


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
# read given raw data
data = pd.DataFrame(pd.read_csv("Google Apps data.csv"))
```

```
data.head()
```



	Unnamed: 0.1	Unnamed: 0	App	Category	Rating	Reviews	Size	Installs	Type	Price
0	0	0	Photo Editor & Candy Camera & Grid & ScrapBook	Art And Design	4.1	159	19.0	10000	Free	0.0
1	1	1	Coloring book moana	Art And Design	3.9	967	14.0	500000	Free	0.0
2	2	5	U Launcher Lite – FREE Live Customization	Art And Design	4.7	87510	8.7	5000000	Free	0.0

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8276 entries, 0 to 8275
Data columns (total 15 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Unnamed: 0.1        8276 non-null  int64
1   Unnamed: 0          8276 non-null  int64
2   App                 8276 non-null  object
3   Category            8276 non-null  object
4   Rating              8276 non-null  float64
5   Reviews             8276 non-null  int64
6   Size                8276 non-null  float64
7   Installs            8276 non-null  int64
8   Type                8276 non-null  object
9   Price               8276 non-null  float64
10  Content Rating      7915 non-null  object
11  Last Updated        8276 non-null  object
12  Current Ver         8276 non-null  object
13  Minimum Android Ver 8276 non-null  object
14  Genres              8276 non-null  object
dtypes: float64(3), int64(4), object(8)
memory usage: 970.0+ KB
```

```
## DATA CLEANING and EXPLORATION
```

```
data.describe()
```

	Unnamed: 0.1	Unnamed: 0	Rating	Reviews	Size	Install
count	8276.000000	8276.000000	8276.000000	8.276000e+03	8276.000000	8.276000e+0
mean	4137.500000	4560.609957	4.175121	2.803270e+05	18.897761	9.658206e+0
std	2389.219747	2560.879748	0.534762	2.096170e+06	22.376521	5.986505e+0
min	0.000000	0.000000	1.000000	1.000000e+00	0.008300	1.000000e+0
25%	2068.750000	2459.750000	4.000000	1.290000e+02	2.800000	1.000000e+0
50%	4137.500000	4613.500000	4.300000	3.213500e+03	9.500000	1.000000e+0
75%	6206.250000	6765.250000	4.500000	4.627800e+04	27.000000	1.000000e+0

```
# Checking the null values
```

```
data.isnull().sum()
```

```
Unnamed: 0.1      0
Unnamed: 0        0
App              0
Category         0
Rating          0
Reviews         0
Size            0
Installs        0
Type            0
Price           0
Content Rating   361
Last Updated     0
Current Ver      0
Minimum Android Ver 0
Genres          0
dtype: int64
```

```
# Dropping the null values from the content rating column
data= data.dropna()
```

```
data['Rating'] = data['Rating'].astype(float)
```

```
# Dropping the Duplicate values
data = data.drop_duplicates()
```

```
# Removing the unnecessary columns
column_to_drop = 'Unnamed: 0.1'
data.drop(columns=column_to_drop, inplace=True)
```

```
#Renaming the desired columns
data.rename(columns={'Unnamed: 0': 'S.No'}, inplace=True)
```

```
data.head(2)
```

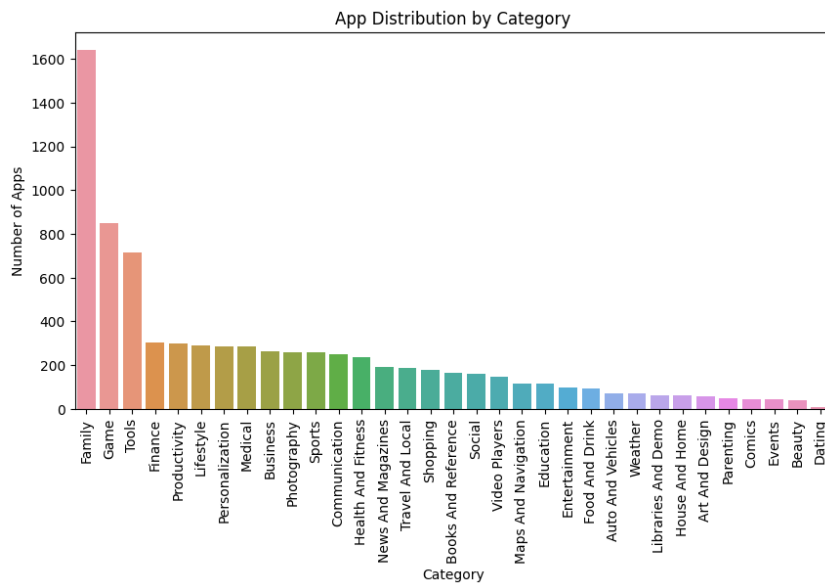
	S.No	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	0	Photo Editor & Candy Camera & Candy	Art And Design	4.1	159	19.0	10000	Free	0.0	Others

```
data.isna().sum()
```

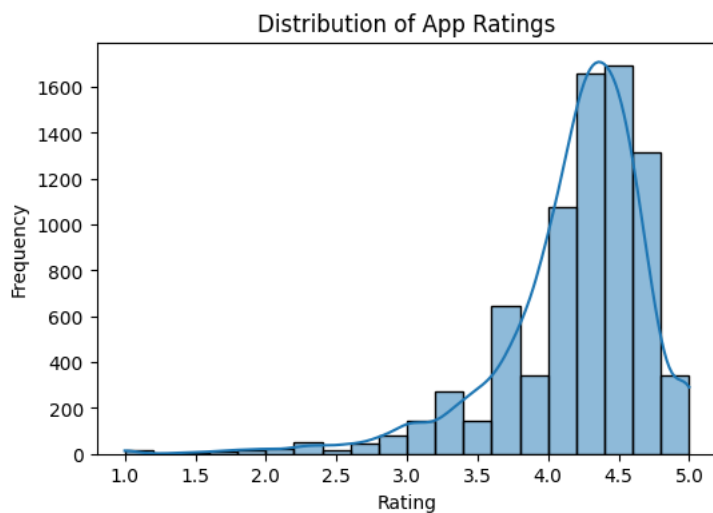
```
S.No      0
App       0
Category  0
Rating    0
Reviews   0
Size      0
Installs  0
Type      0
Price     0
Content Rating 0
Last Updated 0
Current Ver  0
Minimum Android Ver 0
Genres      0
dtype: int64
```

```
# Category analysis
category_counts = data['Category'].value_counts()
plt.figure(figsize=(10, 5))
```

```
sns.barplot(x=category_counts.index, y=category_counts.values)
plt.xticks(rotation=90)
plt.xlabel('Category')
plt.ylabel('Number of Apps')
plt.title('App Distribution by Category')
plt.show()
```

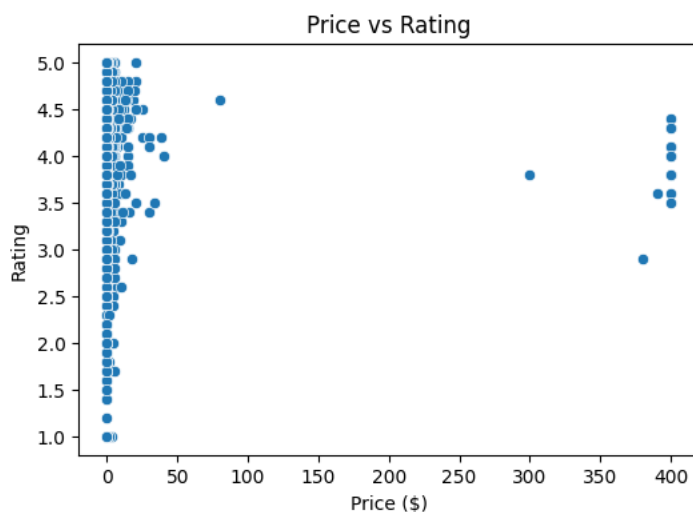


```
# Ratings distribution
plt.figure(figsize=(6, 4))
sns.histplot(data['Rating'], bins=20, kde=True)
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.title('Distribution of App Ratings')
plt.show()
```

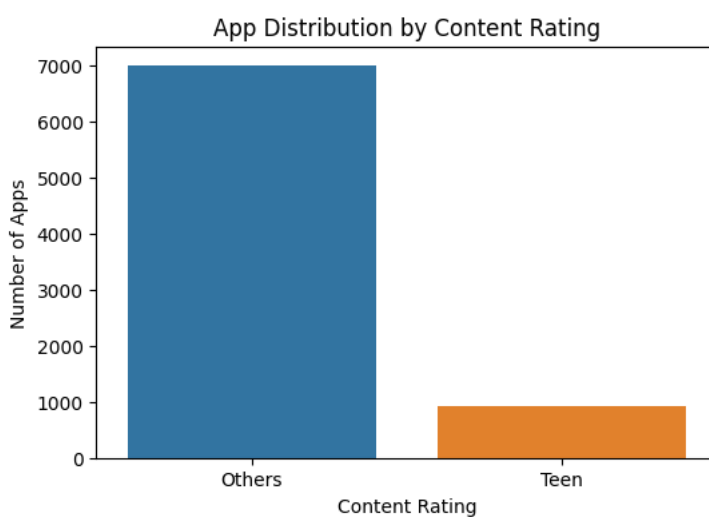


```
#Price vs Rating analysis
plt.figure(figsize=(6, 4))
sns.scatterplot(x='Price', y='Rating', data=data)
plt.xlabel('Price ($)')
plt.ylabel('Rating')
```

```
plt.title('Price vs Rating')
plt.show()
```

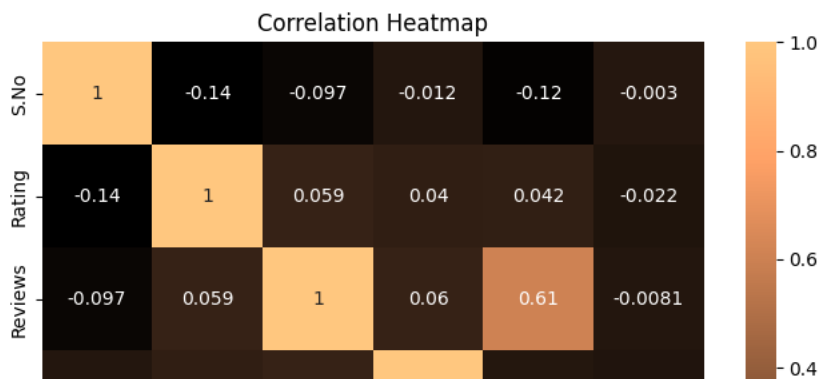


```
# Content Rating analysis
content_rating_counts = data['Content Rating'].value_counts()
plt.figure(figsize=(6, 4))
sns.barplot(x=content_rating_counts.index, y=content_rating_counts.values)
plt.xlabel('Content Rating')
plt.ylabel('Number of Apps')
plt.title('App Distribution by Content Rating')
plt.show()
```



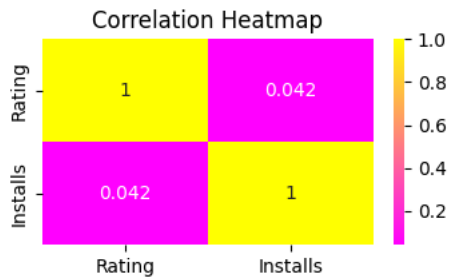
```
# Correlation heatmap
corr_matrix = data.corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='copper')
plt.title('Correlation Heatmap')
plt.show()
```

```
<ipython-input-26-cb585996846b>:2: FutureWarning: The default value of numeric_only
corr_matrix = data.corr()
```



We understand that the reviews and installs have high correlation , next comes the reviews and ratings

```
# Correlation Analysis (for numeric columns)
numeric_columns = ['Rating', 'Installs']
correlation_matrix = data[numeric_columns].corr()
plt.figure(figsize=(4, 2))
sns.heatmap(correlation_matrix, annot=True, cmap='spring')
plt.title('Correlation Heatmap')
plt.show()
```

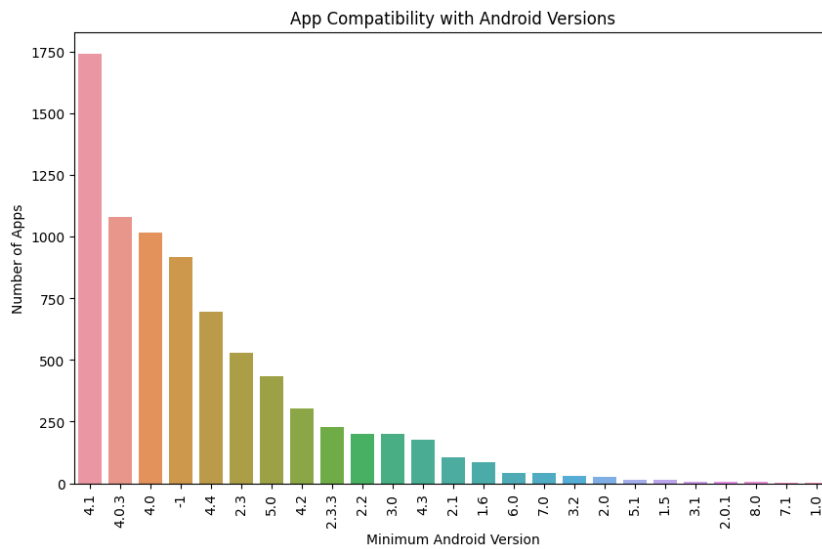


```
# Version Analysis
version_counts = data['Current Ver'].value_counts()
plt.figure(figsize=(8, 4))
sns.barplot(x=version_counts.index[:10], y=version_counts.values[:10])
plt.xticks(rotation=90)
plt.xlabel('Current Version')
plt.ylabel('Number of Apps')
plt.title('Top 10 App Versions')
plt.show()
```

Top 10 App Versions

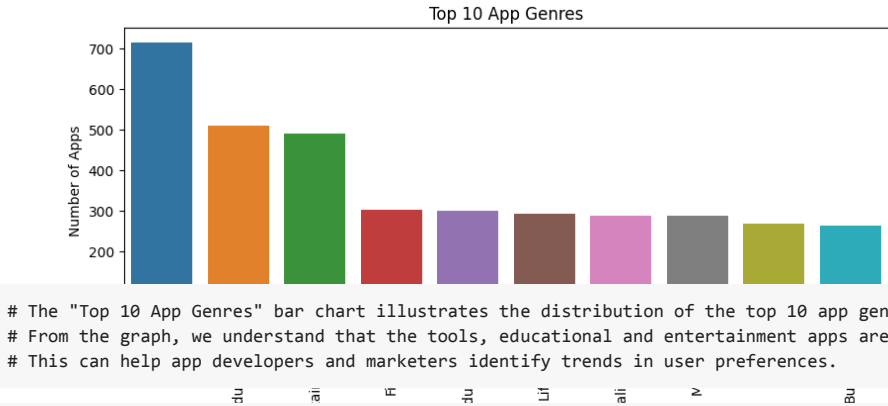
```
# The "Top 10 App Versions" bar chart shows the distribution of the top 10 app versions.
# This can help developers understand which versions are most prevalent among users.
# Developers can focus their support and updates on these popular versions.
```

```
# Android Version Compatibility Analysis
android_version_counts = data['Minimum Android Ver'].value_counts()
plt.figure(figsize=(10, 6))
sns.barplot(x=android_version_counts.index, y=android_version_counts.values)
plt.xticks(rotation=90)
plt.xlabel('Minimum Android Version')
plt.ylabel('Number of Apps')
plt.title('App Compatibility with Android Versions')
plt.show()
```



```
# The "App Compatibility with Android Versions" bar chart displays the distribution of apps based on their minimum Android version requirement.
# This can help developers determine the range of Android versions they need to target when developing or updating their apps.
```

```
# Genre Analysis
genre_counts = data['Genres'].value_counts()
plt.figure(figsize=(10, 4))
sns.barplot(x=genre_counts.index[:10], y=genre_counts.values[:10])
plt.xticks(rotation=90)
plt.xlabel('Genre')
plt.ylabel('Number of Apps')
plt.title('Top 10 App Genres')
plt.show()
```



```
data.to_csv('cleaned_dataGoogleAppstoreDA.csv', index=False)
```