In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df=pd.read_csv("googleplaystore.csv")
df.head()
```

Out[2]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone

In [3]:

```
df.shape
```

Out[3]:

```
(10841, 13)
```

In [4]:

```
df.info()
# ver stands for version
```

```
<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
```

In [5]:

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

Out[5]:

```
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
```

We need to handle object data type

*dataset need cleaning *change the data type

In [6]:

```
df.describe()
```

Out[6]:

	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

In [7]:

```
# check duplicate values
df.duplicated().sum()
```

Out[7]:

```
483
```

In [8]:

```
#see duplicated dataframes
df[df.duplicated()]
```

Out[8]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Conte Rati
229	Quick PDF Scanner + OCR FREE	BUSINESS	4.2	80805	Varies with device	5,000,000+	Free	0	Everyo
236	Box	BUSINESS	4.2	159872	Varies with device	10,000,000+	Free	0	Everyo
239	Google My Business	BUSINESS	4.4	70991	Varies with device	5,000,000+	Free	0	Everyo
256	ZOOM Cloud Meetings	BUSINESS	4.4	31614	37M	10,000,000+	Free	0	Everyo
261	join.me - Simple Meetings	BUSINESS	4.0	6989	Varies with device	1,000,000+	Free	0	Everyo
...	...	...	...	...	...	...	...	...	
8643	Wunderlist: To-Do List & Tasks	PRODUCTIVITY	4.6	404610	Varies with device	10,000,000+	Free	0	Everyo
8654	TickTick: To Do List with Reminder, Day Planner	PRODUCTIVITY	4.6	25370	Varies with device	1,000,000+	Free	0	Everyo
8658	ColorNote Notepad Notes	PRODUCTIVITY	4.6	2401017	Varies with device	100,000,000+	Free	0	Everyo
10049	Airway Ex - Intubate. Anesthetize. Train.	MEDICAL	4.3	123	86M	10,000+	Free	0	Everyo
10768	AAFP	MEDICAL	3.8	63	24M	10,000+	Free	0	Everyo

483 rows × 13 columns

In [9]:

```
# include all features
df.describe(include="all").T
```

Out[9]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
App	10841	9660	ROBLOX	9	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Category	10841	34	FAMILY	1972	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Rating	9367.0	NaN	NaN	NaN	4.193338	0.537431	1.0	4.0	4.3	4.5	19.0
Reviews	10841	6002	0	596	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Size	10841	462	Varies with device	1695	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Installs	10841	22	1,000,000+	1579	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	10840	3	Free	10039	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Price	10841	93	0	10040	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Content Rating	10840	6	Everyone	8714	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Genres	10841	120	Tools	842	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Last Updated	10841	1378	August 3, 2018	326	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Current Ver	10833	2832	Varies with device	1459	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Android Ver	10838	33	4.1 and up	2451	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [10]:

```
#generate 10 randome sample
df.sample(10)
```

Out[10]:

	App	Category	Rating	Reviews	Size	Installs	Type
9616	Governor of Poker 2 - OFFLINE POKER GAME	GAME	4.3	246538	60M	5,000,000+	Free
5431	virtual lover 3D	FAMILY	4.6	5195	64M	100,000+	Free
1816	Merge Dragons!	GAME	4.5	214777	91M	5,000,000+	Free
9993	EW PDF	BOOKS_AND_REFERENCE	NaN	0	8.7M	5+	Free
7507	CL Pro Client for Craigslist	SHOPPING	3.6	48	2.2M	5,000+	Free
3229	DreamTrips	TRAVEL_AND_LOCAL	4.7	9971	22M	500,000+	Free
5095	AG Subway Simulator Lite	FAMILY	4.4	6738	56M	100,000+	Free
9290	EF Forms	BUSINESS	5.0	2	23M	50+	Free
7495	Night Camera Blur Effect	PHOTOGRAPHY	3.6	100	2.5M	10,000+	Free
2865	Cymera Camera- Photo Editor, Filter,Collage,La...	PHOTOGRAPHY	4.4	2418135	Varies with device	100,000,000+	Free

In [11]:

```
#focused on Reviews column its numeric feature but its given as object
df["Reviews"]
```

Out[11]:

```
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
```

In [12]:

```
df["Reviews"].dtypes
```

Out[12]:

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

In [13]:

```
df["Reviews"].shape
```

Out[13]:

```
(10841,)
```

In [14]:

```
df.Reviews.str.isnumeric().sum()
```

Out[14]:

```
10840
```

In [15]:

```
df ['Reviews'].str.isnumeric().sum()
```

Out[15]:

```
10840
```

In [16]:

```
df ['Reviews'].str.isnumeric()
```

Out[16]:

```
0        True
1        True
2        True
3        True
4        True
         ...
10836    True
10837    True
10838    True
10839    True
10840    True
Name: Reviews, Length: 10841, dtype: bool
```

In [17]:

```
# see the negation
# where the value is numeric it gives false and vice versa
~df ['Reviews'].str.isnumeric()
```

Out[17]:

```
0        False
1        False
2        False
3        False
4        False
         ...
10836    False
10837    False
10838    False
10839    False
10840    False
Name: Reviews, Length: 10841, dtype: bool
```

In [18]:

```
# datatype change from object to int
df.Reviews.str.isnumeric().sum().dtype
```

Out[18]:

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

In [19]:

```python
# gives true dataframes
df[df ['Reviews'].str.isnumeric()]
```

Out[19]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	(
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	(
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	(
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	(
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	(
...	...	...	...	...	...	...	...	..
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	(
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	(
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	(
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	(
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	(

10840 rows × 13 columns

In [20]:

```
#one value is categorical  in int conversion 10841-10840
df[~df["Reviews"].str.isnumeric()]
```

Out[20]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Ge
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	1,000+	Free	0	Everyone	NaN	Febr 11,

**we need to remove above one categorical row

In [21]:

```
#Create a copy of data set
df_copy=df.copy()
```

In [22]:

```
#remove the row with index 1042
df_copy=df_copy.drop(df_copy.index[10472])
```

In [23]:

```
df_copy.shape
```

Out[23]:

```
(10840, 13)
```

In [24]:

```
# datatype is object
df_copy["Reviews"].dtype
```

Out[24]:

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

In [25]:

```
# need to change data type
df_copy["Reviews"]=df_copy["Reviews"].astype('int')

```

In [26]:

```
df_copy.shape
```

Out[26]:

```
(10840, 13)
```

In [27]:

```
# datatype changed
df_copy["Reviews"].dtype
```

Out[27]:

```
dtype('int32')
```

In [28]:

```
df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10840 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10840 non-null  object 
 1   Category        10840 non-null  object 
 2   Rating          9366 non-null   float64
 3   Reviews         10840 non-null  int32  
 4   Size            10840 non-null  object 
 5   Installs        10840 non-null  object 
 6   Type            10839 non-null  object 
 7   Price           10840 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10840 non-null  object 
 10  Last Updated    10840 non-null  object 
 11  Current Ver     10832 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(1), int32(1), object(11)
memory usage: 1.1+ MB
```

Consider size feature

dtype is object

- find unique sizes

In [29]:

```
df_copy['Size']
```

Out[29]:

```
0                       19M
1                       14M
2                      8.7M
3                       25M
4                      2.8M
                ...        
10836                   53M
10837                  3.6M
10838                  9.5M
10839    Varies with device
10840                   19M
Name: Size, Length: 10840, dtype: object
```

In [30]:

```
df_copy.duplicated().sum()
```

Out[30]:

```
483
```

In [31]:

```
# find unique sizes
df_copy['Size'].unique()
```

Out[31]:

```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M', '3.1M',
       '28M', '12M', '20M', '21M', '37M', '2.7M', '5.5M', '17M', '39M',
       '31M', '4.2M', '7.0M', '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', '2.5M', '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', '3.7M', '22M', '7.4M', '6.4M', '3.2M', '8.2M', '9.9M',
       '4.9M', '9.5M', '5.0M', '5.9M', '13M', '73M', '6.8M', '3.5M',
       '4.0M', '2.3M', '7.2M', '2.1M', '42M', '7.3M', '9.1M', '55M',
       '23k', '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', '118k', '44M', '695k', '1.6M',
       '6.2M', '18k', '53M', '1.4M', '3.0M', '5.8M', '3.8M', '9.6M',
       '45M', '63M', '49M', '77M', '4.4M', '4.8M', '70M', '6.9M', '9.3M',
       '10.0M', '8.1M', '36M', '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', '34M', '93M', '65M', '79M',
       '100M', '58M', '50M', '68M', '64M', '67M', '60M', '94M', '232k',
       '99M', '624k', '95M', '8.5k', '41k', '292k', '11k', '80M', '1.7M',
       '74M', '62M', '69M', '75M', '98M', '85M', '82M', '96M', '87M',
       '71M', '86M', '91M', '81M', '92M', '83M', '88M', '704k', '862k',
       '899k', '378k', '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', '902k', '818k',
       '81k', '939k', '169k', '45k', '475k', '965k', '90M', '545k', '61k',
       '283k', '655k', '714k', '93k', '872k', '121k', '322k', '1.0M',
       '976k', '172k', '238k', '549k', '206k', '954k', '444k', '717k',
       '210k', '609k', '308k', '705k', '306k', '904k', '473k', '175k',
       '350k', '383k', '454k', '421k', '70k', '812k', '442k', '842k',
       '417k', '412k', '459k', '478k', '335k', '782k', '721k', '430k',
       '429k', '192k', '200k', '460k', '728k', '496k', '816k', '414k',
       '506k', '887k', '613k', '243k', '569k', '778k', '683k', '592k',
       '319k', '186k', '840k', '647k', '191k', '373k', '437k', '598k',
       '716k', '585k', '982k', '222k', '219k', '55k', '948k', '323k',
       '691k', '511k', '951k', '963k', '25k', '554k', '351k', '27k',
       '82k', '208k', '913k', '514k', '551k', '29k', '103k', '898k',
       '743k', '116k', '153k', '209k', '353k', '499k', '173k', '597k',
       '809k', '122k', '411k', '400k', '801k', '787k', '237k', '50k',
       '643k', '986k', '97k', '516k', '837k', '780k', '961k', '269k',
       '20k', '498k', '600k', '749k', '642k', '881k', '72k', '656k',
       '601k', '221k', '228k', '108k', '940k', '176k', '33k', '663k',
       '34k', '942k', '259k', '164k', '458k', '245k', '629k', '28k',
       '288k', '775k', '785k', '636k', '916k', '994k', '309k', '485k',
       '914k', '903k', '608k', '500k', '54k', '562k', '847k', '957k',
       '688k', '811k', '270k', '48k', '329k', '523k', '921k', '874k',
       '981k', '784k', '280k', '24k', '518k', '754k', '892k', '154k',
       '860k', '364k', '387k', '626k', '161k', '879k', '39k', '970k',
       '170k', '141k', '160k', '144k', '143k', '190k', '376k', '193k',
       '246k', '73k', '658k', '992k', '253k', '420k', '404k', '470k',
       '226k', '240k', '89k', '234k', '257k', '861k', '467k', '157k',
       '44k', '676k', '67k', '552k', '885k', '1020k', '582k', '619k'],
      dtype=object)
```

sizes in M byte and K byte

- need to do conversion

1M = 1024 K so

- 19M =19 X 1024 k replace all into k

write loop for conversion or use replace function on frame

In [32]:

```
# replace M
df_copy['Size']= df_copy['Size'].str.replace("M","000")

```

In [33]:

```
# see changes
df_copy['Size']

```

Out[33]:

```
0                      19000
1                      14000
2                     8.7000
3                      25000
4                     2.8000
                ...
10836                  53000
10837                 3.6000
10838                 9.5000
10839     Varies with device
10840                  19000
Name: Size, Length: 10840, dtype: object
```

In [34]:

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

Out[34]:

```
array(['19000', '14000', '8.7000', '25000', '2.8000', '5.6000', '29000',
       '33000', '3.1000', '28000', '12000', '20000', '21000', '37000',
       '2.7000', '5.5000', '17000', '39000', '31000', '4.2000', '7.0000',
       '23000', '6.0000', '6.1000', '4.6000', '9.2000', '5.2000', '11000',
       '24000', 'Varies with device', '9.4000', '15000', '10000',
       '1.2000', '26000', '8.0000', '7.9000', '56000', '57000', '35000',
       '54000', '201k', '3.6000', '5.7000', '8.6000', '2.4000', '27000',
       '2.5000', '16000', '3.4000', '8.9000', '3.9000', '2.9000', '38000',
       '32000', '5.4000', '18000', '1.1000', '2.2000', '4.5000', '9.8000',
       '52000', '9.0000', '6.7000', '30000', '2.6000', '7.1000', '3.7000',
       '22000', '7.4000', '6.4000', '3.2000', '8.2000', '9.9000',
       '4.9000', '9.5000', '5.0000', '5.9000', '13000', '73000', '6.8000',
       '3.5000', '4.0000', '2.3000', '7.2000', '2.1000', '42000',
       '7.3000', '9.1000', '55000', '23k', '6.5000', '1.5000', '7.5000',
       '51000', '41000', '48000', '8.5000', '46000', '8.3000', '4.3000',
       '4.7000', '3.3000', '40000', '7.8000', '8.8000', '6.6000',
       '5.1000', '61000', '66000', '79k', '8.4000', '118k', '44000',
       '695k', '1.6000', '6.2000', '18k', '53000', '1.4000', '3.0000',
       '5.8000', '3.8000', '9.6000', '45000', '63000', '49000', '77000',
       '4.4000', '4.8000', '70000', '6.9000', '9.3000', '10.0000',
       '8.1000', '36000', '84000', '97000', '2.0000', '1.9000', '1.8000',
       '5.3000', '47000', '556k', '526k', '76000', '7.6000', '59000',
       '9.7000', '78000', '72000', '43000', '7.7000', '6.3000', '334k',
       '34000', '93000', '65000', '79000', '100000', '58000', '50000',
       '68000', '64000', '67000', '60000', '94000', '232k', '99000',
       '624k', '95000', '8.5k', '41k', '292k', '11k', '80000', '1.7000',
       '74000', '62000', '69000', '75000', '98000', '85000', '82000',
       '96000', '87000', '71000', '86000', '91000', '81000', '92000',
       '83000', '88000', '704k', '862k', '899k', '378k', '266k', '375k',
       '1.3000', '975k', '980k', '4.1000', '89000', '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', '902k', '818k', '81k', '939k', '169k',
       '45k', '475k', '965k', '90000', '545k', '61k', '283k', '655k',
       '714k', '93k', '872k', '121k', '322k', '1.0000', '976k', '172k',
       '238k', '549k', '206k', '954k', '444k', '717k', '210k', '609k',
       '308k', '705k', '306k', '904k', '473k', '175k', '350k', '383k',
       '454k', '421k', '70k', '812k', '442k', '842k', '417k', '412k',
       '459k', '478k', '335k', '782k', '721k', '430k', '429k', '192k',
       '200k', '460k', '728k', '496k', '816k', '414k', '506k', '887k',
       '613k', '243k', '569k', '778k', '683k', '592k', '319k', '186k',
       '840k', '647k', '191k', '373k', '437k', '598k', '716k', '585k',
       '982k', '222k', '219k', '55k', '948k', '323k', '691k', '511k',
       '951k', '963k', '25k', '554k', '351k', '27k', '82k', '208k',
       '913k', '514k', '551k', '29k', '103k', '898k', '743k', '116k',
       '153k', '209k', '353k', '499k', '173k', '597k', '809k', '122k',
       '411k', '400k', '801k', '787k', '237k', '50k', '643k', '986k',
       '97k', '516k', '837k', '780k', '961k', '269k', '20k', '498k',
       '600k', '749k', '642k', '881k', '72k', '656k', '601k', '221k',
       '228k', '108k', '940k', '176k', '33k', '663k', '34k', '942k',
       '259k', '164k', '458k', '245k', '629k', '28k', '288k', '775k',
       '785k', '636k', '916k', '994k', '309k', '485k', '914k', '903k',
       '608k', '500k', '54k', '562k', '847k', '957k', '688k', '811k',
       '270k', '48k', '329k', '523k', '921k', '874k', '981k', '784k',
       '280k', '24k', '518k', '754k', '892k', '154k', '860k', '364k',
       '387k', '626k', '161k', '879k', '39k', '970k', '170k', '141k',
       '160k', '144k', '143k', '190k', '376k', '193k', '246k', '73k',
       '658k', '992k', '253k', '420k', '404k', '470k', '226k', '240k',
       '89k', '234k', '257k', '861k', '467k', '157k', '44k', '676k',
       '67k', '552k', '885k', '1020k', '582k', '619k'], dtype=object)
```

In [35]:

```
# replace k
df_copy['Size']= df_copy['Size'].str.replace("k","")
df_copy['Size']
```

Out[35]:

```
0                      19000
1                      14000
2                     8.7000
3                      25000
4                     2.8000
                ...
10836                  53000
10837                 3.6000
10838                 9.5000
10839     Varies with device
10840                  19000
Name: Size, Length: 10840, dtype: object
```

In [36]:

```
# check unique values
df_copy['Size'].unique()
```

Out[36]:

```
array(['19000', '14000', '8.7000', '25000', '2.8000', '5.6000', '29000',
       '33000', '3.1000', '28000', '12000', '20000', '21000', '37000',
       '2.7000', '5.5000', '17000', '39000', '31000', '4.2000', '7.0000',
       '23000', '6.0000', '6.1000', '4.6000', '9.2000', '5.2000', '11000',
       '24000', 'Varies with device', '9.4000', '15000', '10000',
       '1.2000', '26000', '8.0000', '7.9000', '56000', '57000', '35000',
       '54000', '201', '3.6000', '5.7000', '8.6000', '2.4000', '27000',
       '2.5000', '16000', '3.4000', '8.9000', '3.9000', '2.9000', '38000',
       '32000', '5.4000', '18000', '1.1000', '2.2000', '4.5000', '9.8000',
       '52000', '9.0000', '6.7000', '30000', '2.6000', '7.1000', '3.7000',
       '22000', '7.4000', '6.4000', '3.2000', '8.2000', '9.9000',
       '4.9000', '9.5000', '5.0000', '5.9000', '13000', '73000', '6.8000',
       '3.5000', '4.0000', '2.3000', '7.2000', '2.1000', '42000',
       '7.3000', '9.1000', '55000', '23', '6.5000', '1.5000', '7.5000',
       '51000', '41000', '48000', '8.5000', '46000', '8.3000', '4.3000',
       '4.7000', '3.3000', '40000', '7.8000', '8.8000', '6.6000',
       '5.1000', '61000', '66000', '79', '8.4000', '118', '44000', '695',
       '1.6000', '6.2000', '18', '53000', '1.4000', '3.0000', '5.8000',
       '3.8000', '9.6000', '45000', '63000', '49000', '77000', '4.4000',
       '4.8000', '70000', '6.9000', '9.3000', '10.0000', '8.1000',
       '36000', '84000', '97000', '2.0000', '1.9000', '1.8000', '5.3000',
       '47000', '556', '526', '76000', '7.6000', '59000', '9.7000',
       '78000', '72000', '43000', '7.7000', '6.3000', '334', '34000',
       '93000', '65000', '79000', '100000', '58000', '50000', '68000',
       '64000', '67000', '60000', '94000', '232', '99000', '624', '95000',
       '8.5', '41', '292', '11', '80000', '1.7000', '74000', '62000',
       '69000', '75000', '98000', '85000', '82000', '96000', '87000',
       '71000', '86000', '91000', '81000', '92000', '83000', '88000',
       '704', '862', '899', '378', '266', '375', '1.3000', '975', '980',
       '4.1000', '89000', '696', '544', '525', '920', '779', '853', '720',
       '713', '772', '318', '58', '241', '196', '857', '51', '953', '865',
       '251', '930', '540', '313', '746', '203', '26', '314', '239',
       '371', '220', '730', '756', '91', '293', '17', '74', '14', '317',
       '78', '924', '902', '818', '81', '939', '169', '45', '475', '965',
       '90000', '545', '61', '283', '655', '714', '93', '872', '121',
       '322', '1.0000', '976', '172', '238', '549', '206', '954', '444',
       '717', '210', '609', '308', '705', '306', '904', '473', '175',
       '350', '383', '454', '421', '70', '812', '442', '842', '417',
       '412', '459', '478', '335', '782', '721', '430', '429', '192',
       '200', '460', '728', '496', '816', '414', '506', '887', '613',
       '243', '569', '778', '683', '592', '319', '186', '840', '647',
       '191', '373', '437', '598', '716', '585', '982', '222', '219',
       '55', '948', '323', '691', '511', '951', '963', '25', '554', '351',
       '27', '82', '208', '913', '514', '551', '29', '103', '898', '743',
       '116', '153', '209', '353', '499', '173', '597', '809', '122',
       '411', '400', '801', '787', '237', '50', '643', '986', '97', '516',
       '837', '780', '961', '269', '20', '498', '600', '749', '642',
       '881', '72', '656', '601', '221', '228', '108', '940', '176', '33',
       '663', '34', '942', '259', '164', '458', '245', '629', '28', '288',
       '775', '785', '636', '916', '994', '309', '485', '914', '903',
       '608', '500', '54', '562', '847', '957', '688', '811', '270', '48',
       '329', '523', '921', '874', '981', '784', '280', '24', '518',
       '754', '892', '154', '860', '364', '387', '626', '161', '879',
       '39', '970', '170', '141', '160', '144', '143', '190', '376',
       '193', '246', '73', '658', '992', '253', '420', '404', '470',
       '226', '240', '89', '234', '257', '861', '467', '157', '44', '676',
       '67', '552', '885', '1020', '582', '619'], dtype=object)
```

In [37]:

```
# need to replace 'Varies with device' as the data type is string so replace with nan
df_copy["Size"]=df_copy["Size"].str.replace("Varies with device", str(np.nan))
```

In [38]:

```
df_copy["Size"].unique()
```

Out[38]:

```
array(['19000', '14000', '8.7000', '25000', '2.8000', '5.6000', '29000',
       '33000', '3.1000', '28000', '12000', '20000', '21000', '37000',
       '2.7000', '5.5000', '17000', '39000', '31000', '4.2000', '7.0000',
       '23000', '6.0000', '6.1000', '4.6000', '9.2000', '5.2000', '11000',
       '24000', 'nan', '9.4000', '15000', '10000', '1.2000', '26000',
       '8.0000', '7.9000', '56000', '57000', '35000', '54000', '201',
       '3.6000', '5.7000', '8.6000', '2.4000', '27000', '2.5000', '16000',
       '3.4000', '8.9000', '3.9000', '2.9000', '38000', '32000', '5.4000',
       '18000', '1.1000', '2.2000', '4.5000', '9.8000', '52000', '9.0000',
       '6.7000', '30000', '2.6000', '7.1000', '3.7000', '22000', '7.4000',
       '6.4000', '3.2000', '8.2000', '9.9000', '4.9000', '9.5000',
       '5.0000', '5.9000', '13000', '73000', '6.8000', '3.5000', '4.0000',
       '2.3000', '7.2000', '2.1000', '42000', '7.3000', '9.1000', '55000',
       '23', '6.5000', '1.5000', '7.5000', '51000', '41000', '48000',
       '8.5000', '46000', '8.3000', '4.3000', '4.7000', '3.3000', '40000',
       '7.8000', '8.8000', '6.6000', '5.1000', '61000', '66000', '79',
       '8.4000', '118', '44000', '695', '1.6000', '6.2000', '18', '53000',
       '1.4000', '3.0000', '5.8000', '3.8000', '9.6000', '45000', '63000',
       '49000', '77000', '4.4000', '4.8000', '70000', '6.9000', '9.3000',
       '10.0000', '8.1000', '36000', '84000', '97000', '2.0000', '1.9000',
       '1.8000', '5.3000', '47000', '556', '526', '76000', '7.6000',
       '59000', '9.7000', '78000', '72000', '43000', '7.7000', '6.3000',
       '334', '34000', '93000', '65000', '79000', '100000', '58000',
       '50000', '68000', '64000', '67000', '60000', '94000', '232',
       '99000', '624', '95000', '8.5', '41', '292', '11', '80000',
       '1.7000', '74000', '62000', '69000', '75000', '98000', '85000',
       '82000', '96000', '87000', '71000', '86000', '91000', '81000',
       '92000', '83000', '88000', '704', '862', '899', '378', '266',
       '375', '1.3000', '975', '980', '4.1000', '89000', '696', '544',
       '525', '920', '779', '853', '720', '713', '772', '318', '58',
       '241', '196', '857', '51', '953', '865', '251', '930', '540',
       '313', '746', '203', '26', '314', '239', '371', '220', '730',
       '756', '91', '293', '17', '74', '14', '317', '78', '924', '902',
       '818', '81', '939', '169', '45', '475', '965', '90000', '545',
       '61', '283', '655', '714', '93', '872', '121', '322', '1.0000',
       '976', '172', '238', '549', '206', '954', '444', '717', '210',
       '609', '308', '705', '306', '904', '473', '175', '350', '383',
       '454', '421', '70', '812', '442', '842', '417', '412', '459',
       '478', '335', '782', '721', '430', '429', '192', '200', '460',
       '728', '496', '816', '414', '506', '887', '613', '243', '569',
       '778', '683', '592', '319', '186', '840', '647', '191', '373',
       '437', '598', '716', '585', '982', '222', '219', '55', '948',
       '323', '691', '511', '951', '963', '25', '554', '351', '27', '82',
       '208', '913', '514', '551', '29', '103', '898', '743', '116',
       '153', '209', '353', '499', '173', '597', '809', '122', '411',
       '400', '801', '787', '237', '50', '643', '986', '97', '516', '837',
       '780', '961', '269', '20', '498', '600', '749', '642', '881', '72',
       '656', '601', '221', '228', '108', '940', '176', '33', '663', '34',
       '942', '259', '164', '458', '245', '629', '28', '288', '775',
       '785', '636', '916', '994', '309', '485', '914', '903', '608',
       '500', '54', '562', '847', '957', '688', '811', '270', '48', '329',
       '523', '921', '874', '981', '784', '280', '24', '518', '754',
       '892', '154', '860', '364', '387', '626', '161', '879', '39',
       '970', '170', '141', '160', '144', '143', '190', '376', '193',
       '246', '73', '658', '992', '253', '420', '404', '470', '226',
       '240', '89', '234', '257', '861', '467', '157', '44', '676', '67',
       '552', '885', '1020', '582', '619'], dtype=object)
```

or we can drope this row

In [39]:

```
df_copy['Size']= df_copy['Size'].astype("float")
```

In [40]:

```
df_copy['Size'].dtype
```

Out[40]:

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

In [41]:

```
df_copy['Size'].isnull().sum()
```

Out[41]:

```
1695
```

there 1695 null values in 'Size' column

In [42]:

```
df_copy['Size'].head()
```

Out[42]:

```
0    19000.0
1    14000.0
2        8.7
3    25000.0
4        2.8
Name: Size, dtype: float64
```

In [43]:

```
# check 3rd value its 8.7
df_copy['Size'][2]
```

Out[43]:

```
8.7
```

In [44]:

```
# roun this value by multiply by 100
df_copy['Size'][2]*1000

```

Out[44]:

```
8700.0
```

In [45]:

```
# do iteration to change all values
for i in df_copy['Size']:
    if i<11:
        df_copy['Size']=df_copy['Size'].replace(i,i*1000)

```

In [46]:

```
df_copy['Size'].head()
```

Out[46]:

```
0    19000.0
1    14000.0
2     8700.0
3    25000.0
4     2800.0
Name: Size, dtype: float64
```

In [47]:

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

Out[47]:

```
array([1.90e+04, 1.40e+04, 8.70e+03, 2.50e+04, 2.80e+03, 5.60e+03,
       2.90e+04, 3.30e+04, 3.10e+03, 2.80e+04, 1.20e+04, 2.00e+04,
       2.10e+04, 3.70e+04, 2.70e+03, 5.50e+03, 1.70e+04, 3.90e+04,
       3.10e+04, 4.20e+03, 7.00e+03, 2.30e+04, 6.00e+03, 6.10e+03,
       4.60e+03, 9.20e+03, 5.20e+03, 1.10e+04, 2.40e+04,      nan,
       9.40e+03, 1.50e+04, 1.00e+04, 1.20e+03, 2.60e+04, 8.00e+03,
       7.90e+03, 5.60e+04, 5.70e+04, 3.50e+04, 5.40e+04, 2.01e+02,
       3.60e+03, 5.70e+03, 8.60e+03, 2.40e+03, 2.70e+04, 2.50e+03,
       1.60e+04, 3.40e+03, 8.90e+03, 3.90e+03, 2.90e+03, 3.80e+04,
       3.20e+04, 5.40e+03, 1.80e+04, 1.10e+03, 2.20e+03, 4.50e+03,
       9.80e+03, 5.20e+04, 9.00e+03, 6.70e+03, 3.00e+04, 2.60e+03,
       7.10e+03, 3.70e+03, 2.20e+04, 7.40e+03, 6.40e+03, 3.20e+03,
       8.20e+03, 9.90e+03, 4.90e+03, 9.50e+03, 5.00e+03, 5.90e+03,
       1.30e+04, 7.30e+04, 6.80e+03, 3.50e+03, 4.00e+03, 2.30e+03,
       7.20e+03, 2.10e+03, 4.20e+04, 7.30e+03, 9.10e+03, 5.50e+04,
       2.30e+01, 6.50e+03, 1.50e+03, 7.50e+03, 5.10e+04, 4.10e+04,
       4.80e+04, 8.50e+03, 4.60e+04, 8.30e+03, 4.30e+03, 4.70e+03,
       3.30e+03, 4.00e+04, 7.80e+03, 8.80e+03, 6.60e+03, 5.10e+03,
       6.10e+04, 6.60e+04, 7.90e+01, 8.40e+03, 1.18e+02, 4.40e+04,
       6.95e+02, 1.60e+03, 6.20e+03, 1.80e+01, 5.30e+04, 1.40e+03,
       3.00e+03, 5.80e+03, 3.80e+03, 9.60e+03, 4.50e+04, 6.30e+04,
       4.90e+04, 7.70e+04, 4.40e+03, 4.80e+03, 7.00e+04, 6.90e+03,
       9.30e+03, 8.10e+03, 3.60e+04, 8.40e+04, 9.70e+04, 2.00e+03,
       1.90e+03, 1.80e+03, 5.30e+03, 4.70e+04, 5.56e+02, 5.26e+02,
       7.60e+04, 7.60e+03, 5.90e+04, 9.70e+03, 7.80e+04, 7.20e+04,
       4.30e+04, 7.70e+03, 6.30e+03, 3.34e+02, 3.40e+04, 9.30e+04,
       6.50e+04, 7.90e+04, 1.00e+05, 5.80e+04, 5.00e+04, 6.80e+04,
       6.40e+04, 6.70e+04, 6.00e+04, 9.40e+04, 2.32e+02, 9.90e+04,
       6.24e+02, 9.50e+04, 4.10e+01, 2.92e+02, 1.10e+01, 8.00e+04,
       1.70e+03, 7.40e+04, 6.20e+04, 6.90e+04, 7.50e+04, 9.80e+04,
       8.50e+04, 8.20e+04, 9.60e+04, 8.70e+04, 7.10e+04, 8.60e+04,
       9.10e+04, 8.10e+04, 9.20e+04, 8.30e+04, 8.80e+04, 7.04e+02,
       8.62e+02, 8.99e+02, 3.78e+02, 2.66e+02, 3.75e+02, 1.30e+03,
       9.75e+02, 9.80e+02, 4.10e+03, 8.90e+04, 6.96e+02, 5.44e+02,
       5.25e+02, 9.20e+02, 7.79e+02, 8.53e+02, 7.20e+02, 7.13e+02,
       7.72e+02, 3.18e+02, 5.80e+01, 2.41e+02, 1.96e+02, 8.57e+02,
       5.10e+01, 9.53e+02, 8.65e+02, 2.51e+02, 9.30e+02, 5.40e+02,
       3.13e+02, 7.46e+02, 2.03e+02, 2.60e+01, 3.14e+02, 2.39e+02,
       3.71e+02, 2.20e+02, 7.30e+02, 7.56e+02, 9.10e+01, 2.93e+02,
       1.70e+01, 7.40e+01, 1.40e+01, 3.17e+02, 7.80e+01, 9.24e+02,
       9.02e+02, 8.18e+02, 8.10e+01, 9.39e+02, 1.69e+02, 4.50e+01,
       4.75e+02, 9.65e+02, 9.00e+04, 5.45e+02, 6.10e+01, 2.83e+02,
       6.55e+02, 7.14e+02, 9.30e+01, 8.72e+02, 1.21e+02, 3.22e+02,
       1.00e+03, 9.76e+02, 1.72e+02, 2.38e+02, 5.49e+02, 2.06e+02,
       9.54e+02, 4.44e+02, 7.17e+02, 2.10e+02, 6.09e+02, 3.08e+02,
       7.05e+02, 3.06e+02, 9.04e+02, 4.73e+02, 1.75e+02, 3.50e+02,
       3.83e+02, 4.54e+02, 4.21e+02, 7.00e+01, 8.12e+02, 4.42e+02,
       8.42e+02, 4.17e+02, 4.12e+02, 4.59e+02, 4.78e+02, 3.35e+02,
       7.82e+02, 7.21e+02, 4.30e+02, 4.29e+02, 1.92e+02, 2.00e+02,
       4.60e+02, 7.28e+02, 4.96e+02, 8.16e+02, 4.14e+02, 5.06e+02,
       8.87e+02, 6.13e+02, 2.43e+02, 5.69e+02, 7.78e+02, 6.83e+02,
       5.92e+02, 3.19e+02, 1.86e+02, 8.40e+02, 6.47e+02, 1.91e+02,
       3.73e+02, 4.37e+02, 5.98e+02, 7.16e+02, 5.85e+02, 9.82e+02,
       2.22e+02, 2.19e+02, 5.50e+01, 9.48e+02, 3.23e+02, 6.91e+02,
       5.11e+02, 9.51e+02, 9.63e+02, 2.50e+01, 5.54e+02, 3.51e+02,
       2.70e+01, 8.20e+01, 2.08e+02, 9.13e+02, 5.14e+02, 5.51e+02,
       2.90e+01, 1.03e+02, 8.98e+02, 7.43e+02, 1.16e+02, 1.53e+02,
       2.09e+02, 3.53e+02, 4.99e+02, 1.73e+02, 5.97e+02, 8.09e+02,
       1.22e+02, 4.11e+02, 4.00e+02, 8.01e+02, 7.87e+02, 2.37e+02,
       5.00e+01, 6.43e+02, 9.86e+02, 9.70e+01, 5.16e+02, 8.37e+02,
       7.80e+02, 9.61e+02, 2.69e+02, 2.00e+01, 4.98e+02, 6.00e+02,
       7.49e+02, 6.42e+02, 8.81e+02, 7.20e+01, 6.56e+02, 6.01e+02,
       2.21e+02, 2.28e+02, 1.08e+02, 9.40e+02, 1.76e+02, 3.30e+01,
       6.63e+02, 3.40e+01, 9.42e+02, 2.59e+02, 1.64e+02, 4.58e+02,
       2.45e+02, 6.29e+02, 2.80e+01, 2.88e+02, 7.75e+02, 7.85e+02,
       6.36e+02, 9.16e+02, 9.94e+02, 3.09e+02, 4.85e+02, 9.14e+02,
       9.03e+02, 6.08e+02, 5.00e+02, 5.40e+01, 5.62e+02, 8.47e+02,
       9.57e+02, 6.88e+02, 8.11e+02, 2.70e+02, 4.80e+01, 3.29e+02,
       5.23e+02, 9.21e+02, 8.74e+02, 9.81e+02, 7.84e+02, 2.80e+02,
       2.40e+01, 5.18e+02, 7.54e+02, 8.92e+02, 1.54e+02, 8.60e+02,
       3.64e+02, 3.87e+02, 6.26e+02, 1.61e+02, 8.79e+02, 3.90e+01,
       9.70e+02, 1.70e+02, 1.41e+02, 1.60e+02, 1.44e+02, 1.43e+02,
       1.90e+02, 3.76e+02, 1.93e+02, 2.46e+02, 7.30e+01, 6.58e+02,
       9.92e+02, 2.53e+02, 4.20e+02, 4.04e+02, 4.70e+02, 2.26e+02,
       2.40e+02, 8.90e+01, 2.34e+02, 2.57e+02, 8.61e+02, 4.67e+02,
       1.57e+02, 4.40e+01, 6.76e+02, 6.70e+01, 5.52e+02, 8.85e+02,
       1.02e+03, 5.82e+02, 6.19e+02])
```

In [48]:

```
df_copy['Size'].dtype
```

Out[48]:

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

In [49]:

```
df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10840 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10840 non-null  object 
 1   Category        10840 non-null  object 
 2   Rating          9366 non-null   float64
 3   Reviews         10840 non-null  int32  
 4   Size            9145 non-null   float64
 5   Installs        10840 non-null  object 
 6   Type            10839 non-null  object 
 7   Price           10840 non-null  object 
 8   Content Rating  10840 non-null  object 
 9   Genres          10840 non-null  object 
 10  Last Updated    10840 non-null  object 
 11  Current Ver     10832 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(2), int32(1), object(10)
memory usage: 1.4+ MB
```

In [50]:

```
# scale the value in paeticular range
df_copy['Size']=df_copy['Size']/1000
```

In [51]:

```
df_copy['Size']
```

Out[51]:

```
0        19.0
1        14.0
2         8.7
3        25.0
4         2.8
         ...
10836    53.0
10837     3.6
10838     9.5
10839     NaN
10840    19.0
Name: Size, Length: 10840, dtype: float64
```

In [52]:

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

Out[52]:

```
array([1.90e+01, 1.40e+01, 8.70e+00, 2.50e+01, 2.80e+00, 5.60e+00,
       2.90e+01, 3.30e+01, 3.10e+00, 2.80e+01, 1.20e+01, 2.00e+01,
       2.10e+01, 3.70e+01, 2.70e+00, 5.50e+00, 1.70e+01, 3.90e+01,
       3.10e+01, 4.20e+00, 7.00e+00, 2.30e+01, 6.00e+00, 6.10e+00,
       4.60e+00, 9.20e+00, 5.20e+00, 1.10e+01, 2.40e+01,      nan,
       9.40e+00, 1.50e+01, 1.00e+01, 1.20e+00, 2.60e+01, 8.00e+00,
       7.90e+00, 5.60e+01, 5.70e+01, 3.50e+01, 5.40e+01, 2.01e-01,
       3.60e+00, 5.70e+00, 8.60e+00, 2.40e+00, 2.70e+01, 2.50e+00,
       1.60e+01, 3.40e+00, 8.90e+00, 3.90e+00, 2.90e+00, 3.80e+01,
       3.20e+01, 5.40e+00, 1.80e+01, 1.10e+00, 2.20e+00, 4.50e+00,
       9.80e+00, 5.20e+01, 9.00e+00, 6.70e+00, 3.00e+01, 2.60e+00,
       7.10e+00, 3.70e+00, 2.20e+01, 7.40e+00, 6.40e+00, 3.20e+00,
       8.20e+00, 9.90e+00, 4.90e+00, 9.50e+00, 5.00e+00, 5.90e+00,
       1.30e+01, 7.30e+01, 6.80e+00, 3.50e+00, 4.00e+00, 2.30e+00,
       7.20e+00, 2.10e+00, 4.20e+01, 7.30e+00, 9.10e+00, 5.50e+01,
       2.30e-02, 6.50e+00, 1.50e+00, 7.50e+00, 5.10e+01, 4.10e+01,
       4.80e+01, 8.50e+00, 4.60e+01, 8.30e+00, 4.30e+00, 4.70e+00,
       3.30e+00, 4.00e+01, 7.80e+00, 8.80e+00, 6.60e+00, 5.10e+00,
       6.10e+01, 6.60e+01, 7.90e-02, 8.40e+00, 1.18e-01, 4.40e+01,
       6.95e-01, 1.60e+00, 6.20e+00, 1.80e-02, 5.30e+01, 1.40e+00,
       3.00e+00, 5.80e+00, 3.80e+00, 9.60e+00, 4.50e+01, 6.30e+01,
       4.90e+01, 7.70e+01, 4.40e+00, 4.80e+00, 7.00e+01, 6.90e+00,
       9.30e+00, 8.10e+00, 3.60e+01, 8.40e+01, 9.70e+01, 2.00e+00,
       1.90e+00, 1.80e+00, 5.30e+00, 4.70e+01, 5.56e-01, 5.26e-01,
       7.60e+01, 7.60e+00, 5.90e+01, 9.70e+00, 7.80e+01, 7.20e+01,
       4.30e+01, 7.70e+00, 6.30e+00, 3.34e-01, 3.40e+01, 9.30e+01,
       6.50e+01, 7.90e+01, 1.00e+02, 5.80e+01, 5.00e+01, 6.80e+01,
       6.40e+01, 6.70e+01, 6.00e+01, 9.40e+01, 2.32e-01, 9.90e+01,
       6.24e-01, 9.50e+01, 4.10e-02, 2.92e-01, 1.10e-02, 8.00e+01,
       1.70e+00, 7.40e+01, 6.20e+01, 6.90e+01, 7.50e+01, 9.80e+01,
       8.50e+01, 8.20e+01, 9.60e+01, 8.70e+01, 7.10e+01, 8.60e+01,
       9.10e+01, 8.10e+01, 9.20e+01, 8.30e+01, 8.80e+01, 7.04e-01,
       8.62e-01, 8.99e-01, 3.78e-01, 2.66e-01, 3.75e-01, 1.30e+00,
       9.75e-01, 9.80e-01, 4.10e+00, 8.90e+01, 6.96e-01, 5.44e-01,
       5.25e-01, 9.20e-01, 7.79e-01, 8.53e-01, 7.20e-01, 7.13e-01,
       7.72e-01, 3.18e-01, 5.80e-02, 2.41e-01, 1.96e-01, 8.57e-01,
       5.10e-02, 9.53e-01, 8.65e-01, 2.51e-01, 9.30e-01, 5.40e-01,
       3.13e-01, 7.46e-01, 2.03e-01, 2.60e-02, 3.14e-01, 2.39e-01,
       3.71e-01, 2.20e-01, 7.30e-01, 7.56e-01, 9.10e-02, 2.93e-01,
       1.70e-02, 7.40e-02, 1.40e-02, 3.17e-01, 7.80e-02, 9.24e-01,
       9.02e-01, 8.18e-01, 8.10e-02, 9.39e-01, 1.69e-01, 4.50e-02,
       4.75e-01, 9.65e-01, 9.00e+01, 5.45e-01, 6.10e-02, 2.83e-01,
       6.55e-01, 7.14e-01, 9.30e-02, 8.72e-01, 1.21e-01, 3.22e-01,
       1.00e+00, 9.76e-01, 1.72e-01, 2.38e-01, 5.49e-01, 2.06e-01,
       9.54e-01, 4.44e-01, 7.17e-01, 2.10e-01, 6.09e-01, 3.08e-01,
       7.05e-01, 3.06e-01, 9.04e-01, 4.73e-01, 1.75e-01, 3.50e-01,
       3.83e-01, 4.54e-01, 4.21e-01, 7.00e-02, 8.12e-01, 4.42e-01,
       8.42e-01, 4.17e-01, 4.12e-01, 4.59e-01, 4.78e-01, 3.35e-01,
       7.82e-01, 7.21e-01, 4.30e-01, 4.29e-01, 1.92e-01, 2.00e-01,
       4.60e-01, 7.28e-01, 4.96e-01, 8.16e-01, 4.14e-01, 5.06e-01,
       8.87e-01, 6.13e-01, 2.43e-01, 5.69e-01, 7.78e-01, 6.83e-01,
       5.92e-01, 3.19e-01, 1.86e-01, 8.40e-01, 6.47e-01, 1.91e-01,
       3.73e-01, 4.37e-01, 5.98e-01, 7.16e-01, 5.85e-01, 9.82e-01,
       2.22e-01, 2.19e-01, 5.50e-02, 9.48e-01, 3.23e-01, 6.91e-01,
       5.11e-01, 9.51e-01, 9.63e-01, 2.50e-02, 5.54e-01, 3.51e-01,
       2.70e-02, 8.20e-02, 2.08e-01, 9.13e-01, 5.14e-01, 5.51e-01,
       2.90e-02, 1.03e-01, 8.98e-01, 7.43e-01, 1.16e-01, 1.53e-01,
       2.09e-01, 3.53e-01, 4.99e-01, 1.73e-01, 5.97e-01, 8.09e-01,
       1.22e-01, 4.11e-01, 4.00e-01, 8.01e-01, 7.87e-01, 2.37e-01,
       5.00e-02, 6.43e-01, 9.86e-01, 9.70e-02, 5.16e-01, 8.37e-01,
       7.80e-01, 9.61e-01, 2.69e-01, 2.00e-02, 4.98e-01, 6.00e-01,
       7.49e-01, 6.42e-01, 8.81e-01, 7.20e-02, 6.56e-01, 6.01e-01,
       2.21e-01, 2.28e-01, 1.08e-01, 9.40e-01, 1.76e-01, 3.30e-02,
       6.63e-01, 3.40e-02, 9.42e-01, 2.59e-01, 1.64e-01, 4.58e-01,
       2.45e-01, 6.29e-01, 2.80e-02, 2.88e-01, 7.75e-01, 7.85e-01,
       6.36e-01, 9.16e-01, 9.94e-01, 3.09e-01, 4.85e-01, 9.14e-01,
       9.03e-01, 6.08e-01, 5.00e-01, 5.40e-02, 5.62e-01, 8.47e-01,
       9.57e-01, 6.88e-01, 8.11e-01, 2.70e-01, 4.80e-02, 3.29e-01,
       5.23e-01, 9.21e-01, 8.74e-01, 9.81e-01, 7.84e-01, 2.80e-01,
       2.40e-02, 5.18e-01, 7.54e-01, 8.92e-01, 1.54e-01, 8.60e-01,
       3.64e-01, 3.87e-01, 6.26e-01, 1.61e-01, 8.79e-01, 3.90e-02,
       9.70e-01, 1.70e-01, 1.41e-01, 1.60e-01, 1.44e-01, 1.43e-01,
       1.90e-01, 3.76e-01, 1.93e-01, 2.46e-01, 7.30e-02, 6.58e-01,
       9.92e-01, 2.53e-01, 4.20e-01, 4.04e-01, 4.70e-01, 2.26e-01,
       2.40e-01, 8.90e-02, 2.34e-01, 2.57e-01, 8.61e-01, 4.67e-01,
       1.57e-01, 4.40e-02, 6.76e-01, 6.70e-02, 5.52e-01, 8.85e-01,
       1.02e+00, 5.82e-01, 6.19e-01])
```

In [53]:

```
df_copy.columns
```

Out[53]:

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

In [54]:

```
df_copy.head()
```

Out[54]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10,000+	Free	0	Everyone	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500,000+	Free	0	Everyone	
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5,000,000+	Free	0	Everyone	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50,000,000+	Free	0	Teen	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100,000+	Free	0	Everyone	D

focus on 'Installs' and 'Price'

In [55]:

```
df_copy['Installs'].dtype
```

Out[55]:

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

In [56]:

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

Out[56]:

```
array(['10,000+', '500,000+', '5,000,000+', '50,000,000+', '100,000+',
       '50,000+', '1,000,000+', '10,000,000+', '5,000+', '100,000,000+',
       '1,000,000,000+', '1,000+', '500,000,000+', '50+', '100+', '500+',
       '10+', '1+', '5+', '0+', '0'], dtype=object)
```

In [57]:

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

Out[57]:

```
array(['0', '$4.99', '$3.99', '$6.99', '$1.49', '$2.99', '$7.99', '$5.99',
       '$3.49', '$1.99', '$9.99', '$7.49', '$0.99', '$9.00', '$5.49',
       '$10.00', '$24.99', '$11.99', '$79.99', '$16.99', '$14.99',
       '$1.00', '$29.99', '$12.99', '$2.49', '$10.99', '$1.50', '$19.99',
       '$15.99', '$33.99', '$74.99', '$39.99', '$3.95', '$4.49', '$1.70',
       '$8.99', '$2.00', '$3.88', '$25.99', '$399.99', '$17.99',
       '$400.00', '$3.02', '$1.76', '$4.84', '$4.77', '$1.61', '$2.50',
       '$1.59', '$6.49', '$1.29', '$5.00', '$13.99', '$299.99', '$379.99',
       '$37.99', '$18.99', '$389.99', '$19.90', '$8.49', '$1.75',
       '$14.00', '$4.85', '$46.99', '$109.99', '$154.99', '$3.08',
       '$2.59', '$4.80', '$1.96', '$19.40', '$3.90', '$4.59', '$15.46',
       '$3.04', '$4.29', '$2.60', '$3.28', '$4.60', '$28.99', '$2.95',
       '$2.90', '$1.97', '$200.00', '$89.99', '$2.56', '$30.99', '$3.61',
       '$394.99', '$1.26', '$1.20', '$1.04'], dtype=object)
```

focus on above two columns we need to remove +, "," ,$ symbol

In [62]:

```
charater_remove=['+', ",", "$",]
columns_clean= ["Installs","Price"]
for i in charater_remove:   # which charates need to be remove
    for col in columns_clean:    # columns to be clean
        df_copy[col]=df_copy[col].str.replace(i, '')
```

In [63]:

```
df_copy["Installs"].unique()
```

Out[63]:

```
array(['10000', '500000', '5000000', '50000000', '100000', '50000',
       '1000000', '10000000', '5000', '100000000', '1000000000', '1000',
       '500000000', '50', '100', '500', '10', '1', '5', '0'], dtype=object)
```

In [64]:

```
df_copy["Price"].unique()
```

Out[64]:

```
array(['0', '4.99', '3.99', '6.99', '1.49', '2.99', '7.99', '5.99',
       '3.49', '1.99', '9.99', '7.49', '0.99', '9.00', '5.49', '10.00',
       '24.99', '11.99', '79.99', '16.99', '14.99', '1.00', '29.99',
       '12.99', '2.49', '10.99', '1.50', '19.99', '15.99', '33.99',
       '74.99', '39.99', '3.95', '4.49', '1.70', '8.99', '2.00', '3.88',
       '25.99', '399.99', '17.99', '400.00', '3.02', '1.76', '4.84',
       '4.77', '1.61', '2.50', '1.59', '6.49', '1.29', '5.00', '13.99',
       '299.99', '379.99', '37.99', '18.99', '389.99', '19.90', '8.49',
       '1.75', '14.00', '4.85', '46.99', '109.99', '154.99', '3.08',
       '2.59', '4.80', '1.96', '19.40', '3.90', '4.59', '15.46', '3.04',
       '4.29', '2.60', '3.28', '4.60', '28.99', '2.95', '2.90', '1.97',
       '200.00', '89.99', '2.56', '30.99', '3.61', '394.99', '1.26',
       '1.20', '1.04'], dtype=object)
```

In [65]:

```
df_copy["Price"]
```

Out[65]:

```
0        0
1        0
2        0
3        0
4        0
        ..
10836    0
10837    0
10838    0
10839    0
10840    0
Name: Price, Length: 10840, dtype: object
```

In [67]:

```
df_copy["Installs"]=df_copy["Installs"].astype('int')
```

In [68]:

```
df_copy["Price"]=df_copy["Price"].astype('float')
```

In [70]:

```
df_copy["Price"].dtype
```

Out[70]:

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

In [71]:

```
df_copy["Installs"].dtype
```

Out[71]:

```
dtype('int32')
```

In [73]:

```
df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10840 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10840 non-null  object 
 1   Category        10840 non-null  object 
 2   Rating          9366 non-null   float64
 3   Reviews         10840 non-null  int32  
 4   Size            9145 non-null   float64
 5   Installs        10840 non-null  int32  
 6   Type            10839 non-null  object 
 7   Price           10840 non-null  float64
 8   Content Rating  10840 non-null  object 
 9   Genres          10840 non-null  object 
 10  Last Updated    10840 non-null  object 
 11  Current Ver     10832 non-null  object 
 12  Android Ver     10838 non-null  object 
dtypes: float64(3), int32(2), object(8)
memory usage: 1.3+ MB
```

Work on 'Last Updated' column which is object

In [74]:

```
df_copy["Last Updated"]
```

Out[74]:

```
0           January 7, 2018
1          January 15, 2018
2             August 1, 2018
3               June 8, 2018
4              June 20, 2018
                ...        
10836          July 25, 2017
10837           July 6, 2018
10838      January 20, 2017
10839      January 19, 2015
10840          July 25, 2018
Name: Last Updated, Length: 10840, dtype: object
```

In [75]:

```
df_copy["Last Updated"].unique()
```

Out[75]:

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

Break into date , Month, year

In [77]:

```
pd.to_datetime(df_copy["Last Updated"])

```

Out[77]:

```
0       2018-01-07
1       2018-01-15
2       2018-08-01
3       2018-06-08
4       2018-06-20
           ...
10836   2017-07-25
10837   2018-07-06
10838   2017-01-20
10839   2015-01-19
10840   2018-07-25
Name: Last Updated, Length: 10840, dtype: datetime64[ns]
```

In [78]:

```
df_copy["Last Updated"]=pd.to_datetime(df_copy["Last Updated"])

```

In [79]:

```
df_copy.head(3)
```

Out[79]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	Ar
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	Desig
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	Ar

In [88]:

```
# create new column of day
df_copy["day"]=df_copy["Last Updated"].dt.day
```

In [90]:

```

df_copy["day"]
```

Out[90]:

```
0         7
1        15
2         1
3         8
4        20
         ..
10836    25
10837     6
10838    20
10839    19
10840    25
Name: day, Length: 10840, dtype: int64
```

In [91]:

```
# create new column of month
df_copy["month"]=df_copy["Last Updated"].dt.month

```

In [92]:

```
df_copy["month"]
```

Out[92]:

```
0        1
1        1
2        8
3        6
4        6
        ..
10836    7
10837    7
10838    1
10839    1
10840    7
Name: month, Length: 10840, dtype: int64
```

In [85]:

```
# create new column of year
df_copy["yrar"]=df_copy["Last Updated"].dt.year
```

In [93]:

```
df_copy["yrar"]
```

Out[93]:

```
0        2018
1        2018
2        2018
3        2018
4        2018
         ...
10836    2017
10837    2018
10838    2017
10839    2015
10840    2018
Name: yrar, Length: 10840, dtype: int64
```

In [94]:

```
df_copy.head()
```

Out[94]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	De
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Des

In [95]:

```
df_copy["yrar"].dtype
```

Out[95]:

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

In [96]:

```
df.head()
```

Out[96]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone

In [98]:

```
# save the  clean data into new csv
df_copy.to_csv ("clean_gpaydata.csv",index= False)
```

In [99]:

```
df1=pd.read_csv('clean_gpaydata.csv')
df1.head()
```

Out[99]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	De
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Des

In [100]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10840 entries, 0 to 10839
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   App             10840 non-null  object 
 1   Category        10840 non-null  object 
 2   Rating          9366 non-null   float64
 3   Reviews         10840 non-null  int64  
 4   Size            9145 non-null   float64
 5   Installs        10840 non-null  int64  
 6   Type            10839 non-null  object 
 7   Price           10840 non-null  float64
 8   Content Rating  10840 non-null  object 
 9   Genres          10840 non-null  object 
 10  Last Updated    10840 non-null  object 
 11  Current Ver     10832 non-null  object 
 12  Android Ver     10838 non-null  object 
 13  day             10840 non-null  int64  
 14  date            10840 non-null  object 
 15  month           10840 non-null  int64  
 16  yrar            10840 non-null  int64  
dtypes: float64(3), int64(5), object(9)
memory usage: 1.4+ MB
```

End

In [ ]:

```

```