In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import missingno as msno
```

In [2]:

```
df = pd.read_csv('googleplaystore.csv')
df.sample(5)
```

Out[2]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
6027	BD Online Passport Application	FAMILY	4.6	12	2.4M	5,000+	Free	0	Everyone	Education	7-Oct-17	3	4.0.3 and up
8714	Dairy Queen	FOOD_AND_DRINK	3.6	742	43M	100,000+	Free	0	Everyone	Food & Drink	25-Jul-18	2.1.0	4.1 and up
4830	Z War-Zombie Modern Combat	FAMILY	4.2	62301	72M	5,000,000+	Free	0	Teen	Strategy	1-Aug-18	1.6	4.1 and up
5577	AS Guía de las Ligas 2017-2018	SPORTS	4.1	4374	53M	100,000+	Free	0	Everyone	Sports	14-Sep-17	1.0.10	4.1 and up
2892	Cameringo Lite. Filters Camera	PHOTOGRAPHY	4.2	140917	5.7M	10,000,000+	Free	0	Everyone	Photography	11-Jun-18	2.2.93	4.0 and up

Data Cleaning

1. Which of the following column(s) has/have null values?

In [3]:

```
df.isna().sum().sort_values(ascending = False)
```

Out[3]:

```
Rating            1474
Current Ver          8
Android Ver          3
Type                 1
Content Rating       1
App                  0
Category             0
Reviews              0
Size                 0
Installs             0
Price                0
Genres               0
Last Updated         0
dtype: int64
```

2. Clean the `Rating` column and the other columns containing null values

In [4]:

```
df.loc[df['Rating'] > 5 , 'Rating'] = np.nan
```

In [5]:

```
df.loc[df['Rating'] > 5]
```

Out[5]:

App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver

In [6]:

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

Out[6]:

```
4.197726785331332
```

In [7]:

```
df['Rating'].fillna(df['Rating'].mean() , inplace = True)
```

In [8]:

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

3. Clean the column `Reviews` and make it numeric

In [9]:

```
df['new rev'] = pd.to_numeric(df['Reviews'] , errors = 'coerce')

#null values will be passed where error is encountered.
```

In [10]:

```
df.loc[df['new rev'].isna()]
```

Out[10]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	new rev
72	Android Auto - Maps, Media, Messaging & Voice	AUTO_AND_VEHICLES	4.2	2M	16M	10,000,000+	Free	0	Teen	Auto & Vehicles	11-Jul-18	Varies with device	5.0 and up	NaN
1778	Block Craft 3D: Building Simulator Games For Free	GAME	4.5	1M	57M	50,000,000+	Free	0	Everyone	Simulation	5-Mar-18	2.10.2	4.0.3 and up	NaN
1781	Trivia Crack	GAME	4.5	6.4M	95M	100,000,000+	Free	0	Everyone	Trivia	3-Aug-18	2.79.0	4.1 and up	NaN

In [11]:

```
df.loc[df['Reviews'].str.contains('M')]
```

Out[11]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	new rev
72	Android Auto - Maps, Media, Messaging & Voice	AUTO_AND_VEHICLES	4.2	2M	16M	10,000,000+	Free	0	Teen	Auto & Vehicles	11-Jul-18	Varies with device	5.0 and up	NaN
1778	Block Craft 3D: Building Simulator Games For Free	GAME	4.5	1M	57M	50,000,000+	Free	0	Everyone	Simulation	5-Mar-18	2.10.2	4.0.3 and up	NaN
1781	Trivia Crack	GAME	4.5	6.4M	95M	100,000,000+	Free	0	Everyone	Trivia	3-Aug-18	2.79.0	4.1 and up	NaN

In [12]:

```
df.loc[df['Reviews'].str.contains('M'), 'Reviews'].str.replace('M', '')
```

Out[12]:

```
72        2
1778      1
1781    6.4
Name: Reviews, dtype: object
```

In [13]:

```
pd.to_numeric(df.loc[df['Reviews'].str.contains('M'), 'Reviews'
                    ].str.replace('M', ''))
# ye string se kaam krne ka method universal hai bahot jagah use kr skte ho
# to ise ache se ratta maar lio, mast method hai.
```

Out[13]:

```
72      2.0
1778    1.0
1781    6.4
Name: Reviews, dtype: float64
```

In [14]:

```
newrev = (pd.to_numeric(df.loc[df['Reviews'].str.contains('M'), 'Reviews'
                    ].str.replace('M', ''))*1000000).astype(str)
newrev
```

Out[14]:

```
72      2000000.0
1778    1000000.0
1781    6400000.0
Name: Reviews, dtype: object
```

In [15]:

```
df.loc[df['Reviews'].str.contains('M'), 'Reviews'] = (
    pd.to_numeric(df.loc[df['Reviews'].str.contains('M'),
                    'Reviews'
                    ].str.replace('M',''))*1000000).astype(str)
# yahan direct hum numeric nhi bana sakte kyunki sirf 3 values hi change
# kar rhe hai. phele sari values string mai krenge fir last mai pure reviews
# column ko numeric bana denge as asked.
```

In [16]:

```
df['Reviews'] = pd.to_numeric(df['Reviews'])
# sari string krdi numeric convert.
```

In [17]:

```
# dataset mai humne new column bana rkha hai df ['new rev'].
# use delete krna zaruri aage prblm krta hai iska compiler.
# no changes allowed in dataset.
del df['new rev']
```

4. How many duplicated apps are there?

In [18]:

```
df.loc[df.duplicated(subset=['App'], keep = False)].sort_values(by='App')

# keep false karne se duplicated value ke alava original bhi show krta hai.
# basically total number of occurences. keep default TRUE hota hai that means
# only duplicates are counted.
```

Out[18]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
1393	10 Best Foods for You	HEALTH_AND_FITNESS	4.0	2490.0	3.8M	500,000+	Free	0	Everyone 10+	Health & Fitness	17-Feb-17	1.9	2.3.3 and up
1407	10 Best Foods for You	HEALTH_AND_FITNESS	4.0	2490.0	3.8M	500,000+	Free	0	Everyone 10+	Health & Fitness	17-Feb-17	1.9	2.3.3 and up
2543	1800 Contacts - Lens Store	MEDICAL	4.7	23160.0	26M	1,000,000+	Free	0	Everyone	Medical	27-Jul-18	7.4.1	5.0 and up
2322	1800 Contacts - Lens Store	MEDICAL	4.7	23160.0	26M	1,000,000+	Free	0	Everyone	Medical	27-Jul-18	7.4.1	5.0 and up
2385	2017 EMRA Antibiotic Guide	MEDICAL	4.4	12.0	3.8M	1,000+	Paid	$16.99	Everyone	Medical	27-Jan-17	1.0.5	4.0.3 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3202	trivago: Hotels & Travel	TRAVEL_AND_LOCAL	4.2	219848.0	Varies with device	50,000,000+	Free	0	Everyone	Travel & Local	2-Aug-18	Varies with device	Varies with device
3118	trivago: Hotels & Travel	TRAVEL_AND_LOCAL	4.2	219848.0	Varies with device	50,000,000+	Free	0	Everyone	Travel & Local	2-Aug-18	Varies with device	Varies with device
3103	trivago: Hotels & Travel	TRAVEL_AND_LOCAL	4.2	219848.0	Varies with device	50,000,000+	Free	0	Everyone	Travel & Local	2-Aug-18	Varies with device	Varies with device
8291	wetter.com - Weather and Radar	WEATHER	4.2	189310.0	38M	10,000,000+	Free	0	Everyone	Weather	6-Aug-18	Varies with device	Varies with device
3652	wetter.com - Weather and Radar	WEATHER	4.2	189313.0	38M	10,000,000+	Free	0	Everyone	Weather	6-Aug-18	Varies with device	Varies with device

1979 rows × 13 columns

In [19]:

```
df.loc[df.duplicated(subset=['App'])].sort_values(by='App').shape
# excluding originals
```

Out[19]:

```
(1181, 13)
```

In [20]:

```
df.loc[df.duplicated(subset=['App'], keep = False)].sort_values(by='App').shape
# including orignals with duplicates as asked in question.
```

Out[20]:

```
(1979, 13)
```

5. Drop duplicated apps keeping the ones with the greatest number of reviews

In [21]:

```
df.sort_values(by = ['App','Reviews'], inplace = True)
# sort krli sari values by app and review order.
# inplace true matlab actual dataset mai changes kar rhe hai.
```

In [22]:

```
df.drop_duplicates(subset=['App'], keep = 'last', inplace = True)
# actual data set mai change krke duplicates drop krdiye keeping last value
# bcz uske reviews max honge.
```

6. Format the `Category` column

In [82]:

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

Out[82]:

```
Category
Family      1863
Game         943
Tools        825
Business     416
Medical      394
Name: count, dtype: int64
```

In [24]:

```
df['Category'] = df['Category'].str.replace('_',' ')
```

In [25]:

```
df['Category'] = df['Category'].str.capitalize()
# this str.capitalize function is used to capitalize first letter of a word
# as asked in the question
```

In [83]:

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

Out[83]:

```
Category
Family      1863
Game         943
Tools        825
Business     416
Medical      394
Name: count, dtype: int64
```

7. Clean and convert the `Installs` column to numeric type

In [27]:

```
df['Installs'].value_counts().head(5)
#remove + and ,
```

Out[27]:

```
Installs
1,000,000+     1416
100,000+       1113
10,000+        1028
10,000,000+     937
1,000+          885
Name: count, dtype: int64
```

In [28]:

```
df['Installs'] = df['Installs'].str.replace('+','').str.replace(',','')
df['Installs'].value_counts().head(5)
```

Out[28]:

```
Installs
1000000     1416
100000      1113
10000       1028
10000000     937
1000         885
Name: count, dtype: int64
```

In [29]:

```
df['Installs'] = pd.to_numeric(df['Installs'])
```

8. Clean and convert the `Size` column to numeric (representing bytes)

In [31]:

```
df.loc[df['Size'].str.contains('M'), 'Size'] = (
    pd.to_numeric(df.loc[df['Size'].str.contains('M'),
                         'Size'].str.replace('M',''))*(1024*1024)
                            ).astype(str)
```

In [32]:

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

Out[32]:

```
8884     3774873.6
324      9542041.6
8532    23068672.0
4541          203k
4636    55574528.0
Name: Size, dtype: object
```

In [33]:

```
df.loc[df['Size'].str.contains('k'), 'Size'] = (
    pd.to_numeric(df.loc[df['Size'].str.contains('k'),
                         'Size'].str.replace('k',''))*1024
                            ).astype(str)

#Phele jo dikkat aarhi thi error aarha tha vo isliye kyunki size column pura
# string type hai ab usmai aadhi string M wli convert krdi numeric mai to aadhi
# numeric adhi string hogyi fer jab code kiya k wli values ke liye tab error
# aarha tha. TO isliye ab humne convert krke numerics mai apna kaam kia multiply
# wla aur fir vapas use string type mai convert krke store kiya.
```

In [34]:

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

Out[34]:

```
8884     3774873.6
324      9542041.6
8532    23068672.0
4541      207872.0
4636    55574528.0
Name: Size, dtype: object
```

In [40]:

```
df.loc[df['Size'] == "Varies with device", 'Size'] = 0

# kuch sizes mai ye string likhi hui thi so isko zero set krdiya.
```

In [41]:

```
df['Size'] = pd.to_numeric(df['Size'])
```

9. Clean and convert the `Price` column to numeric

In [49]:
```
df['Price']
```

Out[49]:
```
8884          0
324           0
8532          0
4541          0
4636          0
          ...
6334          0
4362    $399.99
2575          0
7559          0
882           0
Name: Price, Length: 9648, dtype: object
```

In [48]:
```
df.loc[df['Price'] != "0"].head(3)
```

Out[48]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
7738	10 WPM Amateur ham radio CW Morse code trainer	Communication	3.5	10.0	3984588.8	100	Paid	$1.49	Everyone	Communication	12-May-18	2.1.4	2.1 and up
8219	10,000 Quotes DB (Premium)	Books and reference	4.1	70.0	3670016.0	500	Paid	$0.99	Everyone	Books & Reference	30-Aug-13	1.3	2.1 and up
6760	17th Edition Cable Sizer	Books and reference	4.4	47.0	1468006.4	1000	Paid	$3.08	Everyone	Books & Reference	27-May-16	1.22	2.2 and up

In [50]:
```
df.loc[df['Price'].str.contains('$'), 'Price'] = df.loc[df['Price'].str.contains('$'), 'Price'].str.replace('$','')
```

In [ ]:
```
df['Price'] = pd.to_numeric(df['Price'])

#ValueError: Unable to parse string "Free"
```

In [54]:
```
df.loc[df['Price'] == "Free"] = 0
```

In [55]:
```
df['Price'] = pd.to_numeric(df['Price'])
```

10. Paid or free?

In [57]:
```
df['Distribution'] = "Free"
# pure distribution column ko free set krdia. Ab jismai price ki value greater
# than 0 hai usmai distribution ko Paid set krdenge

df.loc[df['Price'] > 0 , 'Distribution'] = "Paid"
```

Analysis

11. Which app has the most reviews?

In [61]:
```
df.loc[df['Reviews'] == df['Reviews'].max(), 'App']
```

Out[61]:
```
2544    Facebook
Name: App, dtype: object
```

12. What category has the highest number of apps uploaded to the store?

In [64]:
```
df['Category'].value_counts().head(5)
```

Out[64]:
```
Category
Family      1863
Game         943
Tools        825
Business     416
Medical      394
Name: count, dtype: int64
```

13. To which category belongs the most expensive app?

In [65]:
```
df.loc[df['Price'] == df['Price'].max() , 'Category']
```

Out[65]:
```
4367    Lifestyle
Name: Category, dtype: object
```

14. What's the name of the most expensive game?

In [70]:
```
df.loc[(df['Category'] == "Game")].sort_values(by =
                                 'Price', ascending = False).head(3)
```

Out[70]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	Distribution
4203	The World Ends With You	Game	4.6	4108.0	13631488.0	10000	Paid	17.99	Everyone 10+	Arcade	14-Dec-15	1.0.4	4.0 and up	Paid
10782	Trine 2: Complete Story	Game	3.8	252.0	11534336.0	10000	Paid	16.99	Teen	Action	27-Feb-15	2.22	5.0 and up	Paid
6341	Blackjack Verite Drills	Game	4.6	17.0	4928307.2	100	Paid	14.00	Teen	Casino	9-Jul-17	1.1.10	3.0 and up	Paid

15. Which is the most popular Finance App?

In [72]:
```
df.loc[(df['Category'] == "Finance")].sort_values(by =
                                 'Installs', ascending = False).head(3)
```

Out[72]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	Distribution
5601	Google Pay	Finance	4.2	348132.0	0.0	100000000	Free	0.0	Everyone	Finance	26-Jul-18	2.70.206190089	Varies with device	Free
1156	PayPal	Finance	4.3	659760.0	49283072.0	50000000	Free	0.0	Everyone	Finance	18-Jul-18	6.28.0	4.4 and up	Free
1081	İşCep	Finance	4.5	381788.0	33554432.0	10000000	Free	0.0	Everyone	Finance	2-Aug-18	3.22.0	4.1 and up	Free

16. What Teen Game has the most reviews?

In [73]:
```
df.loc[(df['Category'] == "Game") & (df['Content Rating'] == "Teen")].sort_values(by =
                                 'Reviews', ascending = False).head(3)
```

Out[73]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	Distribution
3912	Asphalt 8: Airborne	Game	4.5	8389714.0	96468992.0	100000000	Free	0.0	Teen	Racing	4-Jul-18	3.7.1a	4.0.3 and up	Free
5417	Mobile Legends: Bang Bang	Game	4.4	8219586.0	103809024.0	100000000	Free	0.0	Teen	Action	24-Jul-18	1.2.97.3042	4.0.3 and up	Free
1988	Hungry Shark Evolution	Game	4.5	6074627.0	104857600.0	100000000	Free	0.0	Teen	Arcade	25-Jul-18	6.0.0	4.1 and up	Free

17. Which is the free game with the most reviews?

In [74]:
```
df.loc[(df['Category'] == "Game") & (df['Distribution'] == "Free")].sort_values(by =
                                 'Reviews', ascending = False).head(3)
```

Out[74]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver	Distribution
1879	Clash of Clans	Game	4.6	44893888.0	102760448.0	100000000	Free	0.0	Everyone 10+	Strategy	15-Jul-18	10.322.16	4.1 and up	Free
1917	Subway Surfers	Game	4.5	27725352.0	79691776.0	1000000000	Free	0.0	Everyone 10+	Arcade	12-Jul-18	1.90.0	4.1 and up	Free
1878	Clash Royale	Game	4.6	23136735.0	101711872.0	100000000	Free	0.0	Everyone 10+	Strategy	27-Jun-18	2.3.2	4.1 and up	Free

18. How many TB (terabytes) were transferred (overall) for the most popular Lifestyle app?

In [78]:
```
# Terabytes transferred meaning Size*Installs.
app = df.loc[(df['Category'] == "Lifestyle")].sort_values(by='Installs', ascending = False).iloc[0]
app
```

Out[78]:
```
App                   Tinder
Category           Lifestyle
Rating                   4.0
Reviews            2789775.0
Size              71303168.0
Installs           100000000
Type                    Free
Price                    0.0
Content Rating    Mature 17+
Genres             Lifestyle
Last Updated        2-Aug-18
Current Ver            9.5.0
Android Ver       4.4 and up
Distribution            Free
Name: 4587, dtype: object
```

In [79]:
```
app['Installs']*app['Size'] / (1024*1024*1024*1024)
```

Out[79]:
```
6484.9853515625
```

```
In [81]: tb = round(app['Installs']*app['Size'] / (1024*1024*1024*1024),0)
         tb
```

```
Out[81]: 6485.0
```

```
In [ ]:
```