installs"] = data["Installs"].apply(lambda x: re.findall(r"\d+", x))
```

```
data["Installs"] = data["Installs"].apply(lambda x: "".join(x))
```

```
data["Installs"] = data["Installs"].astype("int64")
```

```
data["Installs"].dtype
```

```
dtype('int64')
```

After Conversion

```
data[["Installs"]].head()
```

```
   Installs
0     10000
1    500000
2   5000000
3  50000000
4    100000
```

Current Ver Column

Before Conversion

```
data[["Current Ver"]].head()
```

```
          Current Ver
0               1.0.0
1               2.0.0
2               1.2.4
3  Varies with device
4                 1.1
```

```
data["Current Ver"] = data["Current Ver"].replace("Varies with
device", "1.0")

data["Current Ver"] = data["Current Ver"].apply(lambda x:
re.findall(r"^(\d+\.\d+)", x))

data["Current Ver"] = data["Current Ver"].apply(lambda x: "".join(x))

data["Current Ver"] = data["Current Ver"].replace("", "1.0")

data["Current Ver"] = data["Current Ver"].astype("float64")

data["Current Ver"].dtype
```

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

After Conversion

```
data[["Current Ver"]].head()
```

```
   Current Ver
0          1.0
1          2.0
2          1.2
3          1.0
4          1.1
```

Android Ver Column

Before Conversion

```
data[["Android Ver"]].head()
```

```
    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["Android Ver"] = data["Android Ver"].replace("Varies with 
device", "1.0")

data["Android Ver"] = data["Android Ver"].apply(lambda x: 
re.findall(r"^(\d+\.\d+)", x))

data["Android Ver"] = data["Android Ver"].apply(lambda x: "".join(x))

data["Android Ver"] = data["Android Ver"].replace("", "1.0")

data["Android Ver"] = data["Android Ver"].astype("float64")

data["Android Ver"].dtype
```

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

After Conversion

```
data[["Android Ver"]].head()
```

```
   Android Ver
0          4.0
1          4.0
2          4.0
3          4.2
4          4.4
```

```
# after conversion
data.dtypes
```

```
App                   object
Category            category
Rating               float64
Reviews                int64
Installs               int64
Type                category
Content Rating      category
Genres              category
Current Ver          float64
```

```
Android Ver         float64
Year                  int64
Month                 int64
Day                   int64
Size in Numbers     float64
Size in M or K     category
Price in $          float64
dtype: object
```

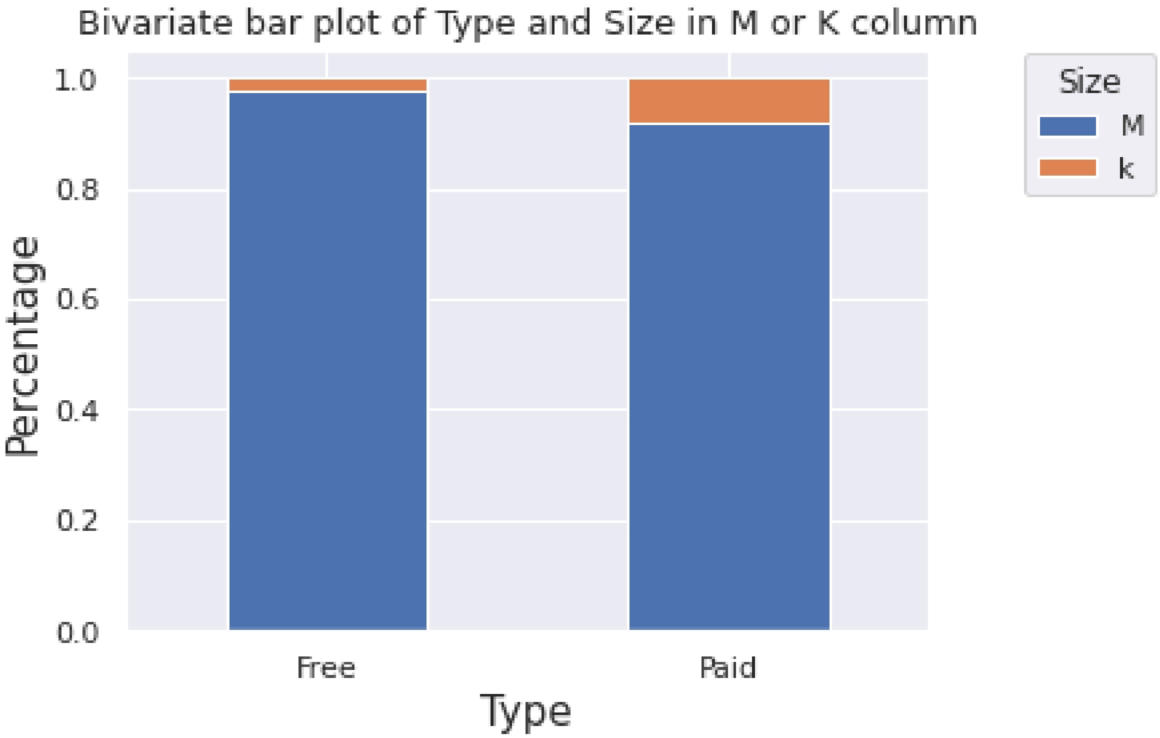
```
data.head()
```

```
                                                 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  Installs  Type Content Rating
Genres  \
0      159     10000  Free       Everyone               Art & Design

1      967    500000  Free       Everyone  Art & Design;Pretend Play

2    87510   5000000  Free       Everyone               Art & Design

3   215644  50000000  Free           Teen               Art & Design

4      967    100000  Free       Everyone    Art & Design;Creativity


   Current Ver  Android Ver  Year  Month  Day  Size in Numbers Size in
M or K  \
0          1.0          4.0  2018      1    7             19.0
M
1          2.0          4.0  2018      1   15             14.0
M
2          1.2          4.0  2018      8    1              8.7
M
3          1.0          4.2  2018      6    8             25.0
M
4          1.1          4.4  2018      6   20              2.8
M
```

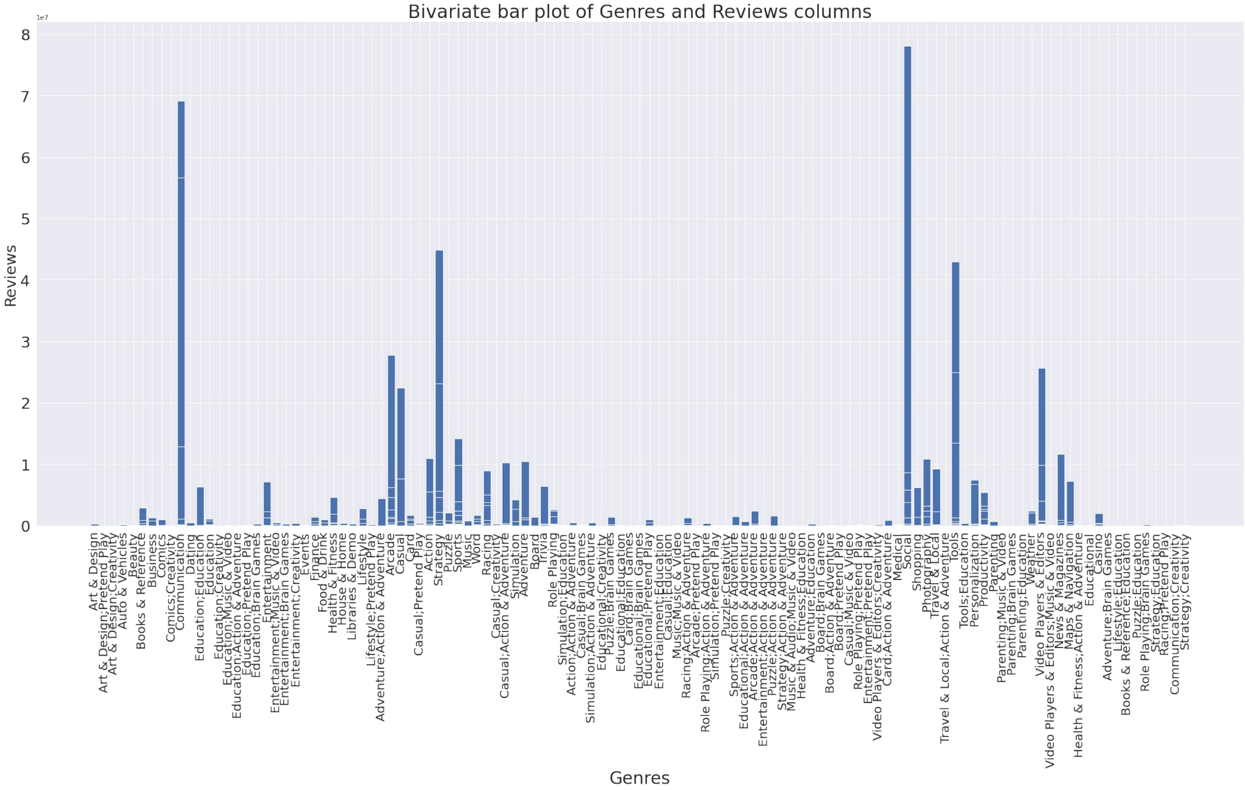
```
   Price in $
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
```

Bivariate Analysis

```
fig, ax = plt.subplots()
plt.rcParams["figure.figsize"] = [5, 5]
score_bin = pd.crosstab(data["Type"], data["Size in M or K"],
normalize = "index")
score_bin.plot.bar(stacked = True, ax = ax)
ax.set_xlabel("Type", fontsize = 15)
ax.set_ylabel("Percentage", fontsize = 15)
plt.xticks(rotation = 0)
pos = ax.get_position()
ax.set_position([pos.x0, pos.y0, pos.width * 0.9, pos.height])
ax.legend(title = "Size",loc='upper right', bbox_to_anchor=(1.3,
1.02))
plt.title("Bivariate bar plot of Type and Size in M or K column",
fontsize = 13)
plt.show()
```

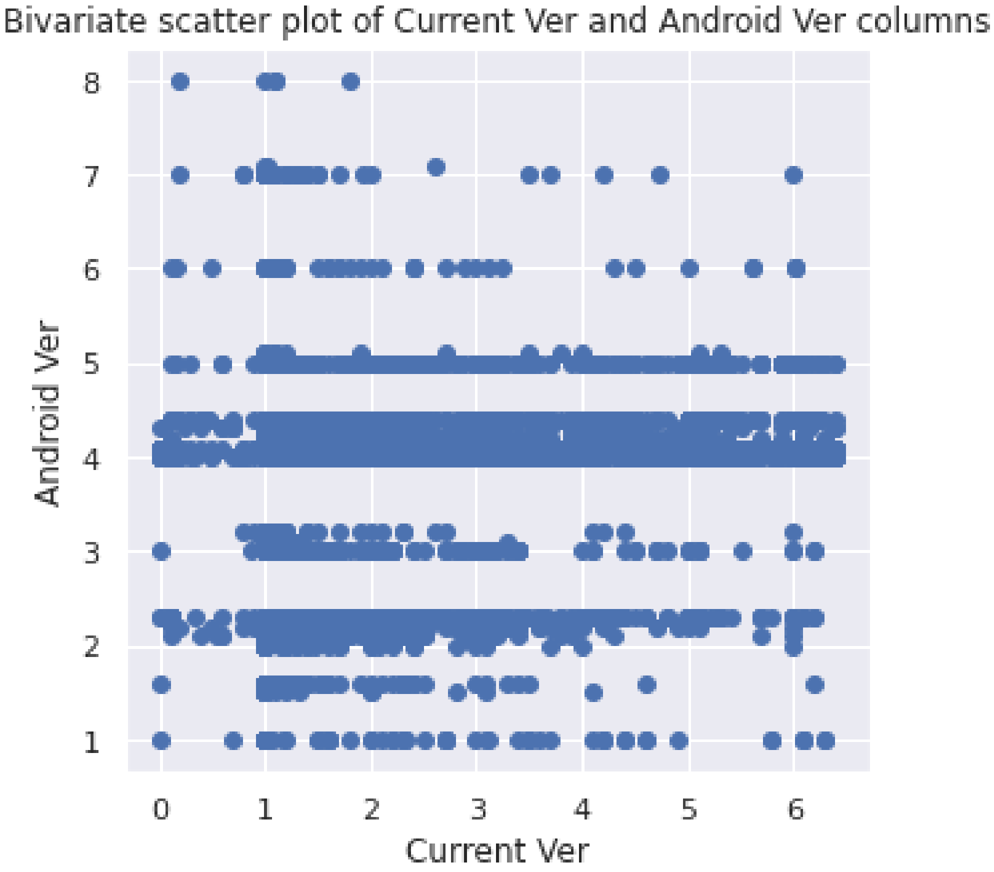


```
fig, ax = plt.subplots(figsize = [35, 15])
ax.bar(data["Genres"], data["Reviews"])
plt.xticks(rotation = 90)
plt.xlabel("Genres", fontsize = 28)
plt.ylabel("Reviews", fontsize = 25)
plt.xticks(fontsize = 20)
plt.yticks(fontsize = 25)
plt.title("Bivariate bar plot of Genres and Reviews columns", fontsize
= 30)
plt.show()
```



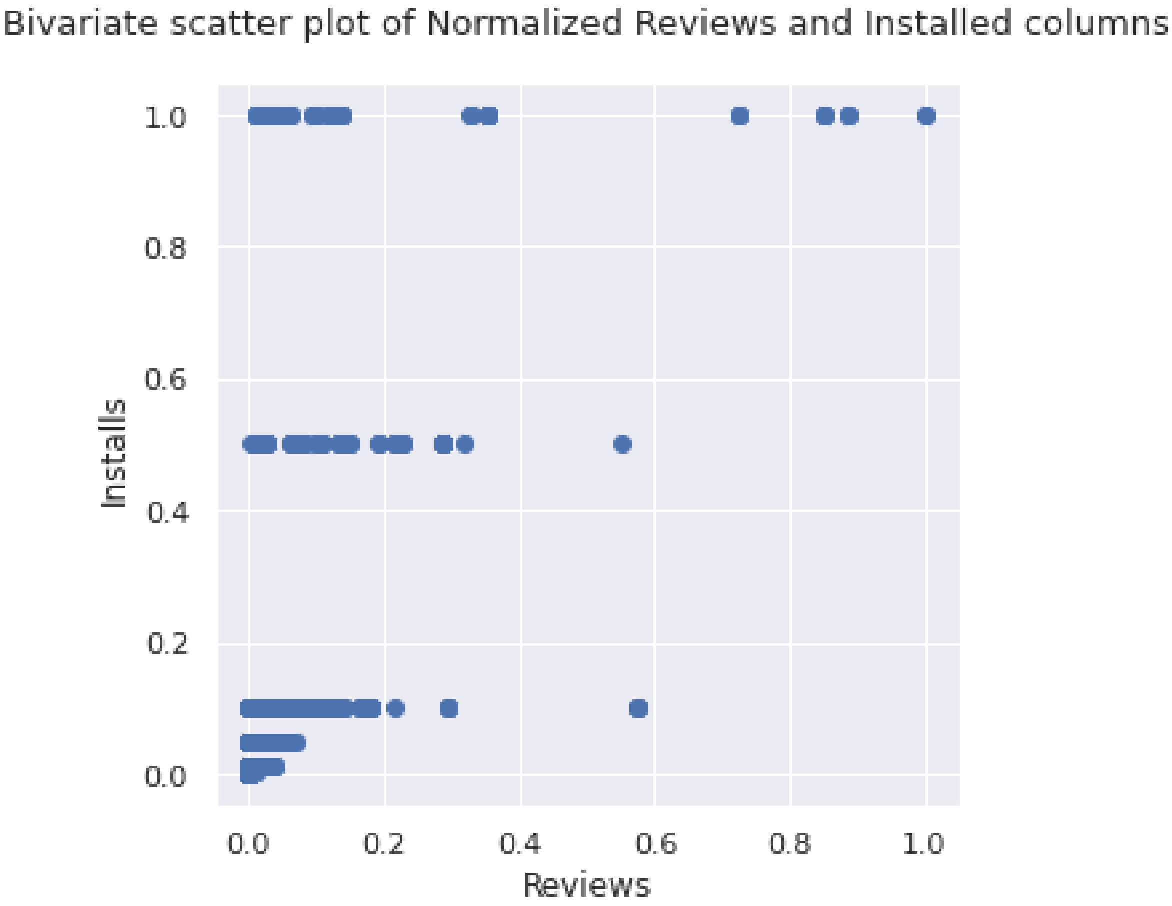
```
data=data.reset_index(drop=False)

dummy_data = data.copy()
Q1 = np.percentile(dummy_data["Current Ver"], 25, interpolation =
"midpoint")
Q3 = np.percentile(dummy_data["Current Ver"], 75, interpolation =
"midpoint")

IQR = Q3 - Q1

upper = np.where(dummy_data["Current Ver"] >= (Q3 + 1.5 * IQR))
lower = np.where(dummy_data["Current Ver"] <= (Q1 - 1.5 * IQR))
```

```
dummy_data.drop(upper[0], inplace = True)
dummy_data.drop(lower[0], inplace = True)

plt.scatter("Current Ver", "Android Ver", data = dummy_data)
plt.xlabel("Current Ver")
plt.ylabel("Android Ver")
plt.title("Bivariate scatter plot of Current Ver and Android Ver
columns")
plt.show()
```



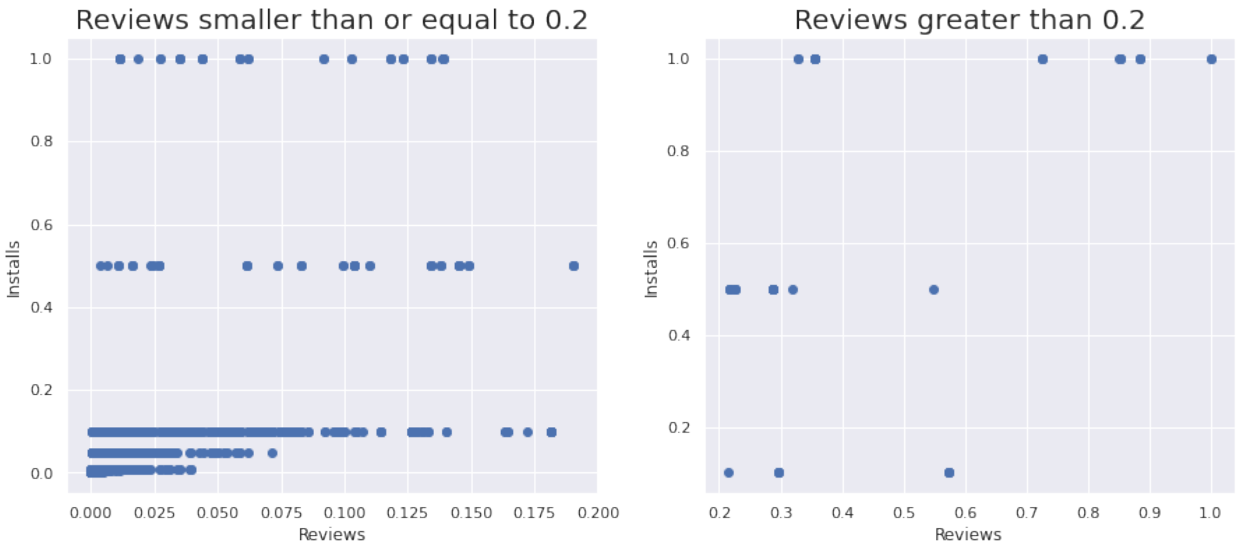
```
from sklearn.preprocessing import MinMaxScaler
dummy_data = data.copy()

column1 = "Reviews"
dummy_data[column1] =
MinMaxScaler().fit_transform(np.array(dummy_data[column1]).reshape(-
1,1))

column2 = "Installs"
dummy_data[column2] =
MinMaxScaler().fit_transform(np.array(dummy_data[column2]).reshape(-
1,1))
```

```
plt.scatter(x = "Reviews", y = "Installs", data = dummy_data)
plt.xlabel("Reviews")
plt.ylabel("Installs")
plt.title("Bivariate scatter plot of Normalized Reviews and Installed
columns", fontsize = 13, pad = 18)
plt.show()
```



```
rev_sml = dummy_data[dummy_data["Reviews"] <= 0.2]

rev_lar = dummy_data[dummy_data["Reviews"] > 0.2]

plt.figure(figsize = [15, 6])

# plot on left
plt.subplot(1, 2, 1)
plt.scatter(x = "Reviews", y = "Installs", data = rev_sml)
plt.xlabel("Reviews")
plt.ylabel("Installs")
plt.title("Reviews smaller than or equal to 0.2", fontsize = 20)

# plot on right
```

```
plt.subplot(1, 2, 2)
plt.scatter(x = "Reviews", y = "Installs", data = rev_lar)
plt.xlabel("Reviews")
plt.ylabel("Installs")
plt.title("Reviews greater than 0.2", fontsize = 20)

plt.show()
```



Encoding And Decoding

```
from sklearn.preprocessing import LabelEncoder
```

Encoding Category Column

```
data[["Category"]].head()
```

```
         Category
0  ART_AND_DESIGN
1  ART_AND_DESIGN
2  ART_AND_DESIGN
3  ART_AND_DESIGN
4  ART_AND_DESIGN
```

```
category_label_encoder = LabelEncoder()
data["Category"] =
category_label_encoder.fit_transform(data["Category"])

data[["Category"]].head()
```

```
   Category
0         0
```

```
1         0
2         0
3         0
4         0
```

Decoding Category Column

```
data["Category"] =
category_label_encoder.inverse_transform(data["Category"])

data[["Category"]].head()
```

```
         Category
0  ART_AND_DESIGN
1  ART_AND_DESIGN
2  ART_AND_DESIGN
3  ART_AND_DESIGN
4  ART_AND_DESIGN
```

Encoding Type Column

```
data[["Type"]].head()
```

```
   Type
0  Free
1  Free
2  Free
3  Free
4  Free
```

```
type_label_encoder = LabelEncoder()
data["Type"] = type_label_encoder.fit_transform(data["Type"])

data[["Type"]].head()
```

```
   Type
0     0
1     0
2     0
3     0
4     0
```

Decoding Type Column

```
data["Type"] = type_label_encoder.inverse_transform(data["Type"])

data[["Type"]].head()
```

```
   Type
0  Free
```

```
1  Free
2  Free
3  Free
4  Free
```

Encoding Content Rating Column

```
data[["Content Rating"]].head()
```

```
  Content Rating
0       Everyone
1       Everyone
2       Everyone
3           Teen
4       Everyone
```

```
content_label_encoder = LabelEncoder()
data["Content Rating"] =
content_label_encoder.fit_transform(data["Content Rating"])
```

```
data[["Content Rating"]].head()
```

```
   Content Rating
0               1
1               1
2               1
3               4
4               1
```

Decoding Content Rating Column

```
data["Content Rating"] =
content_label_encoder.inverse_transform(data["Content Rating"])
```

```
data[["Content Rating"]].head()
```

```
  Content Rating
0       Everyone
1       Everyone
2       Everyone
3           Teen
4       Everyone
```

Encoding Genres Column

```
data[["Genres"]].head()
```

```
                        Genres
0                 Art & Design
1  Art & Design;Pretend Play
```

```
2                Art & Design
3                Art & Design
4     Art & Design;Creativity
```

```
genres_label_encoder = LabelEncoder()
data["Genres"] = genres_label_encoder.fit_transform(data["Genres"])

data[["Genres"]].head()
```

```
   Genres
0       9
1      11
2       9
3       9
4      10
```

Decoding Genres Column

```
data["Genres"] =
genres_label_encoder.inverse_transform(data["Genres"])

data[["Genres"]].head()
```

```
                      Genres
0               Art & Design
1  Art & Design;Pretend Play
2               Art & Design
3               Art & Design
4    Art & Design;Creativity
```

Encoding Size in M or K Column

```
data[["Size in M or K"]].head()
```

```
  Size in M or K
0              M
1              M
2              M
3              M
4              M
```

```
size_label_encoder = LabelEncoder()
data["Size in M or K"] = size_label_encoder.fit_transform(data["Size
in M or K"])

data[["Size in M or K"]].head()
```

```
   Size in M or K
0               0
1               0
2               0
```

```
3               0
4               0
```

Decoding Size in M or K Column

```
data["Size in M or K"] =
size_label_encoder.inverse_transform(data["Size in M or K"])

data[["Size in M or K"]].head()

  Size in M or K
0              M
1              M
2              M
3              M
4              M

data.head()

   index                                                App
Category  \
0      0     Photo Editor & Candy Camera & Grid & ScrapBook
ART_AND_DESIGN
1      1                                Coloring book moana
ART_AND_DESIGN
2      2  U Launcher Lite – FREE Live Cool Themes, Hide ...
ART_AND_DESIGN
3      3                              Sketch - Draw & Paint
ART_AND_DESIGN
4      4              Pixel Draw - Number Art Coloring Book
ART_AND_DESIGN

   Rating  Reviews  Installs  Type Content Rating
Genres  \
0     4.1      159     10000  Free       Everyone                Art &
Design
1     3.9      967    500000  Free       Everyone  Art &
Design;Pretend Play
2     4.7    87510   5000000  Free       Everyone                Art &
Design
3     4.5   215644  50000000  Free           Teen                Art &
Design
4     4.3      967    100000  Free       Everyone    Art &
Design;Creativity

   Current Ver  Android Ver  Year  Month  Day  Size in Numbers Size in
M or K  \
0          1.0          4.0  2018      1    7             19.0
M
1          2.0          4.0  2018      1   15             14.0
```

```
M
2          1.2          4.0  2018      8    1              8.7
M
3          1.0          4.2  2018      6    8             25.0
M
4          1.1          4.4  2018      6   20              2.8
M

   Price in $
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
```