

```
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	
1736	Roll the Ball® - slide puzzle	GAME	4.5	1385093	35M	100,000,000+	Free	0	1
2262	Super Hearing Secret Voices Recorder PRO	MEDICAL	5.0	3	23M	100+	Paid	\$2.99	1
3773	World Newspapers	NEWS_AND_MAGAZINES	4.4	185884	7.5M	1,000,000+	Free	0	
1571	Entel	LIFESTYLE	3.2	16168	55M	1,000,000+	Free	0	1
8476	English for Everyone	FAMILY	3.2	429	18M	10,000+	Free	0	1

```
In [3]: df.head()
```

Out[3]:

	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	Free	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 [4]: 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 [5]: `df.shape`

Out[5]: (10841, 13)

## ▼ Data Cleaning

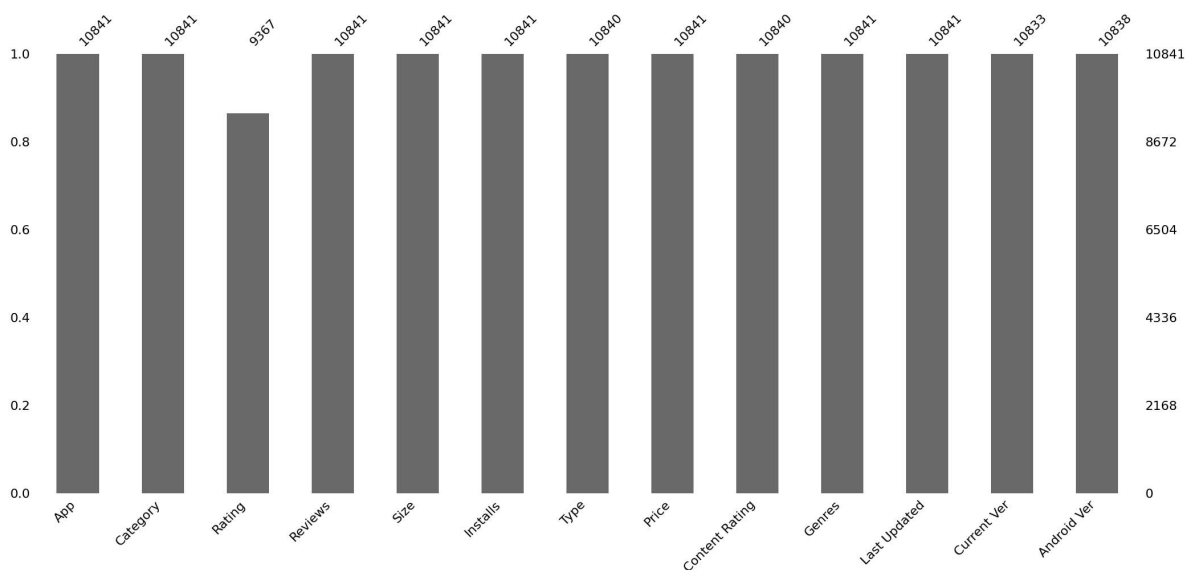
### ▼ 1. Which of the following column(s) has/have null values?

In [6]: `df.isnull().sum()`

```
Out[6]: 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
```

In [7]: `msno.bar(df)`

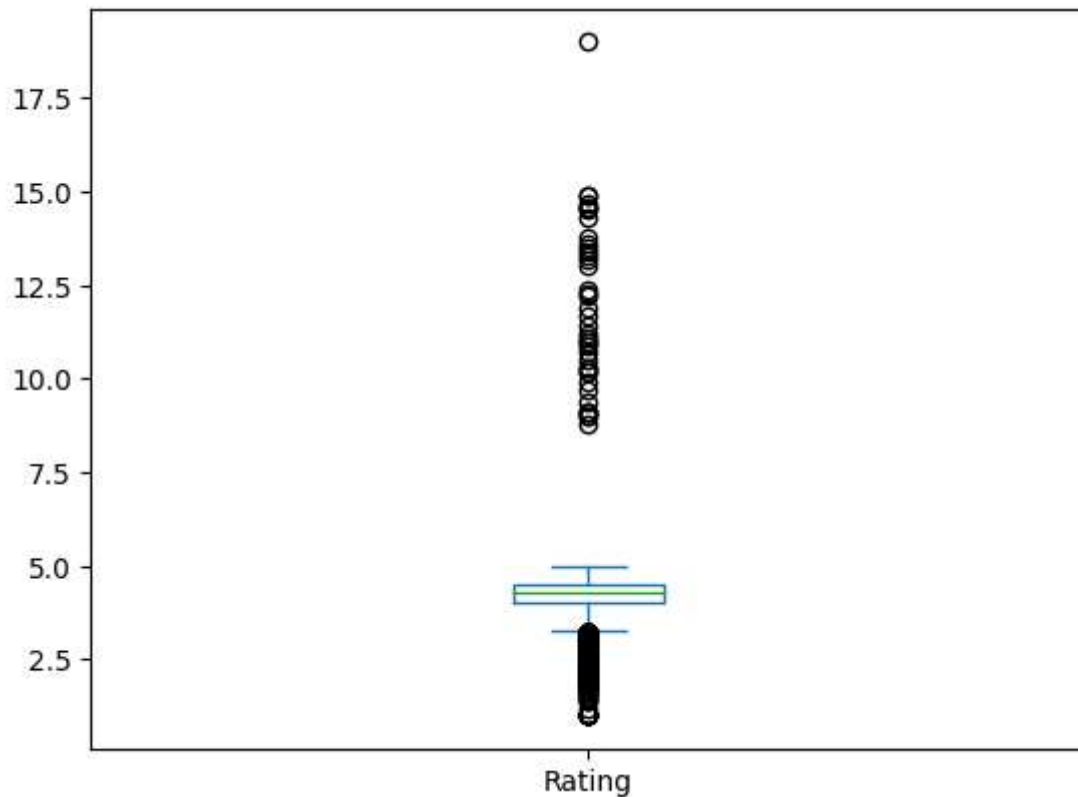
Out[7]: <Axes: >



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

```
In [8]: df['Rating'].plot(kind='box')
```

```
Out[8]: <Axes: >
```



```
In [9]: df.loc[df['Rating']>5, 'Rating']=np.nan
```

```
In [10]: df['Rating'].mean()
```

```
Out[10]: 4.197726785331332
```

```
In [11]: df['Rating'].fillna(df['Rating'].mean(), inplace=True)
```

```
In [12]: df.dropna(inplace=True)
```

### ▼ 3. Clean the column Reviews and make it numeric

```
In [15]: df['Reviews Numeric']=pd.to_numeric(df['Reviews'], errors='coerce')
```

```
In [16]: df.loc[df['Reviews Numeric'].isna()]
```

```
Out[16]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Con Ra
72	Android Auto - Maps, Media, Messaging & Voice	AUTO_AND_VEHICLES	4.2	2M	16M	10,000,000+	Free	0	1
1778	Block Craft 3D: Building Simulator Games For Free	GAME	4.5	1M	57M	50,000,000+	Free	0	Every
1781	Trivia Crack	GAME	4.5	6.4M	95M	100,000,000+	Free	0	Every

```
In [22]: new_reviews=(
    pd.to_numeric(
        df.loc[df['Reviews'].str.contains('M'),'Reviews'].str.replace('M','')
    )*1_00_000).astype(str)
new_reviews
```

```
Out[22]: 72      200000.0
1778     100000.0
1781     640000.0
Name: Reviews, dtype: object
```

```
In [23]: df.loc[df['Reviews'].str.contains('M'),'Reviews'] = (
    pd.to_numeric(
        df.loc[df['Reviews'].str.contains('M'),'Reviews'].str.replace('M','')
    )*1_00_000).astype(str)
```

```
In [25]: df['Reviews']=pd.to_numeric(df['Reviews'])
```

#### ▼ 4. How many duplicated apps are there?

```
In [28]: df.duplicated(keep=False).sum()
```

```
Out[28]: 880
```

```
In [29]: df.loc[df.duplicated(subset=['App'],keep=False)& ~df.duplicated(keep=False)].s
```

Out[29]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genre
<b>3083</b>	365Scores - Live Scores	SPORTS	4.6	666521.0	25M	10,000,000+	Free	0	Everyone	Spor
<b>5415</b>	365Scores - Live Scores	SPORTS	4.6	666246.0	25M	10,000,000+	Free	0	Everyone	Spor
<b>1675</b>	8 Ball Pool	GAME	4.5	14198297.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>1703</b>	8 Ball Pool	GAME	4.5	14198602.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>1755</b>	8 Ball Pool	GAME	4.5	14200344.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>1844</b>	8 Ball Pool	GAME	4.5	14200550.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>3953</b>	8 Ball Pool	SPORTS	4.5	14184910.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>1871</b>	8 Ball Pool	GAME	4.5	14201891.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>1970</b>	8 Ball Pool	GAME	4.5	14201604.0	52M	100,000,000+	Free	0	Everyone	Spor
<b>662</b>	95Live - SG#1 Live Streaming App	DATING	4.1	4954.0	15M	1,000,000+	Free	0	Teen	Datir

```
In [30]: df.duplicated(subset=['App'],keep=False).sum()
```

Out[30]: 1979

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

```
In [31]: df.loc[
    df.duplicated(subset=['App'],keep=False)& ~df.duplicated(keep=False)
].sort_values(by=['App','Reviews']).head(10)
```

Out[31]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genre
<b>5415</b>	365Scores - Live Scores	SPORTS	4.6	666246.0	25M	10,000,000+	Free	0	Everyone	Sports
<b>3083</b>	365Scores - Live Scores	SPORTS	4.6	666521.0	25M	10,000,000+	Free	0	Everyone	Sports
<b>3953</b>	8 Ball Pool	SPORTS	4.5	14184910.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>1675</b>	8 Ball Pool	GAME	4.5	14198297.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>1703</b>	8 Ball Pool	GAME	4.5	14198602.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>1755</b>	8 Ball Pool	GAME	4.5	14200344.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>1844</b>	8 Ball Pool	GAME	4.5	14200550.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>1970</b>	8 Ball Pool	GAME	4.5	14201604.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>1871</b>	8 Ball Pool	GAME	4.5	14201891.0	52M	100,000,000+	Free	0	Everyone	Sports
<b>559</b>	95Live - SG#1 Live Streaming App	DATING	4.1	4953.0	15M	1,000,000+	Free	0	Teen	Dating



```
In [33]: #df_copy=df.copy()
```

```
In [35]: df.sort_values(by=['App', 'Reviews'],inplace=True)
```

```
In [36]: del df['Reviews Numeric']
```

```
In [37]: df.drop_duplicates(subset=['App'], keep='last', inplace=True)
```

## ▼ 6. Format the Category column

```
In [41]: df['Category'].value_counts()
```

```
Out[41]: Category
Family          1874
Game            945
Tools           827
Business        420
Medical         395
Productivity    374
Personalization 374
Lifestyle       369
Finance         345
Sports          325
Communication   315
Health and fitness 288
Photography     281
News and magazines 254
Social          239
Books and reference 221
Travel and local 219
Shopping        202
Dating          170
Video players   164
Maps and navigation 131
Food and drink  112
Education       105
Entertainment   86
Auto and vehicles 85
Libraries and demo 83
Weather         79
House and home  73
Events          64
Art and design  60
Parenting       60
Comics          56
Beauty          53
Name: count, dtype: int64
```

```
In [39]: df['Category']=df['Category'].str.replace('_', ' ')
```

```
In [40]: df['Category']=df['Category'].str.capitalize()
```

▼ **7. Clean and convert the *Installs* column to numeric type**



```
In [42]: df.loc[pd.to_numeric(df['Installs'],errors='coerce').isna()].head()
```

```
Out[42]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
8884	"i DT" Fútbol. Todos Somos Técnicos.	Sports	4.197727	27.0	3.6M	500+	Free	0	Everyone
324	#NAME?	Comics	3.500000	115.0	9.1M	10,000+	Free	0	Mature 17+
8532	+Download 4 Instagram Twitter	Social	4.500000	40467.0	22M	1,000,000+	Free	0	Everyone
4541	.R	Tools	4.500000	259.0	203k	10,000+	Free	0	Everyone
4636	/u/app	Communication	4.700000	573.0	53M	10,000+	Free	0	Mature 17+ C

```
In [43]: pd.to_numeric(df['Installs'].str.replace('+', '').str.replace(',', ''))
```

```
Out[43]: 8884      500
324      10000
8532     1000000
4541      10000
4636      10000
...
6334     100000
4362      10000
2575     1000000
7559      10000
882      1000000
Name: Installs, Length: 9648, dtype: int64
```

```
In [44]: df['Installs']=pd.to_numeric(df['Installs'].str.replace('+', '').str.replace(',',''))
```

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

```
In [46]: df['Size'] = df['Size'].replace('Varies with device', "0").astype(str)
```

```
In [47]: new_value = (pd.to_numeric(
    df.loc[df['Size'].str.contains('M'), 'Size'].str.replace('M', ''))
    * (1024 * 1024)).astype(str)
df.loc[df['Size'].str.contains('M'), 'Size'] = new_value
```

```
In [48]: new_value = (pd.to_numeric(
            df.loc[df['Size'].str.contains('k'), 'Size'].str.replace('k', ''))
            * 1024).astype(str)
df.loc[df['Size'].str.contains('k'), 'Size'] = new_value
```

```
In [49]: df['Size'] = df['Size'].str.replace('+', '')
df['Size'] = df['Size'].str.replace(',', '')
```

```
In [50]: df['Size'] = pd.to_numeric(df['Size'])
```

## ▼ 9. Clean and convert the Price column to numeric

```
In [52]: df.loc[df['Price'] == 'Free', 'Price'] = "0"
df['Price'] = df['Price'].str.replace('$', '').str.replace(',', '.')
df['Price'] = pd.to_numeric(df['Price'])
```

## ▼ 10. Paid or free?

```
In [56]: df['Price'].apply(lambda p: "Free" if p>0 else "Paid")
```

```
Out[56]: 8884    Paid
          324    Paid
          8532   Paid
          4541   Paid
          4636   Paid
          ...
          6334   Paid
          4362   Free
          2575   Paid
          7559   Paid
          882    Paid
          Name: Price, Length: 9648, dtype: object
```

```
In [57]: df['Distribution'] = 'Free'
```

```
In [58]: df.loc[df['Price']>0, 'Distribution']='Paid'
```

## ▼ Analysis

### ▼ 11. What company has the most reviews?

```
In [60]: df.sort_values(by='Reviews',ascending=False).head()
```

Out[60]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Co R
2544	Facebook	Social	4.1	78158306.0	0.0	1000000000	Free	0.0	
381	WhatsApp Messenger	Communication	4.4	69119316.0	0.0	1000000000	Free	0.0	Eve
2604	Instagram	Social	4.5	66577446.0	0.0	1000000000	Free	0.0	
382	Messenger – Text and Video Chat for Free	Communication	4.0	56646578.0	0.0	1000000000	Free	0.0	Eve
1879	Clash of Clans	Game	4.6	44893888.0	102760448.0	1000000000	Free	0.0	Eve

▼ 12. Which is the category with the most most uploaded apps?




```
In [61]: df['Category'].value_counts()
```

```
Out[61]: Category
Family          1874
Game            945
Tools           827
Business        420
Medical         395
Productivity    374
Personalization 374
Lifestyle       369
Finance         345
Sports          325
Communication   315
Health and fitness 288
Photography     281
News and magazines 254
Social          239
Books and reference 221
Travel and local 219
Shopping        202
Dating          170
Video players   164
Maps and navigation 131
Food and drink  112
Education       105
Entertainment   86
Auto and vehicles 85
Libraries and demo 83
Weather         79
House and home  73
Events          64
Art and design  60
Parenting       60
Comics          56
Beauty          53
Name: count, dtype: int64
```

▼ **13. To which category belongs the most expensive app?**

```
In [62]: df.sort_values(by='Price',ascending=False)
```

Out[62]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Cor Ri
4367	I'm Rich - Trump Edition	Lifestyle	3.6	275.0	7654604.8	10000	Paid	400.00	Ever
5358	I am Rich!	Finance	3.8	93.0	23068672.0	1000	Paid	399.99	Ever
5356	I Am Rich Premium	Finance	4.1	1867.0	4928307.2	50000	Paid	399.99	Ever
5362	I Am Rich Pro	Family	4.4	201.0	2831155.2	5000	Paid	399.99	Ever
4197	most expensive app (H)	Family	4.3	6.0	1572864.0	100	Paid	399.99	Ever
...	...	...	...	...	...	...	...	...	...
10438	Dolphin and fish coloring book	Family	3.9	2249.0	0.0	500000	Free	0.00	Ever
3434	Dolphins Live Wallpaper	Personalization	4.2	25807.0	5767168.0	1000000	Free	0.00	Ever
1242	Domino's Pizza USA	Food and drink	4.7	1032935.0	0.0	10000000	Free	0.00	Ever
2158	Dominos Game 	Family	4.1	2903.0	16777216.0	1000000	Free	0.00	Ever
882	 Football Wallpapers 4K   Full HD Backgrounds 	Entertainment	4.7	11661.0	4194304.0	1000000	Free	0.00	Ever

9648 rows × 14 columns

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

```
In [65]: df.query("Category == 'Game' ").sort_values(by='Price',ascending=False)
```

Out[65]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
4203	The World Ends With You	Game	4.6	4108.0	13631488.0	10000	Paid	17.99	Everyone 10+
10782	Trine 2: Complete Story	Game	3.8	252.0	11534336.0	10000	Paid	16.99	Teen
6341	Blackjack Verite Drills	Game	4.6	17.0	4928307.2	100	Paid	14.00	Teen
1838	Star Wars ™: DIRTY	Game	4.5	38207.0	15728640.0	100000	Paid	9.99	Teen
6198	Backgammon NJ for Android	Game	4.4	1644.0	15728640.0	10000	Paid	7.99	Everyone
...	...	...	...	...	...	...	...	...	...
7600	Dreamland Arcade - Steven Universe	Game	4.0	6386.0	25165824.0	500000	Free	0.00	Everyone
10522	Drift Legends	Game	4.2	33788.0	28311552.0	1000000	Free	0.00	Everyone
4434	Drink-O-Tron The Drinking Game	Game	4.1	140.0	47185920.0	50000	Free	0.00	Mature 17+
10508	Drive 4x4 Luxury SUV Jeep	Game	4.2	2183.0	48234496.0	500000	Free	0.00	Everyone
3960	► MultiCraft — Free Miner! 🍷	Game	4.3	1305050.0	0.0	50000000	Free	0.00	Everyone 10+

945 rows × 14 columns

```
In [66]: df.loc[df['Category'] == 'Game'].sort_values(by='Price', ascending=False).head
```

```
Out[66]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genre
4203	The World Ends With You	Game	4.6	4108.0	13631488.0	10000	Paid	17.99	Everyone 10+	Ar
10782	Trine 2: Complete Story	Game	3.8	252.0	11534336.0	10000	Paid	16.99	Teen	A
6341	Blackjack Verite Drills	Game	4.6	17.0	4928307.2	100	Paid	14.00	Teen	Ce
1838	Star Wars™: DIRT	Game	4.5	38207.0	15728640.0	100000	Paid	9.99	Teen	Pl
6198	Backgammon NJ for Android	Game	4.4	1644.0	15728640.0	10000	Paid	7.99	Everyone	B

#### 15. Which is the most popular Finance App?

```
In [68]: df.query("Category == 'Finance' ").sort_values(by='Installs',ascending=False).
```

```
Out[68]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genre
5601	Google Pay	Finance	4.2	348132.0	0.0	100000000	Free	0.0	Everyone	Finance
1156	PayPal	Finance	4.3	659760.0	49283072.0	50000000	Free	0.0	Everyone	Finance
1081	İşCep	Finance	4.5	381788.0	33554432.0	10000000	Free	0.0	Everyone	Finance
1168	Wells Fargo Mobile	Finance	4.4	250719.0	38797312.0	10000000	Free	0.0	Everyone	Finance
1169	Capital One® Mobile	Finance	4.6	510401.0	82837504.0	10000000	Free	0.0	Everyone	Finance

#### 16. What Teen Game has the most reviews?

```
In [69]: df['Content Rating'].value_counts()
```

```
Out[69]: Content Rating
Everyone      7893
Teen          1036
Mature 17+    393
Everyone 10+   321
Adults only 18+ 3
Unrated        2
Name: count, dtype: int64
```

```
In [71]: df.query("Category=='Game' and `Content Rating`=='Teen'").sort_values(by='Reviews')
```

```
Out[71]:
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genre
3912	Asphalt 8: Airborne	Game	4.5	8389714.0	96468992.0	100000000	Free	0.0	Teen	R
5417	Mobile Legends: Bang Bang	Game	4.4	8219586.0	103809024.0	100000000	Free	0.0	Teen	/
1988	Hungry Shark Evolution	Game	4.5	6074627.0	104857600.0	100000000	Free	0.0	Teen	A
10327	Garena Free Fire	Game	4.5	5534114.0	55574528.0	100000000	Free	0.0	Teen	/
3967	Pixel Gun 3D: Survival shooter & Battle Royale	Game	4.5	4487182.0	57671680.0	50000000	Free	0.0	Teen	/

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



In [74]: `df.query("Category=='Game' and Price==0 ").sort_values(by='Reviews',ascending=`

Out[74]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	
1879	Clash of Clans	Game	4.6	44893888.0	102760448.0	100000000	Free	0.0	Everyone 10+	S
1917	Subway Surfers	Game	4.5	27725352.0	79691776.0	1000000000	Free	0.0	Everyone 10+	
1878	Clash Royale	Game	4.6	23136735.0	101711872.0	100000000	Free	0.0	Everyone 10+	S
1966	Candy Crush Saga	Game	4.4	22430188.0	77594624.0	500000000	Free	0.0	Everyone	
1908	My Talking Tom	Game	4.5	14892469.0	0.0	500000000	Free	0.0	Everyone	

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

In [77]: `app=df.query("Category == 'Lifestyle'").sort_values(by='Installs',ascending=Fa`

In [78]: `(app['Installs']*app['Size'])/(1024*1024*1024*1024)`

Out[78]: 6484.9853515625

In [ ]: