In [1]:

```python
# Importing basic libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```python
# Laoding the dataset
df = pd.read_csv('googleplaystore.csv')
```

In [3]:

```python
# Getting top 5 rows of the dataset
df.head()
```

Out[3]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

In [4]:

```python
# Getting last 5 rows of the dataset
df.tail()
```

Out[4]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device

In [5]:

```python
# Getting datatypes of the features
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10841 entries, 0 to 10840
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   App             10841 non-null  object
 1   Category        10841 non-null  object
 2   Rating          9367 non-null   float64
 3   Reviews         10841 non-null  object
 4   Size            10841 non-null  object
 5   Installs        10841 non-null  object
 6   Type            10840 non-null  object
 7   Price           10841 non-null  object
 8   Content Rating  10840 non-null  object
 9   Genres          10841 non-null  object
 10  Last Updated    10841 non-null  object
 11  Current Ver     10833 non-null  object
 12  Android Ver     10838 non-null  object
dtypes: float64(1), object(12)
memory usage: 1.1+ MB
```

In [6]:

```python
# Getting number of rows and number of columns
df.shape
```

Out[6]:

```
(10841, 13)
```

In [7]:

```python
# Checking if our dataset is having duplicates values or not
df[df.duplicated()]
```

Out[7]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
229	Quick PDF Scanner + OCR FREE	BUSINESS	4.2	80805	Varies with device	5,000,000+	Free	0	Everyone	Business	February 26, 2018	Varies with device	4.0.3 and up
236	Box	BUSINESS	4.2	159872	Varies with device	10,000,000+	Free	0	Everyone	Business	July 31, 2018	Varies with device	Varies with device
239	Google My Business	BUSINESS	4.4	70991	Varies with device	5,000,000+	Free	0	Everyone	Business	July 24, 2018	2.19.0.204537701	4.4 and up
256	ZOOM Cloud Meetings	BUSINESS	4.4	31614	37M	10,000,000+	Free	0	Everyone	Business	July 20, 2018	4.1.28165.0716	4.0 and up
261	join.me - Simple Meetings	BUSINESS	4.0	6989	Varies with device	1,000,000+	Free	0	Everyone	Business	July 16, 2018	4.3.0.508	4.4 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...
8643	Wunderlist: To-Do List & Tasks	PRODUCTIVITY	4.6	404610	Varies with device	10,000,000+	Free	0	Everyone	Productivity	April 6, 2018	Varies with device	Varies with device
8654	TickTick: To Do List with Reminder, Day Planner	PRODUCTIVITY	4.6	25370	Varies with device	1,000,000+	Free	0	Everyone	Productivity	August 6, 2018	Varies with device	Varies with device
8658	ColorNote Notepad Notes	PRODUCTIVITY	4.6	2401017	Varies with device	100,000,000+	Free	0	Everyone	Productivity	June 27, 2018	Varies with device	Varies with device
10049	Airway Ex - Intubate. Anesthetize. Train.	MEDICAL	4.3	123	86M	10,000+	Free	0	Everyone	Medical	June 1, 2018	0.6.88	5.0 and up
10768	AAFP	MEDICAL	3.8	63	24M	10,000+	Free	0	Everyone	Medical	June 22, 2018	2.3.1	5.0 and up

483 rows × 13 columns

In [8]:

```python
# Getting 10 sample of the dataset
df.sample(10)
```

Out[8]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
4578	Samsung Smart Switch Mobile	TOOLS	4.3	146913	24M	100,000,000+	Free	0	Everyone	Tools	July 18, 2018	3.5.02.15	4.0 and up
2355	MDLIVE: Talk to a Doctor 24/7	MEDICAL	3.4	650	27M	100,000+	Free	0	Everyone	Medical	July 23, 2018	4.23.0	4.4 and up
7527	CL Pebble Apps	LIFESTYLE	3.6	67	499k	5,000+	Free	0	Everyone	Lifestyle	December 23, 2015	3.5	4.0 and up
434	Full Screen Caller ID	COMMUNICATION	4.2	104990	10M	5,000,000+	Free	0	Everyone	Communication	May 15, 2018	3.5.0	4.0.3 and up
4513	Q Remote Control	TOOLS	3.8	4264	14M	500,000+	Free	0	Everyone	Tools	July 12, 2014	2.6.1	2.3 and up
5380	Vikings: an Archer's Journey	GAME	4.5	10256	39M	1,000,000+	Free	0	Everyone	Action	December 11, 2017	2.1.0	4.3 and up
2311	Pregnancy Week By Week	MEDICAL	4.8	78825	11M	1,000,000+	Free	0	Everyone	Medical	August 2, 2018	1.2.16	4.1 and up
9438	Egg Baby	FAMILY	4.4	351607	8.7M	5,000,000+	Free	0	Everyone 10+	Casual	June 29, 2016	1.26.03	2.3.3 and up
3685	video player for android	VIDEO_PLAYERS	4.5	351168	13M	10,000,000+	Free	0	Everyone	Video Players & Editors	October 18, 2016	2.0.0	2.3 and up
235	Tiny Scanner Pro: PDF Doc Scan	BUSINESS	4.8	10295	39M	100,000+	Paid	$4.99	Everyone	Business	April 11, 2017	3.4.6	3.0 and up

In [9]:

```python
# Getting basic statistical terms
df.describe(include='all')
```

Out[9]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
count	10841	10841	9367.000000	10841	10841	10841	10840	10841	10840	10841	10841	10833	10838
unique	9660	34	NaN	6002	462	22	3	93	6	120	1378	2832	33
top	ROBLOX	FAMILY	NaN	0	Varies with device	1,000,000+	Free	0	Everyone	Tools	August 3, 2018	Varies with device	4.1 and up
freq	9	1972	NaN	596	1695	1579	10039	10040	8714	842	326	1459	2451
mean	NaN	NaN	4.193338	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	0.537431	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN	1.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	NaN	NaN	4.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	NaN	NaN	4.300000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	NaN	NaN	4.500000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	NaN	NaN	19.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [10]:

```python
df['Reviews'].head()
```

Out[10]:

```
0       159
1       967
2     87510
3    215644
4       967
Name: Reviews, dtype: object
```

In [11]:

```python
df['Reviews'].shape
```

Out[11]:

```
(10841,)
```

In [12]:

```python
df['Reviews'].dtype
```

Out[12]:

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

In [13]:

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

Out[13]:

```
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 [14]:

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

Out[14]:

```
10840
```

In [15]:

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

Out[15]:

```
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 [16]:

```python
# (~) denotes the compliment
```

In [17]:

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

Out[17]:

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

In [18]:

```python
# copying the original dataset
df_copy = df.copy()
```

In [19]:

```python
# Dropping the 10472 index of the review columns
df_copy = df_copy.drop(df_copy.index[10472])
```

In [20]:

```python
df_copy.shape
```

Out[20]:

```
(10840, 13)
```

In [21]:

```python
df_copy['Reviews'].dtype
```

Out[21]:

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

In [22]:

```python
# Changing the datatype into int

df_copy['Reviews'] = df_copy['Reviews'].astype('int')
```

In [23]:

```python
## it is good practice to copy our data
```

In [24]:

```python
df_copy['Reviews'].dtype
```

Out[24]:

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

In [25]:

```python
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
```

In [26]:

```python
# Cleaning the size feature
df_copy['Size'].unique()
```

Out[26]:

```
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)
```

In [27]:

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

In [28]:

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

Out[28]:

```
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 [29]:

```python
# Replacing 'k' with ''(space)
df_copy['Size'] = df_copy.Size.str.replace('k','')
```

In [30]:

```python
df_copy.Size
```

Out[30]:

```
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 [31]:

```python
df_copy.Size.unique()
```

Out[31]:

```
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 [32]:

```python
# Replacing 'Varies with device' with NAN value
df_copy['Size'] = df_copy['Size'].str.replace('Varies with device',str(np.nan))
```

In [33]:

```python
np.nan
```

Out[33]:

```
nan
```

In [34]:

```python
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', '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)
```

In [35]:

```python
# Changing the datatype of the feature into float

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

In [36]:

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

Out[36]:

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

In [37]:

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

Out[37]:

```
1695
```

In [38]:

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

Out[38]:

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

In [39]:

```python
df_copy['Size'][2]
```

Out[39]:

```
8.7
```

In [40]:

```python
df_copy['Size'].unique
```

Out[40]:

```
<bound method Series.unique of 0        19000.0
1        14000.0
2            8.7
3        25000.0
4            2.8
          ...
10836    53000.0
10837        3.6
10838        9.5
10839        NaN
10840    19000.0
Name: Size, Length: 10840, dtype: float64>
```

We Can see we still have decimal values in our feature

In [41]:

```python
# Changing this decimal values
for i in df_copy['Size']:
    if i<10:
        df_copy['Size'] = df_copy['Size'].replace(i,i*1000)
```

In [42]:

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

Out[42]:

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

In [43]:

```python
df_copy['Size'].unique
```

Out[43]:

```
<bound method Series.unique of 0        19000.0
1        14000.0
2         8700.0
3        25000.0
4         2800.0
          ...
10836    53000.0
10837     3600.0
10838     9500.0
10839        NaN
10840    19000.0
Name: Size, Length: 10840, dtype: float64>
```

In [44]:

```python
df_copy['Size'] = df_copy['Size']/1000
```

In [45]:

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

Out[45]:

```
0    19.0
1    14.0
2     8.7
3    25.0
4     2.8
Name: Size, dtype: float64
```

In [46]:

```python
df.columns
```

Out[46]:

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

In [47]:

```python
df_copy.head(2)
```

Out[47]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

In [48]:

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

Out[48]:

```
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 [49]:

```python
df_copy['Installs']
```

Out[49]:

```
0            10,000+
1           500,000+
2         5,000,000+
3        50,000,000+
4           100,000+
             ...
10836         5,000+
10837           100+
10838         1,000+
10839         1,000+
10840    10,000,000+
Name: Installs, Length: 10840, dtype: object
```

We can see there is '$' and '+' values in Price and Installs feature. So We have to replace it with ' '.

In [50]:

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

Out[50]:

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

In [51]:

```python
chars_to_remove = ['+',',','$']
cols_to_clean = ['Installs','Price']
for item in chars_to_remove:
    for col in cols_to_clean:
        df_copy[col] = df_copy[col].str.replace(item,'')
```

```
C:\Users\rosha\AppData\Local\Temp/ipykernel_16268/3221494010.py:5: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.
  df_copy[col] = df_copy[col].str.replace(item,'')
```

In [52]:

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

Out[52]:

```
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 [53]:

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

Out[53]:

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

In [54]:

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

In [55]:

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

In [56]:

```python
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
```

In [57]:

```python
df_copy['Last Updated']
```

Out[57]:

```
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 [58]:

```python
df_copy['Last Updated'].dtype
```

Out[58]:

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

In [60]:

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

In [62]:

```python
df_copy['Day'] = df_copy['Last Updated'].dt.day
```

In [63]:

```python
df_copy['Month'] = df_copy['Last Updated'].dt.month
```

In [64]:

```python
df_copy['Day']
```

Out[64]:

```
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 [65]:

```python
df_copy['Month']
```

Out[65]:

```
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 [67]:

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

Out[67]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

In [68]:

```python
df_copy.head(2)
```

Out[68]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	Day	Month
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	Art & Design	2018-01-07	1.0.0	4.0.3 and up	7	1
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	Art & Design;Pretend Play	2018-01-15	2.0.0	4.0.3 and up	15	1

In [69]:

```python
df_copy.to_csv('Google_cleaned.csv',index=False)
```

In [70]:

```python
pd.read_csv('Google_cleaned.csv')
```

Out[70]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	Day	Month
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10000	Free	0.0	Everyone	Art & Design	2018-01-07	1.0.0	4.0.3 and up	7	1
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500000	Free	0.0	Everyone	Art & Design;Pretend Play	2018-01-15	2.0.0	4.0.3 and up	15	1
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7	5000000	Free	0.0	Everyone	Art & Design	2018-08-01	1.2.4	4.0.3 and up	1	8
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25.0	50000000	Free	0.0	Teen	Art & Design	2018-06-08	Varies with device	4.2 and up	8	6
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8	100000	Free	0.0	Everyone	Art & Design;Creativity	2018-06-20	1.1	4.4 and up	20	6
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10835	Sya9a Maroc - FR	FAMILY	4.5	38	53.0	5000	Free	0.0	Everyone	Education	2017-07-25	1.48	4.1 and up	25	7
10836	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6	100	Free	0.0	Everyone	Education	2018-07-06	1.0	4.1 and up	6	7
10837	Parkinson Exercices FR	MEDICAL	NaN	3	9.5	1000	Free	0.0	Everyone	Medical	2017-01-20	1.0	2.2 and up	20	1
10838	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	NaN	1000	Free	0.0	Mature 17+	Books & Reference	2015-01-19	Varies with device	Varies with device	19	1
10839	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19.0	10000000	Free	0.0	Everyone	Lifestyle	2018-07-25	Varies with device	Varies with device	25	7

10840 rows × 15 columns

In [ ]: