

GOOGLE PLAY STORE Data Analysis

Data set Link : <https://www.kaggle.com/datasets/lava18/google-play-store-apps>

GitHub Link : <https://github.com/Official-Vivek-Singh?tab=repositories>

```
#Importing libraries
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# importing Dataset from local machine
df= pd.read_csv('D:\Vivek_Stuff\Learning Stuff\Python_Work\Python
Project\googleplaystore.csv')
print('Dataset loaded !!')

Dataset loaded !!
```

Show top 5 rows of your dataset

```
# .head()
df.head()
```

	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

	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

Show last 5 rows of your dataset

```
# Show Dataset last 5 Rows
df.tail()
```

Show size of your dataset

```
df.shape

# .shape is used to find the size of dataset

(10841, 13)
```

Show Headers/Columns of your dataset

```
df.columns

# .columns returns the headers of Dataset

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

Show All information of your dataset

```
df.info()

# .info() returnt the complete information of dataset , it returns the
column wise Total value count, datatype etc..
```

Show Null/Missing Values of your dataset

```
df.isnull()
df.isna()
```

isna() and isnull() both returns the True/False

	App	Category	Rating	Reviews	Size	Installs	Type	Price
\								
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...
10836	False	False	False	False	False	False	False	False
10837	False	False	False	False	False	False	False	False
10838	False	False	True	False	False	False	False	False
10839	False	False	False	False	False	False	False	False
10840	False	False	False	False	False	False	False	False
	Content	Rating	Genres	Last Updated	Current Ver	Android Ver		
0		False	False	False	False	False		
1		False	False	False	False	False		
2		False	False	False	False	False		
3		False	False	False	False	False		
4		False	False	False	False	False		
...			
10836		False	False	False	False	False		
10837		False	False	False	False	False		
10838		False	False	False	False	False		

10839	False	False	False	False	False
10840	False	False	False	False	False

[10841 rows x 13 columns]

df.isna()

.isna() returnt he Null status in True/False

	App	Category	Rating	Reviews	Size	Installs	Type	Price
\								
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...
10836	False	False	False	False	False	False	False	False
10837	False	False	False	False	False	False	False	False
10838	False	False	True	False	False	False	False	False
10839	False	False	False	False	False	False	False	False
10840	False	False	False	False	False	False	False	False

	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...
10836	False	False	False	False	False

10837	False	False	False	False	False
10838	False	False	False	False	False
10839	False	False	False	False	False
10840	False	False	False	False	False

[10841 rows x 13 columns]

Show Count of Missing/Null Values of your dataset

```
df.isnull().sum()

# .sum() to add the value ...

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
```

Percentage of Missing Value in Each Column

```
## Percentage of Missing Value in Each Column
(df.isna().sum() / df.shape[0]) * 100

App                0.000000
Category           0.000000
Rating            13.596532
Reviews            0.000000
Size               0.000000
Installs           0.000000
Type               0.009224
Price              0.000000
Content Rating     0.009224
Genres             0.000000
Last Updated       0.000000
Current Ver        0.073794
Android Ver        0.027673
```

```

Installs_1      0.000000
dtype: float64

# Show columns Name
df.columns

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

```

Show Over ALL Statistics of your dataset

```

df.describe() # for Numerical data only

# .describe() returns the statistic information
# By default it works on Numerical data

```

```

          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

```

```

# To show the Statistics of all
# for all data types we can use as below code

```

```

df.describe(include='all')

```

	App	Category	Rating	Reviews	Size
Installs \					
count	10841	10841	9367.000000	10841	10841
unique	9660	34	NaN	6002	462
top	ROBLOX	FAMILY	NaN	0	Varies with device
freq	9	1972	NaN	596	1695
mean	NaN	NaN	4.193338	NaN	NaN
std	NaN	NaN	0.537431	NaN	NaN

NaN						
min	NaN	NaN	1.000000	NaN		NaN
NaN						
25%	NaN	NaN	4.000000	NaN		NaN
NaN						
50%	NaN	NaN	4.300000	NaN		NaN
NaN						
75%	NaN	NaN	4.500000	NaN		NaN
NaN						
max	NaN	NaN	19.000000	NaN		NaN
NaN						

	Type	Price	Content	Rating	Genres	Last Updated	\
count	10840	10841		10840	10841		10841
unique	3	93		6	120		1378
top	Free	0		Everyone	Tools	August 3, 2018	
freq	10039	10040		8714	842		326
mean	NaN	NaN		NaN	NaN		NaN
std	NaN	NaN		NaN	NaN		NaN
min	NaN	NaN		NaN	NaN		NaN
25%	NaN	NaN		NaN	NaN		NaN
50%	NaN	NaN		NaN	NaN		NaN
75%	NaN	NaN		NaN	NaN		NaN
max	NaN	NaN		NaN	NaN		NaN

	Current Ver	Android Ver
count	10833	10838
unique	2832	33
top	Varies with device	4.1 and up
freq	1459	2451
mean	NaN	NaN
std	NaN	NaN
min	NaN	NaN
25%	NaN	NaN
50%	NaN	NaN
75%	NaN	NaN
max	NaN	NaN

Show Total Number of Apps Title containig Astrology

```
# Searching the Astrology in APP
```

```
df['App'].str.contains('Astrology', case= False)
```

```
0      False
1      False
```

```

2         False
3         False
4         False
...
10836    False
10837    False
10838    False
10839    False
10840     True
Name: App, Length: 10841, dtype: bool

df[df['App'].str.contains('Astrology', case= False)]
# .str.contains('') use dto search any string type text in the dataset

```

Rating \	App	Category
1570 4.6	Horoscopes – Daily Zodiac Horoscope and Astrology	LIFESTYLE
1592 4.7	☐☐☐☐ Astrology - Min Thein Kha BayDin	LIFESTYLE
10840 4.5	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE

	Reviews	Size	Installs	Type	Price	Content	Rating	Genres
1570	161143	11M	10,000,000+	Free	0	Everyone	10+	Lifestyle
1592	2225	15M	100,000+	Free	0	Everyone		Lifestyle
10840	398307	19M	10,000,000+	Free	0	Everyone		Lifestyle

	Last Updated	Current Ver	Android Ver
1570	June 25, 2018	5.2.4(881)	4.0.3 and up
1592	July 26, 2018	4.2.1	4.0.3 and up
10840	July 25, 2018	Varies with device	Varies with device

```

len(df[df['App'].str.contains('Astrology', case= False)])
3

```

so we've 3 Astrology app in our dataset`m

Finf the Avg App rating

```
# showing columns
```

```
df.columns
```



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

Showing Data of rating Column Only

```
df['Rating']
```

```
0      4.1  
1      3.9  
2      4.7  
3      4.5  
4      4.3
```

```
...
```

```
10836    4.5  
10837    5.0  
10838    NaN  
10839    4.5  
10840    4.5
```

```
Name: Rating, Length: 10841, dtype: float64
```

extracting Average of Rating Column

```
df['Rating'].mean()
```

```
4.193338315362443
```

Find the Unique App category

Showing Columns

```
df.columns
```

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

Showing the Category Column data here

```
df['Category']
```

```
0      ART_AND_DESIGN  
1      ART_AND_DESIGN
```

```
2          ART_AND_DESIGN
3          ART_AND_DESIGN
4          ART_AND_DESIGN
```

```
...
10836          FAMILY
10837          FAMILY
10838          MEDICAL
10839  BOOKS_AND_REFERENCE
10840          LIFESTYLE
```

```
Name: Category, Length: 10841, dtype: object
```

```
# extracting Unique category From The category Column
```

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

```
# .unique() returns the unique value form the given data
```

```
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',
      '1.9'], dtype=object)
```

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

```
['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',  
'1.9']
```

extracting Unique category From The category Column

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

.nunique() returns the total number of unique value form the given data

34

Show Category Wise Total App

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

Category	
FAMILY	1972
GAME	1144
TOOLS	843
MEDICAL	463
BUSINESS	460
PRODUCTIVITY	424
PERSONALIZATION	392
COMMUNICATION	387
SPORTS	384
LIFESTYLE	382
FINANCE	366
HEALTH_AND_FITNESS	341
PHOTOGRAPHY	335
SOCIAL	295
NEWS_AND_MAGAZINES	283
SHOPPING	260
TRAVEL_AND_LOCAL	258
DATING	234

BOOKS_AND_REFERENCE	231
VIDEO_PLAYERS	175
EDUCATION	156
ENTERTAINMENT	149
MAPS_AND_NAVIGATION	137
FOOD_AND_DRINK	127
HOUSE_AND_HOME	88
LIBRARIES_AND_DEMO	85
AUTO_AND_VEHICLES	85
WEATHER	82
ART_AND_DESIGN	65
EVENTS	64
PARENTING	60
COMICS	60
BEAUTY	53
1.9	1

Name: count, dtype: int64

Which category getting Avg Highest Rating

Showing Columns

```
df.columns
```

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

Showing the Category wise avg rating here

```
df.groupby('Category')['Rating'].mean()
```

.groupby() is used to grouping the data and applying the Mean() aggregation

.mean() returns the average value

Category	
1.9	19.000000
ART_AND_DESIGN	4.358065
AUTO_AND_VEHICLES	4.190411
BEAUTY	4.278571
BOOKS_AND_REFERENCE	4.346067
BUSINESS	4.121452
COMICS	4.155172
COMMUNICATION	4.158537

DATING	3.970769
EDUCATION	4.389032
ENTERTAINMENT	4.126174
EVENTS	4.435556
FAMILY	4.192272
FINANCE	4.131889
FOOD_AND_DRINK	4.166972
GAME	4.286326
HEALTH_AND_FITNESS	4.277104
HOUSE_AND_HOME	4.197368
LIBRARIES_AND_DEMO	4.178462
LIFESTYLE	4.094904
MAPS_AND_NAVIGATION	4.051613
MEDICAL	4.189143
NEWS_AND_MAGAZINES	4.132189
PARENTING	4.300000
PERSONALIZATION	4.335987
PHOTOGRAPHY	4.192114
PRODUCTIVITY	4.211396
SHOPPING	4.259664
SOCIAL	4.255598
SPORTS	4.223511
TOOLS	4.047411
TRAVEL_AND_LOCAL	4.109292
VIDEO_PLAYERS	4.063750
WEATHER	4.244000

Name: Rating, dtype: float64

Show the List of Content Category

```
df['Content Rating'].unique().tolist()
```

```
['Everyone',
 'Teen',
 'Everyone 10+',
 'Mature 17+',
 'Adults only 18+',
 'Unrated',
 nan]
```

Showing the Category wise avg rating here

```
df.groupby('Category')['Rating'].mean().sort_values(ascending=False)
```

.groupby() is used to grouping the data and applying the Mean() aggregation

```
# .mean() returns the average value
# .sort_values(ascending=False) used to sort the data in ASC/DESC
order
```

Category	
1.9	19.000000
EVENTS	4.435556
EDUCATION	4.389032
ART_AND_DESIGN	4.358065
BOOKS_AND_REFERENCE	4.346067
PERSONALIZATION	4.335987
PARENTING	4.300000
GAME	4.286326
BEAUTY	4.278571
HEALTH_AND_FITNESS	4.277104
SHOPPING	4.259664
SOCIAL	4.255598
WEATHER	4.244000
SPORTS	4.223511
PRODUCTIVITY	4.211396
HOUSE_AND_HOME	4.197368
FAMILY	4.192272
PHOTOGRAPHY	4.192114
AUTO_AND_VEHICLES	4.190411
MEDICAL	4.189143
LIBRARIES_AND_DEMO	4.178462
FOOD_AND_DRINK	4.166972
COMMUNICATION	4.158537
COMICS	4.155172
NEWS_AND_MAGAZINES	4.132189
FINANCE	4.131889
ENTERTAINMENT	4.126174
BUSINESS	4.121452
TRAVEL_AND_LOCAL	4.109292
LIFESTYLE	4.094904
VIDEO_PLAYERS	4.063750
MAPS_AND_NAVIGATION	4.051613
TOOLS	4.047411
DATING	3.970769

Name: Rating, dtype: float64

List the Total Number of App having 5 rating

```
# Applying here condition on rating Column
```

```
df[df['Rating']== 5.0]
```

Reviews \	App	Category	Rating				
329	Hojiboy Tojiboyev Life Hacks	COMICS	5.0				
15							
612	American Girls Mobile Numbers	DATING	5.0				
5							
615	Awake Dating	DATING	5.0				
2							
633	Spine- The dating app	DATING	5.0				
5							
636	Girls Live Talk - Free Text and Video Chat	DATING	5.0				
6							
...				
...							
10721	Mad Dash Fo' Cash	GAME	5.0				
14							
10742	GKPB FP Online Church	LIFESTYLE	5.0				
32							
10776	Monster Ride Pro	GAME	5.0				
1							
10820	Fr. Daoud Lamei	FAMILY	5.0				
22							
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0				
4							
Size	Installs	Type	Price	Content	Rating	Genres	Last
Updated \							
329	37M	1,000+	Free	0	Everyone	Comics	June
26, 2018							
612	4.4M	1,000+	Free	0	Mature 17+	Dating	July
17, 2018							
615	70M	100+	Free	0	Mature 17+	Dating	July
24, 2018							
633	9.3M	500+	Free	0	Teen	Dating	July
14, 2018							
636	5.0M	100+	Free	0	Mature 17+	Dating	August
1, 2018							
...
...							
10721	16M	100+	Free	0	Everyone	Arcade	June
19, 2017							
10742	7.9M	1,000+	Free	0	Everyone	Lifestyle	December
31, 2017							
10776	24M	10+	Free	0	Everyone	Racing	March
5, 2018							
10820	8.6M	1,000+	Free	0	Teen	Education	June
27, 2018							
10837	3.6M	100+	Free	0	Everyone	Education	July
6, 2018							

	Current Ver	Android Ver
329	2.0	4.0.3 and up
612	3.0	4.0.3 and up
615	2.2.9	4.4 and up
633	4.0	4.0.3 and up
636	8.2	4.0.3 and up
...
10721	2.5a	4.1 and up
10742	0.7.1	4.4 and up
10776	2.0	2.3 and up
10820	3.8.0	4.1 and up
10837	1.0	4.1 and up

[274 rows x 13 columns]

We can see above that result has 274 rows, means 274 app has 5.0 Rating

Find Average Value of Review

showing Columns

```
df.columns
```

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

Showing the Reviews Column data

```
df['Reviews']
```

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

```
...
10836     38
10837      4
10838      3
10839    114
10840  398307
```

```
Name: Reviews, Length: 10841, dtype: object
```



```
# Checking the Datatype of the column
```

```
df['Reviews'].dtype  
dtype('O')
```

we can see that Review Column has O data type lets convert it to Numeric

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

Reviews \	App Category	Rating
10472 Life Made WI-Fi Touchscreen Photo Frame 3.0M	1.9	19.0

Genres \	Size	Installs	Type	Price	Content	Rating
10472 1,000+ 2018	Free	0	Everyone	NaN	February 11,	

	Last Updated	Current Ver	Android Ver	Ver
10472	1.0.19	4.0 and up	NaN	

Replacing the value

```
# Replacing the M from the review column value
```

```
df['Reviews'] = df['Reviews'].replace('3.0M', 3.0)  
print('Value replaced !!')  
Value replaced !!
```

Changing the datatype

```
df['Reviews'] = df['Reviews'].astype('float')  
print('datatype Chnaged !!')  
datatype Chnaged !!
```

checking data type

```
df.dtypes  
App          object  
Category     object
```

```

Rating          float64
Reviews         float64
Size            object
Installs        object
Type            object
Price           object
Content Rating  object
Genres          object
Last Updated    object
Current Ver     object
Android Ver     object
dtype: object

```

Extracting the AVerage valuefrom the Reviews Column data

```
df['Reviews'].mean()
```

.mean() returns the Average

```
444111.9265750392
```

Find the Free & Paid Apps

showing Columns

```
df.columns
```

```

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

```

showing data

```
df.head()
```

	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

4.3

	Reviews	Size	Installs	Type	Price	Content Rating	\
0	159.0	19M	10,000+	Free	0	Everyone	
1	967.0	14M	500,000+	Free	0	Everyone	
2	87510.0	8.7M	5,000,000+	Free	0	Everyone	
3	215644.0	25M	50,000,000+	Free	0	Teen	
4	967.0	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

we can see that we've Type column that contain the Free and Paid Value,

Let's extract the result as per our requirments

```
df['Type'].value_counts()

# .value_counts() return the Total count of Each unique data from the
given dataset

Type
Free    10039
Paid      800
0         1
Name: count, dtype: int64
```

Which App has Maximum reviews ?

```
# Lets find the Max Review

df['Reviews'].max()

78158306.0
```

```
# Lets extract the record of App by the Max Review
```

```
df[df['Reviews'] == df['Reviews'].max()]
```

	App	Category	Rating	Reviews	Size	\
2544	Facebook	SOCIAL	4.1	78158306.0	Varies with device	

	Installs	Type	Price	Content Rating	Genres	Last Updated	\
2544	1,000,000,000+	Free	0	Teen	Social	August 3, 2018	

	Current Ver	Android Ver
2544	Varies with device	Varies with device

Display the Top 5 Apps Having Highest Reviews ?

```
# Showing Review Column data and sorting them in desc order
```

```
df['Reviews'].sort_values(ascending=False)
```

2544	78158306.0
3943	78128208.0
381	69119316.0
336	69119316.0
3904	69109672.0

	...
9713	0.0
8495	0.0
6277	0.0
8489	0.0
5086	0.0

```
Name: Reviews, Length: 10841, dtype: float64
```

```
index = df['Reviews'].sort_values(ascending=False).head().index
```

```
df.iloc[index]['App']
```

2544	Facebook
3943	Facebook
381	WhatsApp Messenger
336	WhatsApp Messenger
3904	WhatsApp Messenger

```
Name: App, dtype: object
```

Find the Average rating OF Free & Paid App

```
df.columns
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs',
      'Type',
      'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current
Ver',
      'Android Ver'],
      dtype='object')
```

Grouping the Type Column for applying aggregation

```
df.groupby('Type')['Rating'].mean()
```

```
Type
0      19.000000
Free    4.186203
Paid    4.266615
Name: Rating, dtype: float64
```

Display The Top 5 Apps Maximum Installed

```
df.head(2)
```

	App	Category
Rating \		
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN
4.1		
1	Coloring book moana	ART_AND_DESIGN
3.9		

	Reviews	Size	Installs	Type	Price	Content Rating \
0	159.0	19M	10,000+	Free	0	Everyone
1	967.0	14M	500,000+	Free	0	Everyone

	Genres	Last Updated	Current Ver	Android
Ver				
0	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

Check data type of Install Column

```
df['Installs'].dtypes
```

```
dtype('O')
```

```
# Replave the , & + form the data value
```

```
df['Installs_1'] = df['Installs'].str.replace(',', '')
```

```
print(', Replaced')
```

```
, Replaced
```

```
df.head(2)
```

	App	Category
Rating \		
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN
4.1		
1	Coloring book moana	ART_AND_DESIGN
3.9		

	Reviews	Size	Installs	Type	Price	Content	Rating \
0	159.0	19M	10,000+	Free	0		Everyone
1	967.0	14M	500,000+	Free	0		Everyone

	Genres	Last Updated	Current Ver	Android
Ver \				
0	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

	Installs_1
0	10000+
1	500000+

```
df['Installs_1'] = df['Installs_1'].str.replace('+', '')
```

```
print('+ Replaced')
```

```
+ Replaced
```

```
df.head(2)
```

	App	Category
Rating \		
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN
4.1		
1	Coloring book moana	ART_AND_DESIGN
3.9		

	Reviews	Size	Installs	Type	Price	Content	Rating \
--	---------	------	----------	------	-------	---------	----------

0	159.0	19M	10,000+	Free	0	Everyone
1	967.0	14M	500,000+	Free	0	Everyone

	Genres	Last Updated	Current Ver	Android
Ver \				
0	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

	Installs_1
0	10000
1	500000

lets Replace

```
df[df['Installs_1']=='Free']
```

		App Category	Rating
Reviews \			
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0
3.0			

	Size	Installs	Type	Price	Content	Rating
Genres \						
10472	1,000+	Free	0	Everyone	NaN	February 11, 2018

	Last Updated	Current Ver	Android Ver	Installs_1
10472	1.0.19	4.0 and up	NaN	Free

```
df['Installs_1'] = df['Installs_1'].str.replace('Free','0')
```

```
print('Free Replaced by 0')
```

```
print('Data Replaced')
```

```
Free Replaced by 0
```

```
Data Replaced
```

convert the data type

```
df['Installs_1'] = df['Installs_1'].astype('int')
```

```
print('data type changed')
```

```
data type changed
```

Checking Datatypes

```
df.dtypes
```

```
App          object
Category     object
Rating       float64
Reviews      float64
Size         object
Installs     object
Type         object
Price        object
Content Rating object
Genres       object
Last Updated object
Current Ver  object
Android Ver  object
Installs_1   int32
dtype: object
```

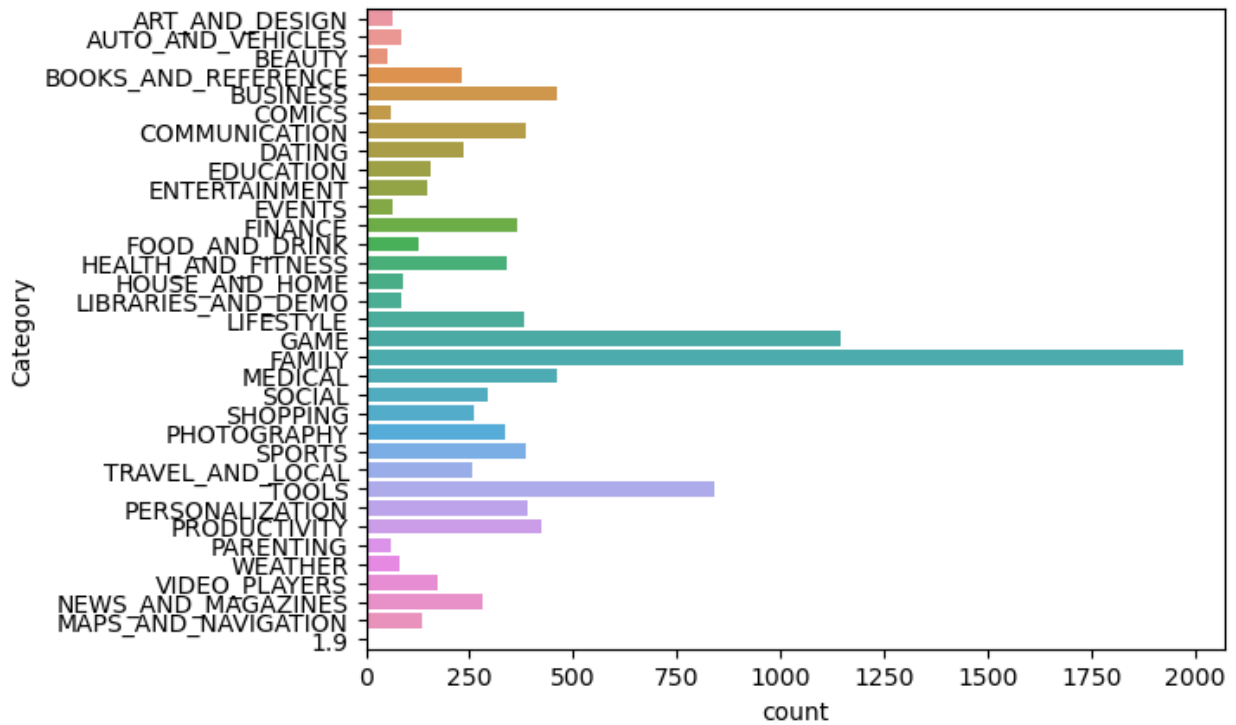
```
index = df['Installs_1'].sort_values(ascending=False).head().index
index
```

```
Index([3896, 3943, 335, 3523, 3565], dtype='int64')
```

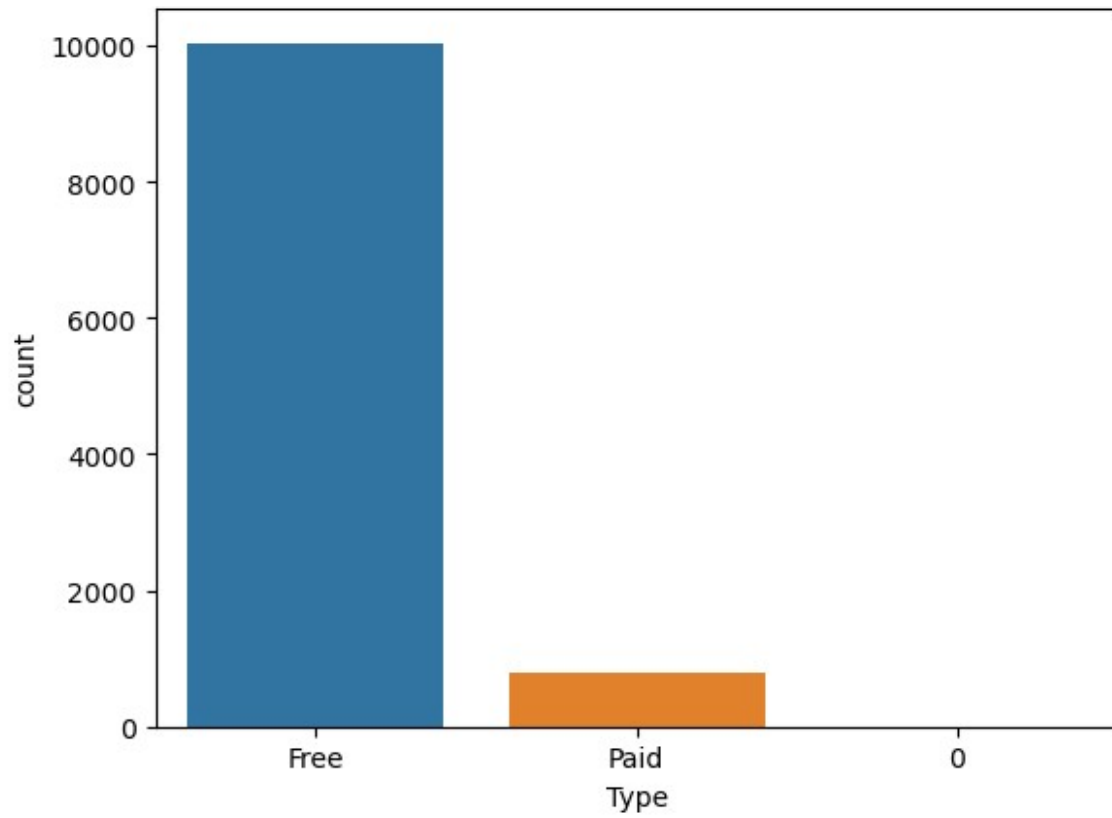
```
df.iloc[index]['App']
```

```
3896          Subway Surfers
3943          Facebook
335    Messenger – Text and Video Chat for Free
3523          Google Drive
3565          Google Drive
Name: App, dtype: object
```

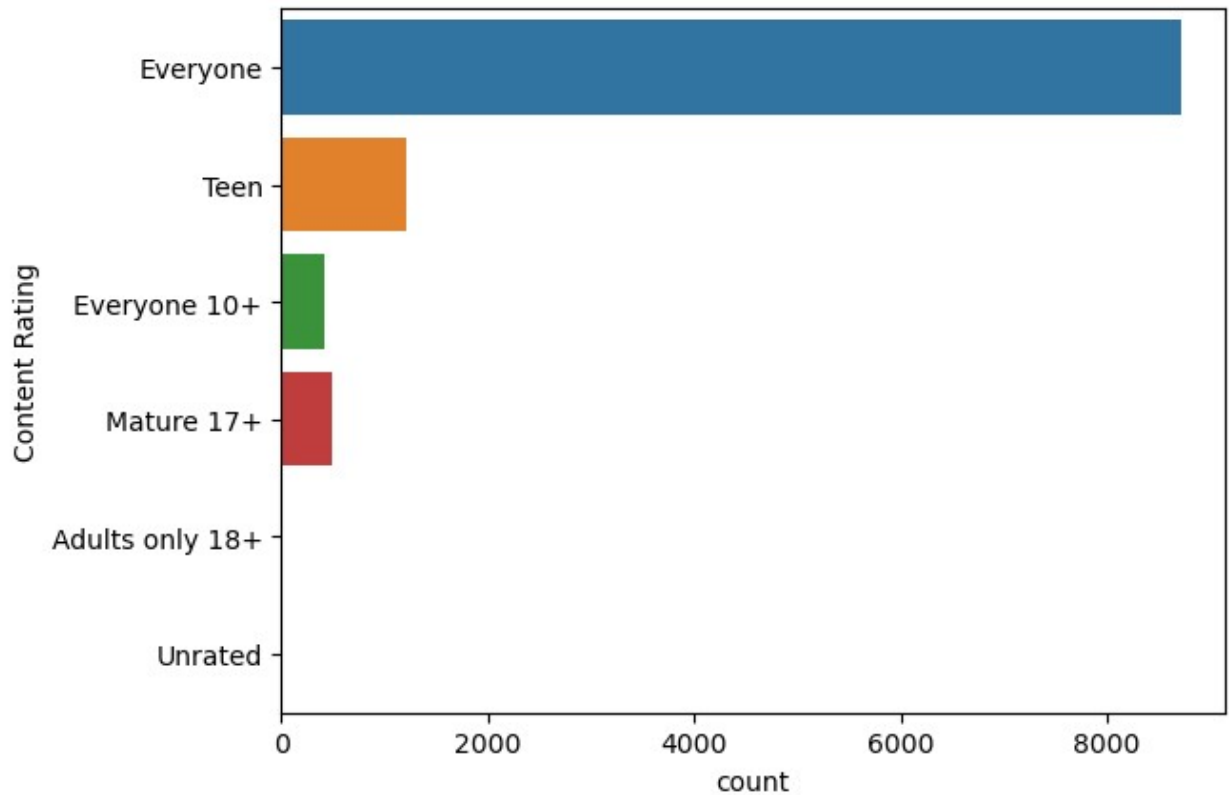
```
sns.countplot(data=df,y='Category');
```

```
sns.countplot(data=df,x='Type');
```



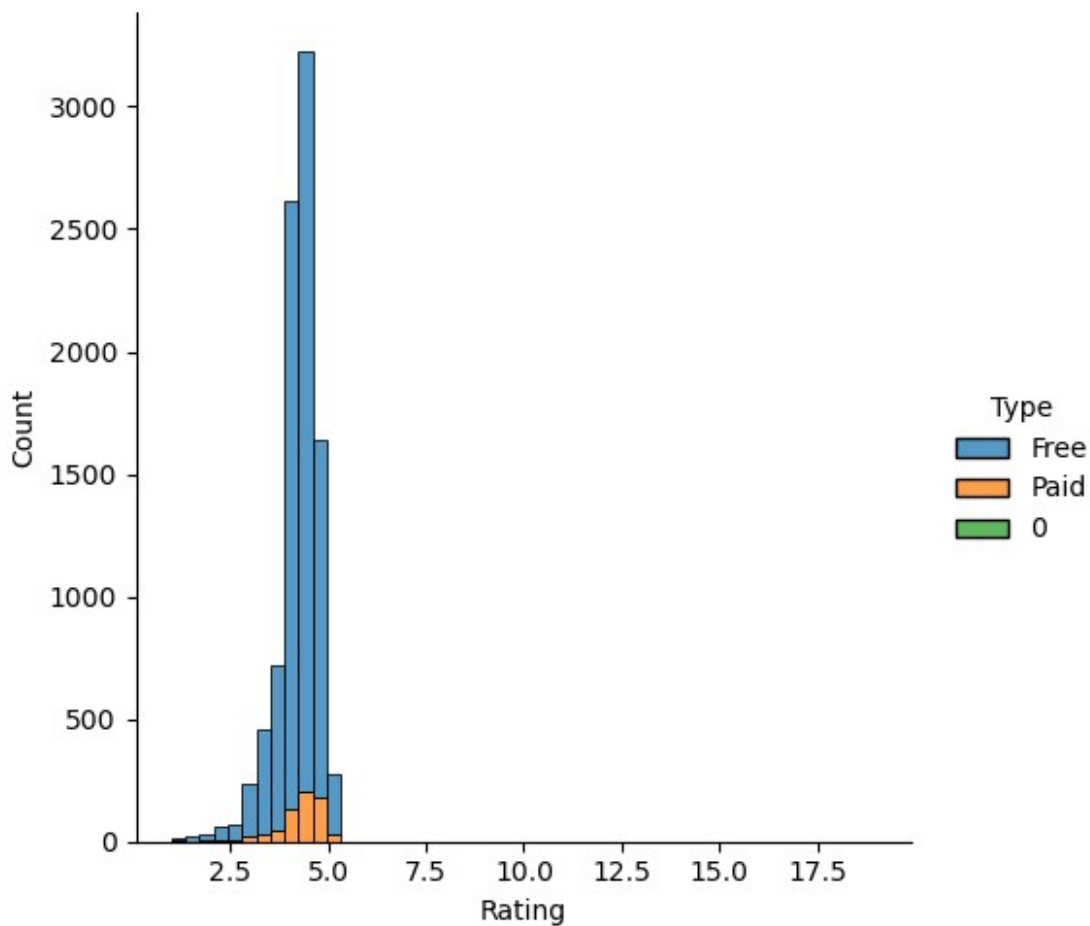
```
sns.countplot(data=df,y='Content Rating');
```



```
##
colors = sns.color_palette()
sns.displot(data=df, x='Rating', hue='Type', bins=50, color=colors[:2], multiple='stack')
plt.show()
```

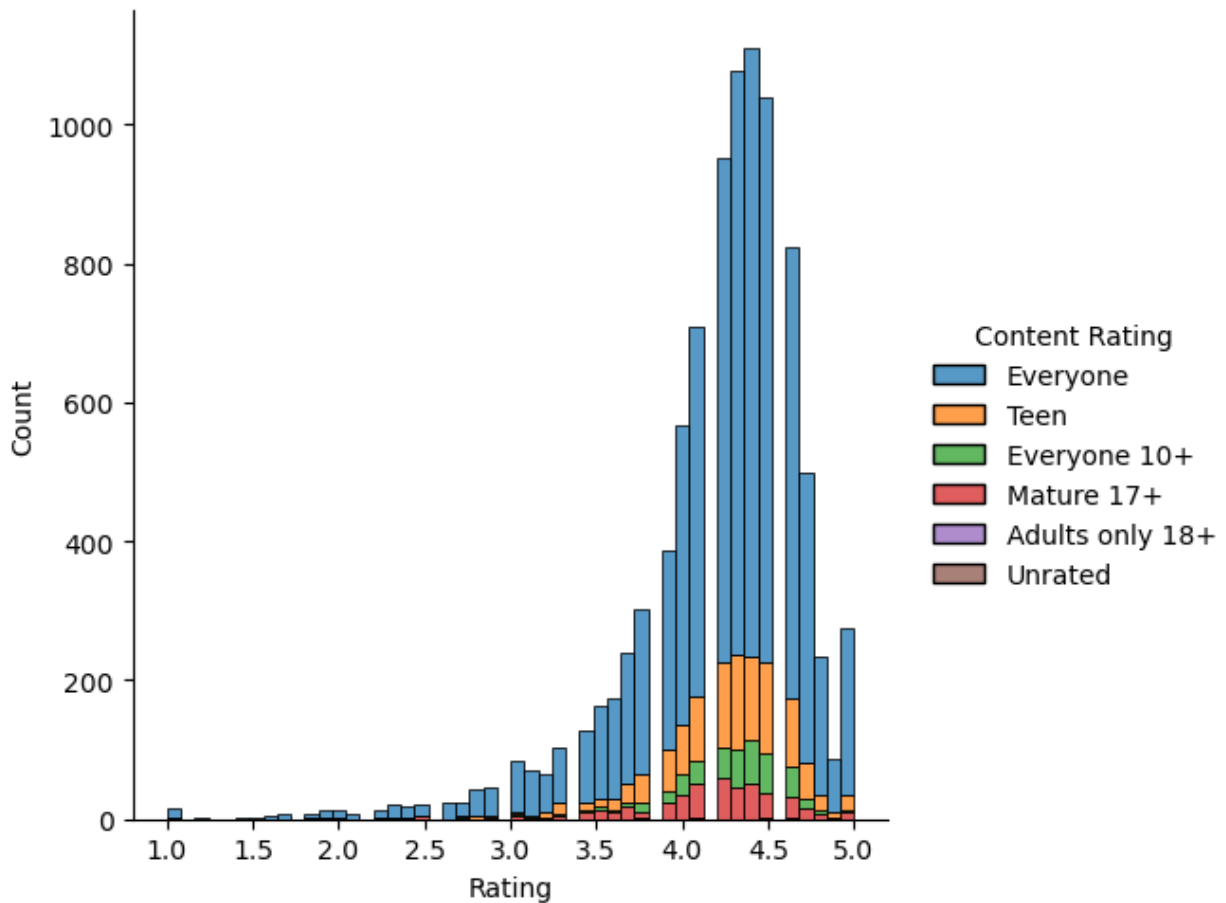
```
C:\Users\180329775\AppData\Local\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
```

```
    with pd.option_context('mode.use_inf_as_na', True):
```



```
colors = sns.color_palette()
sns.displot(data=df, x='Rating', hue='Content
Rating', bins=50, color=colors[:2], multiple='stack')
plt.show()
```

```
C:\Users\180329775\AppData\Local\anaconda3\Lib\site-packages\seaborn\
_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated
and will be removed in a future version. Convert inf values to NaN
before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



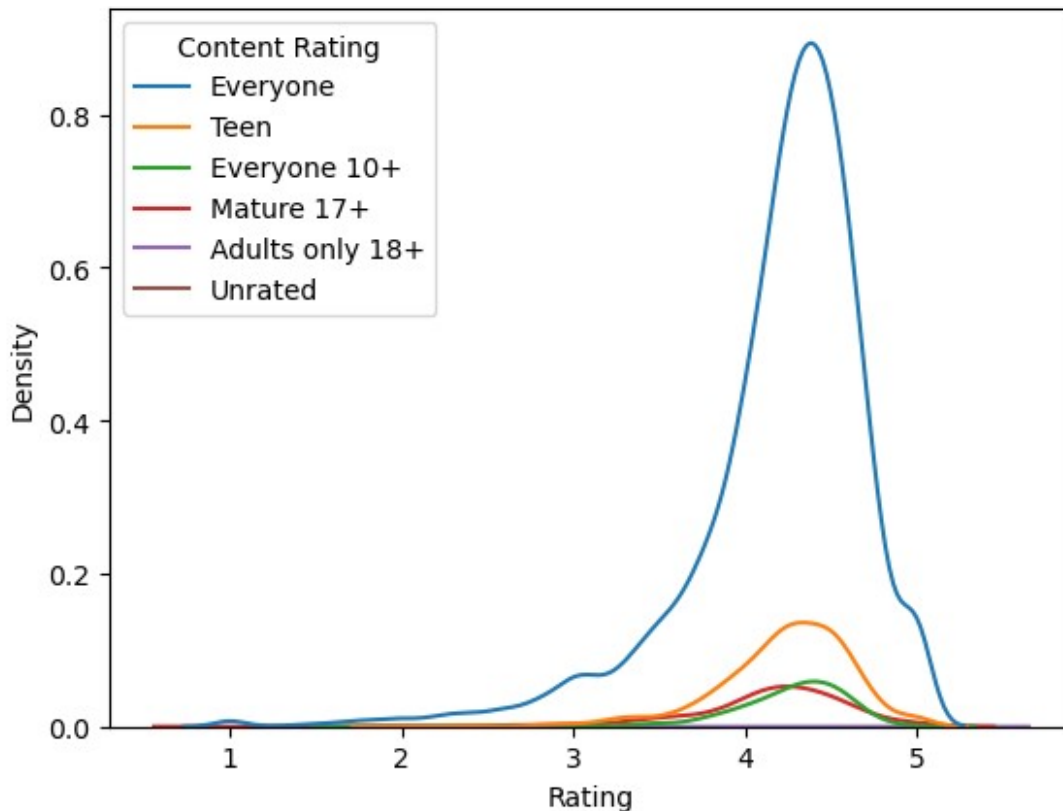
```
colors = sns.color_palette()
sns.kdeplot(data=df,x='Rating',hue='Content Rating',color=colors[:2])
plt.show()
```

C:\Users\180329775\AppData\Local\anaconda3\Lib\site-packages\seaborn_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.

```
with pd.option_context('mode.use_inf_as_na', True):
```

C:\Users\180329775\AppData\Local\Temp\ipykernel_15392\852677419.py:2: UserWarning: Dataset has 0 variance; skipping density estimate. Pass `warn_singular=False` to disable this warning.

```
sns.kdeplot(data=df,x='Rating',hue='Content
Rating',color=colors[:2])
```



1 - What is the most expensive app on the store?

```
df[df['Price'] == df['Price'].max()]['App'].iloc[0]
```

'Life Made WI-Fi Touchscreen Photo Frame'

2 - How many apps that has more than 50K reviews?

```
df.columns
```

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs',  
      'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current  
Ver', 'Android Ver', 'Installs_1'],  
      dtype='object')
```

```
df[ df['Reviews'] > 50000 ]
```

						App	
Category \							
2	U Launcher Lite – FREE Live Cool Themes, Hide ...						
ART_AND_DESIGN							
3	Sketch - Draw & Paint						
ART_AND_DESIGN							
18	FlipaClip - Cartoon animation						
ART_AND_DESIGN							
19	ibis Paint X						
ART_AND_DESIGN							
42	Textgram - write on photos						
ART_AND_DESIGN							
...		...					
..							
10796	Inf VPN - Global Proxy & Unlimited Free WIFI VPN						
TOOLS							
10803	Fatal Raid - No.1 Mobile FPS						
GAME							
10809	Castle Clash: RPG War and Strategy FR						
FAMILY							
10826	Frim: get new friends on local chat rooms						
SOCIAL							
10840	iHoroscope - 2018 Daily Horoscope & Astrology						
LIFESTYLE							
Price	Rating	Reviews	Size	Installs	Type		
2	4.7	87510.0	8.7M	5,000,000+	Free	0	
3	4.5	215644.0	25M	50,000,000+	Free	0	
18	4.3	194216.0	39M	5,000,000+	Free	0	
19	4.6	224399.0	31M	10,000,000+	Free	0	
42	4.4	295221.0	Varies with device	10,000,000+	Free	0	
...		
10796	4.7	61445.0	7.8M	1,000,000+	Free	0	
10803	4.3	56496.0	81M	1,000,000+	Free	0	
10809	4.7	376223.0	24M	1,000,000+	Free	0	
10826	4.0	88486.0	Varies with device	5,000,000+	Free	0	
10840	4.5	398307.0	19M	10,000,000+	Free	0	
Content Rating		Genres	Last Updated		Current Ver		

55 Tickets SDA 2018 and Exam from the State Traff...
 AUTO_AND_VEHICLES
 61 CDL Practice Test 2018 Edition
 AUTO_AND_VEHICLES
 64 DMV Permit Practice Test 2018 Edition
 AUTO_AND_VEHICLES
 70 Fines of the State Traffic Safety Inspectorate...
 AUTO_AND_VEHICLES
 ...
 ...
 10801 Fr Ignacio Outreach
 FAMILY
 10810 Fr Lupupa Sermons
 BUSINESS
 10820 Fr. Daoud Lamei
 FAMILY
 10833 Chemin (fr)
 BOOKS_AND_REFERENCE
 10837 Fr. Mike Schmitz Audio Teachings
 FAMILY

	Rating	Reviews	Size	Installs	Type	Price	Content
Rating \							
25	4.8	192.0	6.0M	10,000+	Free	0	Everyone
55	4.9	10479.0	33M	100,000+	Free	0	Everyone
61	4.9	7774.0	17M	100,000+	Free	0	Everyone
64	4.9	6090.0	27M	100,000+	Free	0	Everyone
70	4.8	116986.0	35M	5,000,000+	Free	0	Everyone
...
10801	4.9	52.0	19M	1,000+	Free	0	Everyone
10810	4.8	19.0	21M	100+	Free	0	Everyone
10820	5.0	22.0	8.6M	1,000+	Free	0	Teen
10833	4.8	44.0	619k	1,000+	Free	0	Everyone
10837	5.0	4.0	3.6M	100+	Free	0	Everyone

	Genres	Last Updated	Current Ver	Android
Ver \				
25	Art & Design	April 25, 2018	1.5	3.0 and up
55	Auto & Vehicles	July 18, 2018	1.7.1	4.0 and up

61	Auto & Vehicles	July 3, 2018	1.7	4.2 and up
64	Auto & Vehicles	July 3, 2018	1.7	4.2 and up
70	Auto & Vehicles	August 2, 2018	1.9.7	4.0.3 and up
...
10801	Education	January 19, 2018	1.0	4.4 and up
10810	Business	June 12, 2018	1.0	4.4 and up
10820	Education	June 27, 2018	3.8.0	4.1 and up
10833	Books & Reference	March 23, 2014	0.8	2.2 and up
10837	Education	July 6, 2018	1.0	4.1 and up

	Installs_1
25	10000+
55	100000+
61	100000+
64	100000+
70	5000000+
...	...
10801	1000+
10810	100+
10820	1000+
10833	1000+
10837	100+

[596 rows x 14 columns]

```
df['Price'].dtypes
```

```
dtype('O')
```

changing Data Type of Price Column here

```
df['Price'] = df['Price'].str.replace('$', '')
print('$ replaced !!')
```

```
$ replaced !!
```

```
df['Price_1']
```

0	0
1	0

```

2      0
3      0
4      0
..
10836  0
10837  0
10838  0
10839  0
10840  0
Name: Price_1, Length: 10841, dtype: object

df[df['Price']=='Everyone']

Reviews \
10472  Life Made WI-Fi Touchscreen Photo Frame      1.9      19.0
3.0

Size Installs Type      Price Content Rating
Genres \
10472  1,000+      Free      0  Everyone      NaN  February 11,
2018

Last Updated Current Ver Android Ver Installs_1  Price_1
10472      1.0.19  4.0 and up      NaN      Free  Everyone

df['Price'] = df['Price'].str.replace('Everyone','0')
print('Value replaced !!')
Value replaced !!

df['Price'] = df['Price'].astype('float')
print('Data Type Changed !!')
Data Type Changed !!

df[df['Rating'] > 4.7]['Price'].mean()

0.501006711409396

```

How much google earned from the 5,000,000+ installed apps?

```

df[df['Installs'] == '5,000,000+']['Price'].sum()

0.0

```

Which Genre has more apps?

```
df.head(2)
```

	App	Category
Rating \		
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN
4.1		
1	Coloring book moana	ART_AND_DESIGN
3.9		

	Reviews	Size	Installs	Type	Price	Content Rating \
0	159.0	19M	10,000+	Free	0.0	Everyone
1	967.0	14M	500,000+	Free	0.0	Everyone

	Genres	Last Updated	Current Ver	Android
Ver \				
0	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

	Installs_1	Price_1
0	10000+	0
1	500000+	0

```
df.columns
```

```
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs',  
      'Type', 'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current  
Ver', 'Android Ver', 'Installs_1', 'Price_1'],  
      dtype='object')
```

```
df['Genres'].value_counts(ascending= False)
```

Genres	
Tools	842
Entertainment	623
Education	549
Medical	463
Business	460
...	
Arcade;Pretend Play	1
Card;Brain Games	1
Lifestyle;Pretend Play	1
Comics;Creativity	1
Strategy;Creativity	1
Name: count, Length: 120, dtype: int64	

How many apps that can work on android version 4?

```
df.columns
Index(['App', 'Category', 'Rating', 'Reviews', 'Size', 'Installs',
      'Type',
      'Price', 'Content Rating', 'Genres', 'Last Updated', 'Current
Ver',
      'Android Ver', 'Installs_1', 'Price_1'],
      dtype='object')
```

```
df['Android Ver'].value_counts(ascending=False)
```

```
Android Ver
4.1 and up          2451
4.0.3 and up        1501
4.0 and up          1375
Varies with device  1362
4.4 and up           980
2.3 and up           652
5.0 and up           601
4.2 and up           394
2.3.3 and up         281
2.2 and up           244
4.3 and up           243
3.0 and up           241
2.1 and up           134
1.6 and up           116
6.0 and up            60
7.0 and up            42
3.2 and up            36
2.0 and up            32
5.1 and up            24
1.5 and up            20
4.4W and up           12
3.1 and up            10
2.0.1 and up           7
8.0 and up            6
7.1 and up            3
4.0.3 - 7.1.1         2
5.0 - 8.0              2
1.0 and up            2
7.0 - 7.1.1           1
4.1 - 7.1.1           1
5.0 - 6.0              1
2.2 - 7.1.1           1
5.0 - 7.1.1           1
Name: count, dtype: int64
```

How many apps for each type for each content rating?

```
pd.pivot_table(index='Type',columns='Content Rating',data=df,aggfunc='count')['Rating']
```

Content Rating \ Type	Adults only 18+	Everyone	Everyone 10+	Mature 17+
-----------------------	-----------------	----------	--------------	------------

Free	3.0	6868.0	364.0	444.0
1039.0				
Paid	NaN	552.0	33.0	17.0
45.0				

Content Rating	Unrated
Free	1.0
Paid	NaN

Thank you for your interest and reaching to this point

Regards : Vivek Kr Singh