GOOGLE APP STORE DATA ANALYSIS

IMPORTING LIBRARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#some additional libraries
import missingno as msno
import plotly.graph_objects as go
import plotly.express as px
```

LOADING DATA

```
#READING DATA FROM APPS
df = pd.read_csv('/content/Google Apps data.csv')
```

```
#QUICK GLANCE AT THE DATA
df
```

	Unnamed: 0.1	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type
0	0	0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.0	10000	Free
1	1	1	Coloring book moana	Art And Design	3.9	967	14.0	500000	Free
2	2	5	U Launcher Lite – FREE Live Cool Themes, Hide ...	Art And Design	4.7	87510	8.7	5000000	Free
3	3	6	Sketch - Draw & Paint	Art And Design	4.5	215644	25.0	50000000	Free
4	4	7	Pixel Draw - Number Art Coloring Book	Art And Design	4.3	967	2.8	100000	Free
...	...	...	...	...	...	...	...	...	...

```
df.head()
```

	Unnamed: 0.1	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	La Updat
0	0	0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.0	10000	Free	0.0	Others	Janu 7, 20
1	1	1	Coloring book moana	Art And Design	3.9	967	14.0	500000	Free	0.0	Others	Janu 20
2	2	5	U Launcher Lite –	Art And								Aug

```
#shape of data
df.shape
```

```
(8276, 15)
```

```
#Checking Column Names in the Dataset
df.columns
```

```
Index(['Unnamed: 0.1', 'Unnamed: 0', 'App', 'Category', 'Rating', 'Reviews',
       'Size', 'Installs', 'Type', 'Price', 'Content Rating', 'Last Updated',
       'Current Ver', 'Minimum Android Ver', 'Genres'],
      dtype='object')
```

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

```
array(['Art And Design', 'Auto And Vehicles', 'Beauty',
       'Books And Reference', 'Business', 'Comics', 'Communication',
       'Dating', 'Education', 'Entertainment', 'Events', 'Finance',
       'Food And Drink', 'Health And Fitness', 'House And Home',
       'Libraries And Demo', 'Lifestyle', 'Game', 'Family', 'Medical',
       'Social', 'Shopping', 'Photography', 'Sports', 'Travel And Local',
       'Tools', 'Personalization', 'Productivity', 'Parenting', 'Weather',
       'Video Players', 'News And Magazines', 'Maps And Navigation'],
      dtype=object)
```

```
df['Category'].nunique()
```

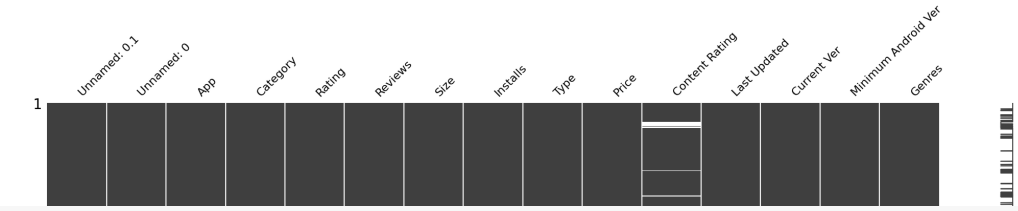
```
33
```

```
df.info()
```

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

```
msno.matrix(df)
```

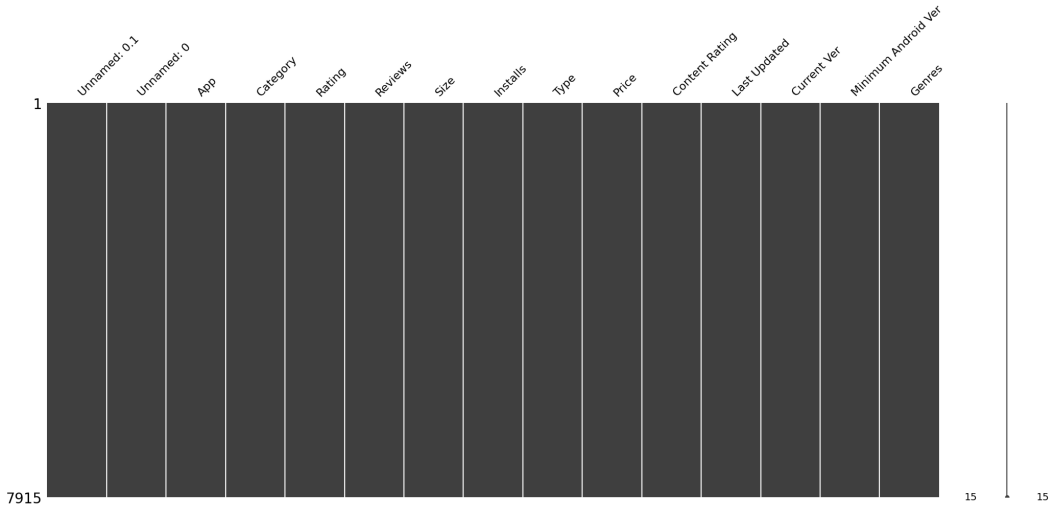
```
<Axes: >
```




```
#dropping nans, null entries in the data
df =df.dropna()
```

```
#no white lines
msno.matrix(df)
```

```
<Axes: >
```




```
#droping of columns
df.drop(['Unnamed: 0.1','Unnamed: 0'], axis = 1, inplace=True)
```

```
<ipython-input-18-44a81415c684>: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
  df.drop(['Unnamed: 0.1','Unnamed: 0'], axis = 1, inplace=True)
```

```
df
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.00000	10000	Free	0.0	Others	January 7, 2018	1.0.0
1	Coloring book moana	Art And Design	3.9	967	14.00000	500000	Free	0.0	Others	January 15, 2018	2.0.0
	U										

```
df.describe()
#Basic Statistics
```

	Rating	Reviews	Size	Installs	Price
count	7915.000000	7.915000e+03	7915.000000	7.915000e+03	7915.000000
mean	4.177486	2.821057e+05	18.714311	9.790449e+06	1.063405
std	0.535871	2.133745e+06	22.239824	6.085541e+07	17.149233
min	1.000000	1.000000e+00	0.008300	1.000000e+00	0.000000
25%	4.000000	1.250000e+02	2.700000	1.000000e+04	0.000000
50%	4.300000	3.053000e+03	9.200000	1.000000e+05	0.000000
75%	4.500000	4.546750e+04	26.000000	1.000000e+06	0.000000
max	5.000000	7.815831e+07	100.000000	1.000000e+09	400.000000

```
#Checking null values
df.isnull().sum()
```

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

```
columns = list(df)
columns
```

```
['App',
 'Category',
 'Rating',
 'Reviews',
 'Size',
 'Installs',
 'Type',
 'Price',
 'Content Rating',
 'Last Updated',
 'Current Ver',
 'Minimum Android Ver',
 'Genres']
```

```
(df[columns[1:]]==0).sum()
```

```
Category                  0
Rating                    0
Reviews                   0
Size                      0
Installs                  0
Type                      0
Price                  7326
Content Rating            0
Last Updated              0
Current Ver               0
Minimum Android Ver       0
```

```
Genres                    0
dtype: int64
```

```
#Replace statement
df[columns[1:]]=df[columns[1:]].replace(0,np.nan)
```

```
<ipython-input-24-8e73b16d4fe3>: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
  df[columns[1:]]=df[columns[1:]].replace(0,np.nan)
```

```
#before drop statement
df.shape
```

```
(7915, 13)
```

```
df.dropna(inplace =True)
```

```
<ipython-input-26-bd0d564509cf>: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
  df.dropna(inplace =True)
```

```
df.shape
```

```
(589, 13)
```

```
#Distribution Plot to Identify which technique is used
sns.distplot(df['Price'])
```

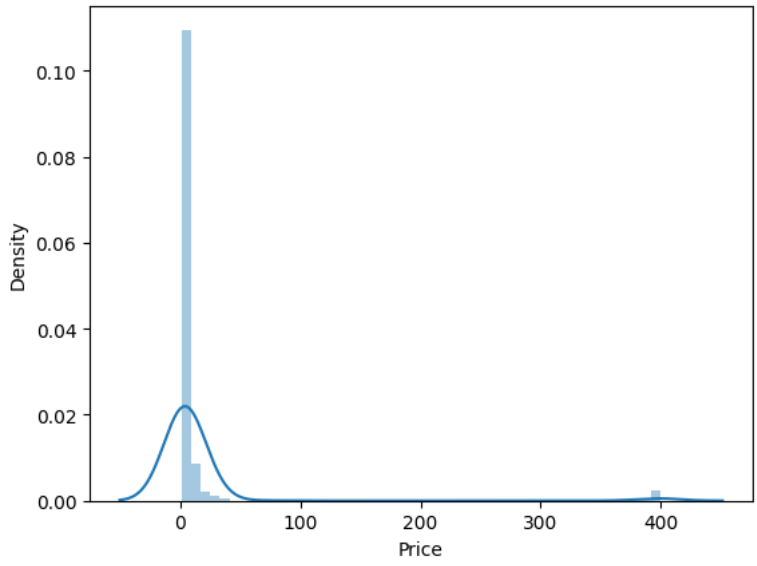
```
<ipython-input-28-2ba8b70b70bd>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Price'])
<Axes: xlabel='Price', ylabel='Density'>
```




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

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content	Last	Current	Minimum Android	Genre

```
df.duplicated()
```

```
221     False
222     False
359     False
```

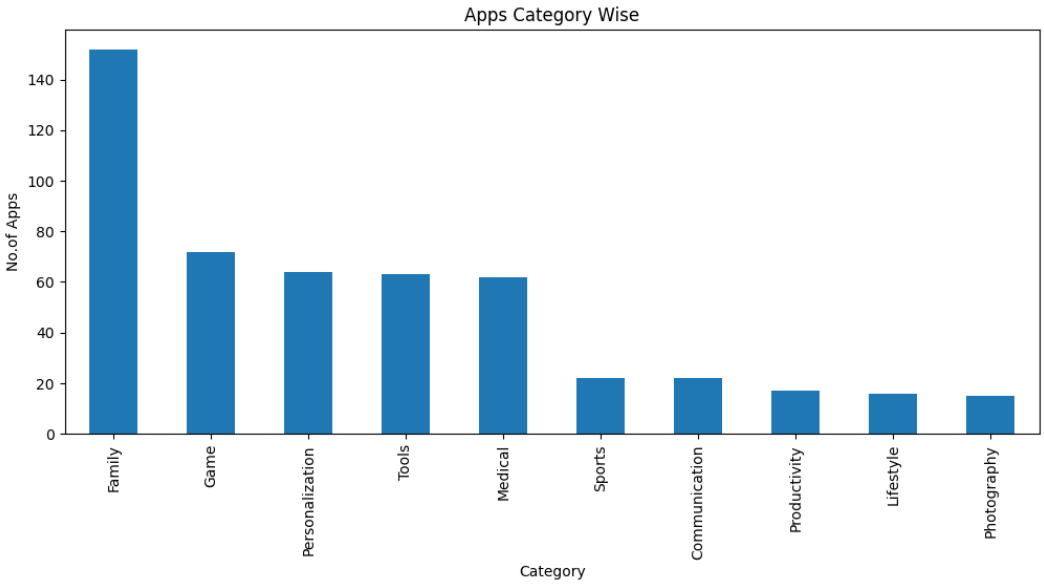
```
395     False
637     False
        ...
8179    False
8181    False
8222    False
8235    False
8238    False
Length: 589, dtype: bool
```

```
category_series = df['Category'].value_counts().head(10)
```

```
category_series
```

```
Family            152
Game               72
Personalization    64
Tools              63
Medical            62
Sports             22
Communication      22
Productivity       17
Lifestyle          16
Photography        15
Name: Category, dtype: int64
```

```
#Plot Bar Graph for the no.of Apps in each Category
plt.figure(figsize=(12,5))
plt.title("Apps Category Wise")
plt.ylabel('No.of Apps')
plt.xlabel('Category')
plt.xticks(rotation=60,fontsize=10)
df['Category'].value_counts().head(10).plot(kind='bar')
plt.show()
```




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

```
#droping of columns
df.drop(['Unnamed: 0.1','Unnamed: 0'], axis = 1, inplace=True)
```

```
df.head()
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver	Minimum Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.0	10000	Free	0.0	Others	January 7, 2018	1.0.0	4.0.3
	Coloring	Art And								January		

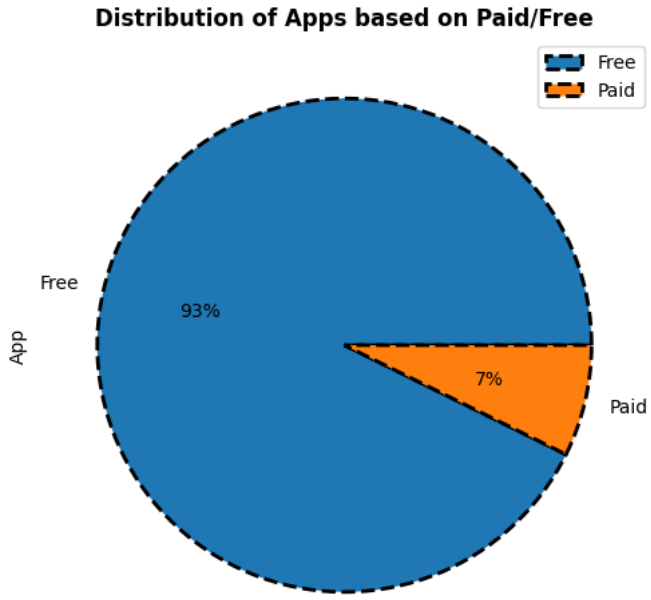
```
#Find out how many Apps are Paid and Free
free_or_paid_df=df.groupby('Type')[['App']].count()
```

```
free_or_paid_df
```

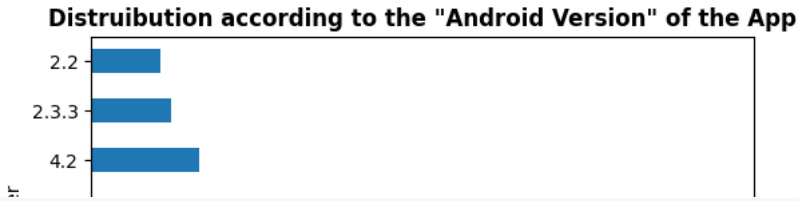
Type	App
Free	7672
Paid	604

```
#Plot pie graph for no.of apps in paid and free Type
free_or_paid_df.plot.pie(subplots=True, figsize=(12, 6), wedgeprops={"edgecolor":"0",'linewidth': 2,
                  'linestyle': 'dashed', 'antialiased': True}, autopct='%1.0f%%')
plt.title('Distribution of Apps based on Paid/Free',fontweight=600)
```

```
Text(0.5, 1.0, 'Distribution of Apps based on Paid/Free')
```

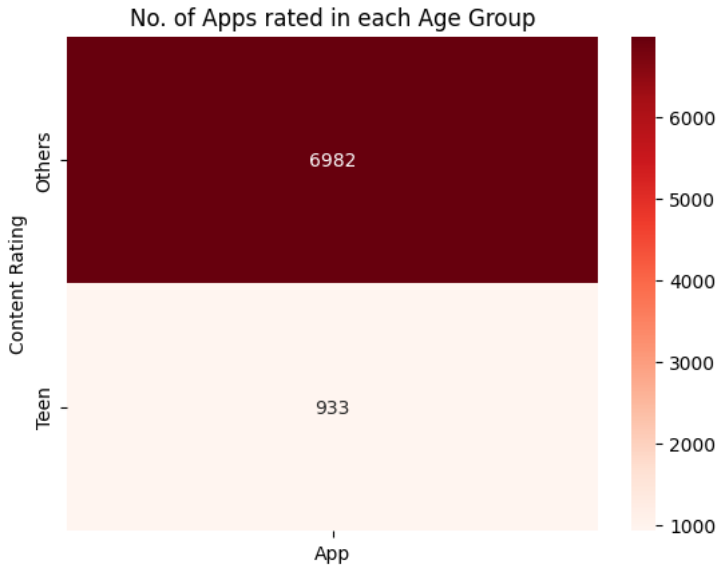



```
#* Plot Horizontal bar graph for no. of Apps per each Android Version
plt.title('Distruibution according to the "Android Version" of the App',fontweight=600)
plt.ylabel('Minimum Android Ver')
plt.xlabel('No. of Apps')
df['Minimum Android Ver'].value_counts().head(10).plot(kind='barh')
plt.show()
```

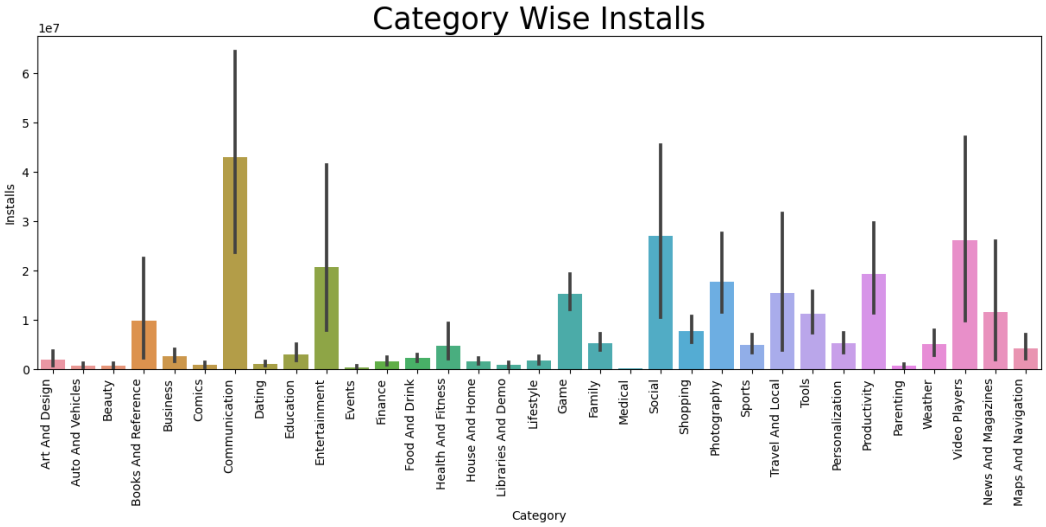



```
#* Plot Heatmap for no. of Apps in each age group
plt.title("No. of Apps rated in each Age Group")
sns.heatmap(df.groupby('Content Rating')[['App']].count(),fmt="d", annot=True, cmap='Reds')
```

```
<Axes: title={'center': 'No. of Apps rated in each Age Group'}, ylabel='Content Rating'>
```




```
#Bar Plot Graph for how many Apps installed in each Category
plt.figure(figsize=(15,5))
bar_plot_df = sns.barplot(x=df['Category'], y=df.Installs, data=df)
bar_plot_df.set_xticklabels(bar_plot_df.get_xticklabels(), rotation=90, ha="right")
plt.title('Category Wise Installs',fontsize=25)
plt.show()
```




```
## Asking and Answering Questions
df.sort_values(by=['Reviews'],ascending=False).head(10)
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Curr
1892	Facebook	Social	4.1	78158306	1.0	1000000000	Free	0.0	Teen	August 3, 2018	Va de
287	WhatsApp Messenger	Communication	4.4	69119316	1.0	1000000000	Free	0.0	Others	August 3, 2018	Va de
1893	Instagram	Social	4.5	66577313	1.0	1000000000	Free	0.0	Teen	July 31, 2018	Va de
286	Messenger – Text and Video Chat for Free	Communication	4.0	56642847	1.0	1000000000	Free	0.0	Others	August 1, 2018	Va de
1291	Clash of Clans	Game	4.6	44891723	98.0	100000000	Free	0.0	Others	July 15, 2018	10.32
3054	Clash of Clans	Family	4.6	44881447	98.0	100000000	Free	0.0	Others	July 15, 2018	10.32
3072	Clean Master-Space	Tools	4.7	42916526	1.0	500000000	Free	0.0	Others	August 3, 2018	Va

```
#to find top 10 Apps with highest Rating
df.sort_values(by=['Rating'],ascending=False).head(10)
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Curren Ve
4080	AJ Gray Dark Icon Pack	Personalization	5.0	2	35.0	10	Paid	0.99	Others	April 29, 2018	1.
5507	CD CHOICE TUBE	Family	5.0	10	5.8	500	Free	0.00	Others	July 23, 2017	0.0.
7168	EG India	Lifestyle	5.0	3	4.0	100	Free	0.00	Others	July 29, 2018	1.1.
5520	CE Smart	Tools	5.0	3	29.0	100	Free	0.00	Others	May 28, 2018	2.2.
5526	TI-84 CE Graphing Calculator Manual TI 84	Family	5.0	1	27.0	100	Paid	4.99	Others	March 28, 2018	1.5.
5532	MCQ CE	Family	5.0	22	3.6	1000	Free	0.00	Others	November	2

```
df[df.Rating >= 5.0]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver	Min And
280	Hojiboy Tojiboyev Life Hacks	Comics	5.0	15	37.0	1000	Free	0.0	Others	June 26, 2018	2.0	
495	American Girls	Dating	5.0	5	4.4	1000	Free	0.0	NaN	July 17,	3.0	

```
df.sort_values(by=['Installs'],ascending=False).head(10)
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Last Updated	Current Ver
651	Google Play Games	Entertainment	4.3	7165362	1.0	1000000000	Free	0.0	Teen	July 16, 2018	Varies with device
2856	Google News	News And Magazines	3.9	877635	13.0	1000000000	Free	0.0	Teen	August 1, 2018	5.2.0
2809	Google Play Movies & TV	Video Players	3.7	906384	1.0	1000000000	Free	0.0	Teen	August 6, 2018	Varies with device
2787	YouTube	Video Players	4.3	25655305	1.0	1000000000	Free	0.0	Teen	August 2, 2018	Varies with device
2319	Google Street View	Travel And Local	4.2	2129689	1.0	1000000000	Free	0.0	Others	August 6, 2018	Varies with device
2310	Maps - Navigate & Explore	Travel And Local	4.3	9235155	1.0	1000000000	Free	0.0	Others	July 31, 2018	Varies with device
144	Google Play Books	Books And Reference	3.9	1433233	1.0	1000000000	Free	0.0	Teen	August 3, 2018	Varies with device
										August	Varies

```
pip install squarify
```

```
Collecting squarify
  Downloading squarify-0.4.3-py3-none-any.whl (4.3 kB)
Installing collected packages: squarify
Successfully installed squarify-0.4.3
```

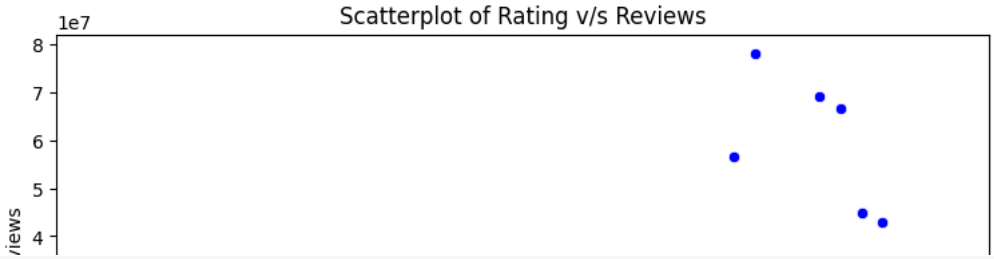
```
import matplotlib.pyplot as plt
import squarify
import pandas as pd
```

```
plt.figure(figsize = (9,4))

Rating=df['Rating']
Reviews=df['Reviews']

sns.scatterplot(x = Rating, y = Reviews, color = 'blue',)

plt.title("Scatterplot of Rating v/s Reviews")
plt.xlabel('Rating')
plt.ylabel('Reviews')
plt.show()
```

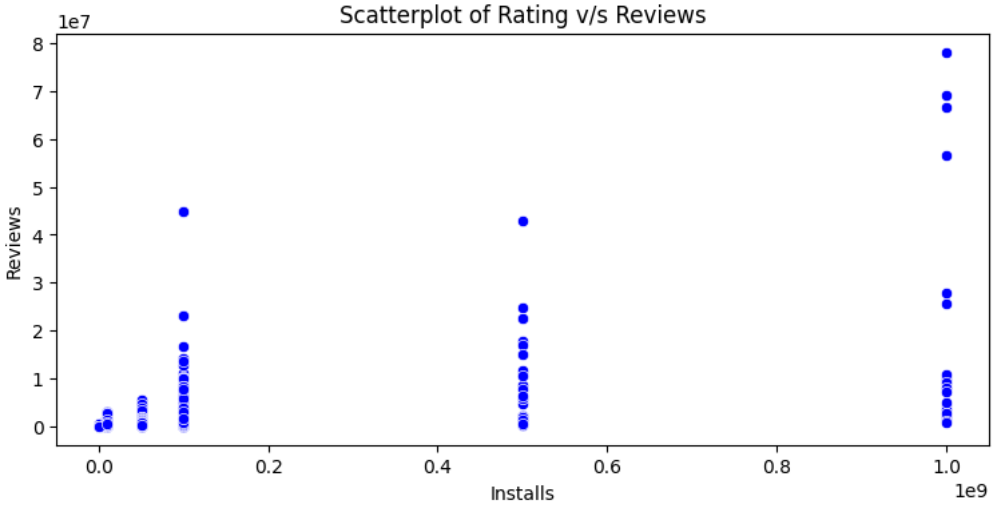



```
plt.figure(figsize = (9,4))

Rating=df['Installs']
Reviews=df['Reviews']

sns.scatterplot(x = Rating, y = Reviews, color = 'blue',)

plt.title("Scatterplot of Rating v/s Reviews")
plt.xlabel('Installs')
plt.ylabel('Reviews')
plt.show()
```

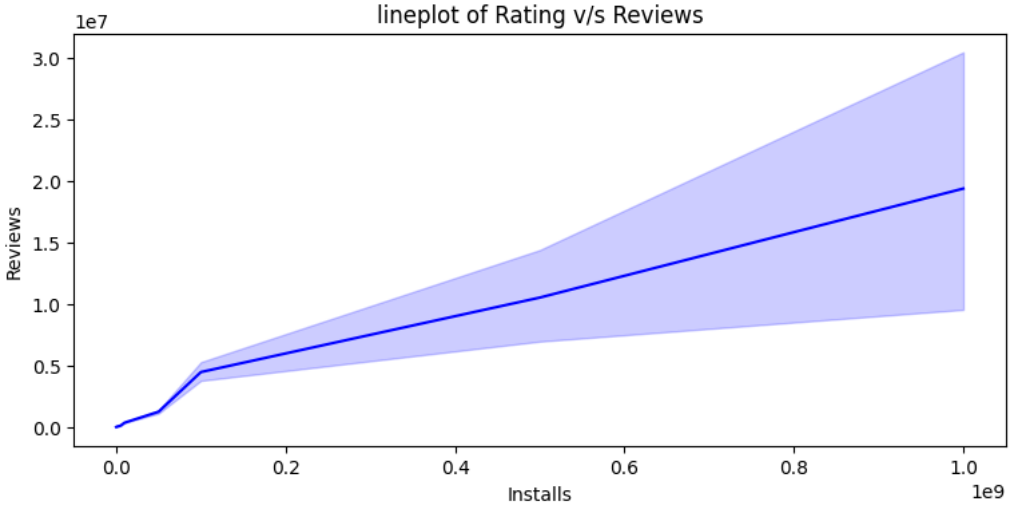



```
plt.figure(figsize = (9,4))

Rating=df['Installs']
Reviews=df['Reviews']

sns.lineplot(x = Rating, y = Reviews, color = 'blue',)

plt.title("lineplot of Rating v/s Reviews")
plt.xlabel('Installs')
plt.ylabel('Reviews')
plt.show()
```



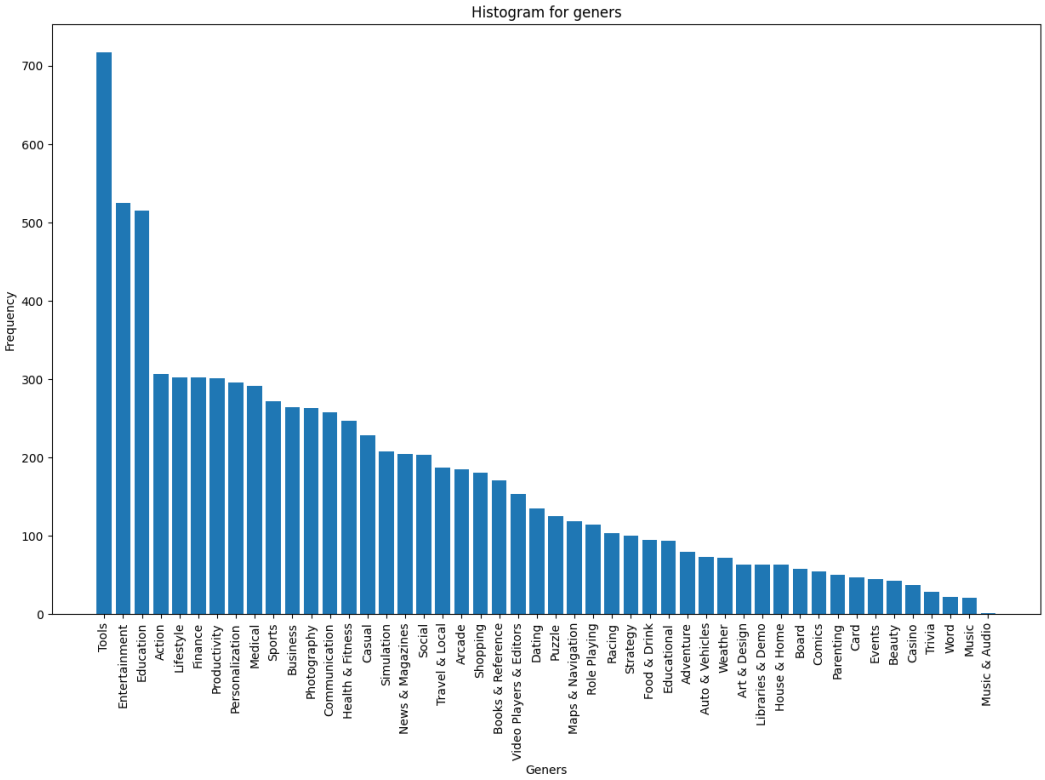

```
df['Genres'].unique()
```

```
array(['Art & Design', 'Auto & Vehicles', 'Beauty', 'Books & Reference',
       'Business', 'Comics', 'Communication', 'Dating', 'Education',
       'Entertainment', 'Events', 'Finance', 'Food & Drink',
       'Health & Fitness', 'House & Home', 'Libraries & Demo',
       'Lifestyle', 'Adventure', 'Arcade', 'Casual', 'Card', 'Action',
       'Strategy', 'Puzzle', 'Sports', 'Music', 'Word', 'Racing',
       'Simulation', 'Board', 'Trivia', 'Role Playing', 'Educational',
       'Music & Audio', 'Video Players & Editors', 'Medical', 'Social',
       'Shopping', 'Photography', 'Travel & Local', 'Tools',
       'Personalization', 'Productivity', 'Parenting', 'Weather',
       'News & Magazines', 'Maps & Navigation', 'Casino'], dtype=object)
```

```
data=df['Genres']
value=data.value_counts()
print(value)
```

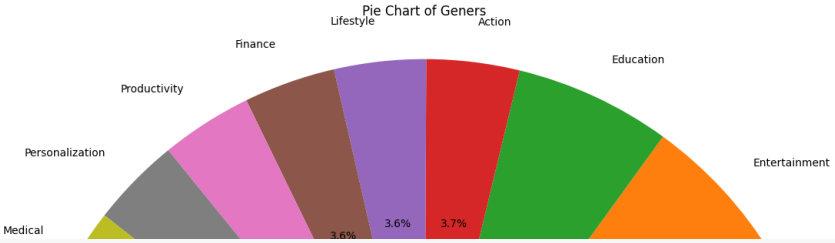
```
Tools                     717
Entertainment             525
Education                 515
Action                    306
Lifestyle                 302
Finance                   302
Productivity              301
Personalization           296
Medical                   291
Sports                    272
Business                  264
Photography               263
Communication             257
Health & Fitness          247
Casual                    228
Simulation                207
News & Magazines          204
Social                    203
Travel & Local            187
Arcade                    185
Shopping                  180
Books & Reference         171
Video Players & Editors   153
Dating                    135
Puzzle                    125
Maps & Navigation         118
Role Playing              114
Racing                    103
Strategy                  100
Food & Drink               94
Educational                93
Adventure                  79
Auto & Vehicles            73
Weather                    72
Art & Design               63
Libraries & Demo           63
House & Home               63
Board                      58
Comics                     54
Parenting                  50
Card                       47
Events                     45
Beauty                     42
Casino                     37
Trivia                     28
Word                       22
Music                      21
Music & Audio               1
Name: Genres, dtype: int64
```

```
plt.figure(figsize = (15,9))
plt.bar(value.index, value.values)
plt.xlabel('Geners')
plt.ylabel('Frequency')
plt.title('Histogram for geners')
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize=(15,15))  # Optional: Set the figure size
plt.pie(value.values, labels=value.index,autopct='%1.1f%%')
plt.title('Pie Chart of Geners')
plt.xticks(rotation=90)
plt.axis('equal')
```

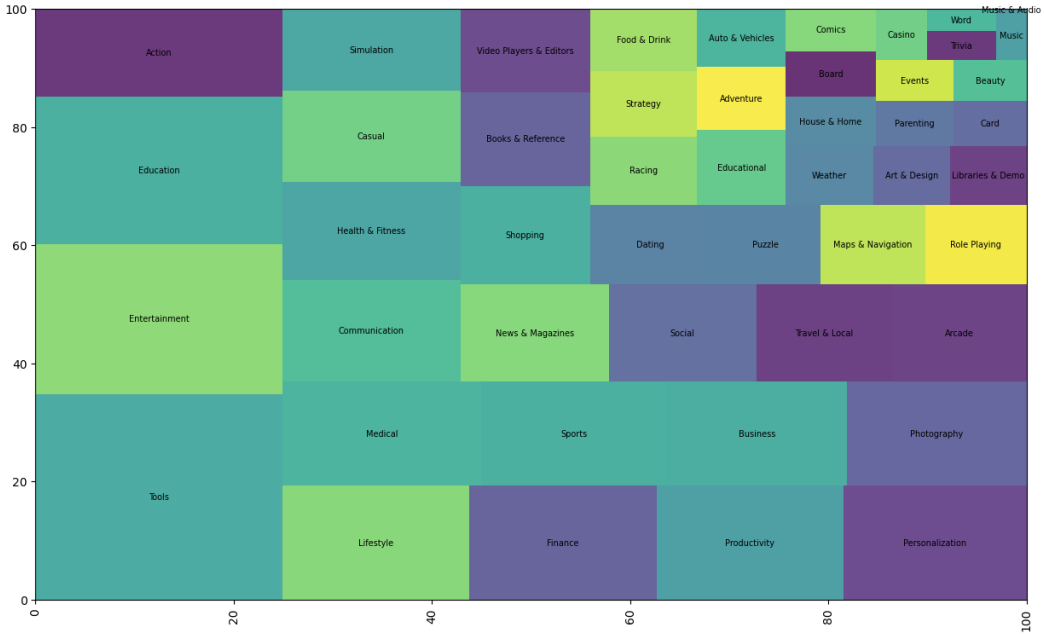
```
(-1.0999999999015733,
 1.099999999995313,
 -1.0999999999869048,
 1.0999999999983445)
```




```
df.corr()
```

```
plt.figure(figsize=(15, 9))
text_kwargs = {'fontsize': 7, 'fontweight': 'ultralight', 'color': 'black'}
squarify.plot(label=value.index, sizes=value.values,alpha=0.8, text_kwargs=text_kwargs)
```

```
(array([  0.,  20.,  40.,  60.,  80., 100.]),
 [Text(0.0, 0, '0'),
  Text(20.0, 0, '20'),
  Text(40.0, 0, '40'),
  Text(60.0, 0, '60'),
  Text(80.0, 0, '80'),
  Text(100.0, 0, '100')])
```




```
plt.figure(figsize=(21, 20))
sns.displot(df['Genres'], kde=True, bins=50)
plt.xticks(rotation=90)
```

```
(array([ 0,
         1,
         2,
         3,
         4,
         5,
         6,
         7,
         8,
         9,
         10,
         11,
         14,
         15,
         16,
         17,
         18,
         19,
         20,
         21,
         22,
         23,
         24,
         25,
         26,
         27,
         28,
         29,
         30,
         31,
         32,
         33,
         34,
         35,
         36,
         37,
```