Aarush Coaching Classes

Google Playstore Data Set Analytics

In [2]:
```
import pandas as pd
data=pd.read_csv("googleplaystore.csv")
data.head()
```

Out[2]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Androi Ve
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19000.0	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	14000.0	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	8700.0	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	25000.0	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	2800.0	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

In [3]:
```
data.shape
```

Out[3]:
```
(10841, 13)
```

In [4]:
```
data.isnull().sum()
```

Out[4]:
```
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
```

Data Cleaning

In [5]:
```
# Remove rows with null values
data.dropna(inplace=True)
data.shape
```

Out[5]:
```
(9360, 13)
```

In [6]:
```
# Convert reviews columns to numeric
data['Reviews']=pd.to_numeric(data['Reviews'],errors="coerce")
```

In [7]:
```
def convert_size(size):
    if 'M' in size:
        return float(size.replace('M', ''))*1000
    elif 'K' in size:
        return float(size.replace('K', ''))
    else:
        return None

data['Size']=data['Size'].apply(convert_size)
data.dropna(subset=['size'],inplace=True)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 9
      6     else:
      7         return None
----> 9 data['Size']=data['Size'].apply(convert_size)
     10 data.dropna(subset=['size'],inplace=True)

File D:\New folder\Lib\site-packages\pandas\core\series.py:4764, in Series.apply(self, func, convert_dtype, args, by_row, **kwa
rgs)
   4629 def apply(
   4630     self,
   4631     func: AggFuncType,
   (...)
   4636     **kwargs,
   4637 ) -> DataFrame | Series:
   4638     """
   4639     Invoke function on values of Series.
   4640
   (...)
   4755     dtype: float64
   4756     """
   4757     return SeriesApply(
   4758         self,
   4759         func,
   4760         convert_dtype=convert_dtype,
   4761         by_row=by_row,
   4762         args=args,
   4763         kwargs=kwargs,
-> 4764     ).apply()

File D:\New folder\Lib\site-packages\pandas\core\apply.py:1209, in SeriesApply.apply(self)
   1206     return self.apply_compat()
   1208 # self.func is Callable
-> 1209 return self.apply_standard()

File D:\New folder\Lib\site-packages\pandas\core\apply.py:1289, in SeriesApply.apply_standard(self)
   1283 # row-wise access
   1284 # apply doesn't have a `na_action` keyword and for backward compat reasons
   1285 # we need to give `na_action="ignore"` for categorical data.
```

```
   1286 # TODO: remove the `na_action="ignore"` when that default has been changed in
   1287 #  Categorical (GH51645).
   1288 action = "ignore" if isinstance(obj.dtype, CategoricalDtype) else None
-> 1289 mapped = obj._map_values(
   1290     mapper=curried, na_action=action, convert=self.convert_dtype
   1291 )
   1293 if len(mapped) and isinstance(mapped[0], ABCSeries):
   1294     # GH#43986 Need to do list(mapped) in order to get treated as nested
   1295     #  See also GH#25959 regarding EA support
   1296     return obj._constructor_expanddim(list(mapped), index=obj.index)

File D:\New folder\Lib\site-packages\pandas\core\base.py:921, in IndexOpsMixin._map_values(self, mapper, na_action, convert)
    918 if isinstance(arr, ExtensionArray):
    919     return arr.map(mapper, na_action=na_action)
--> 921 return algorithms.map_array(arr, mapper, na_action=na_action, convert=convert)

File D:\New folder\Lib\site-packages\pandas\core\algorithms.py:1814, in map_array(arr, mapper, na_action, convert)
   1812 values = arr.astype(object, copy=False)
   1813 if na_action is None:
-> 1814     return lib.map_infer(values, mapper, convert=convert)
   1815 else:
   1816     return lib.map_infer_mask(
   1817         values, mapper, mask=isna(values).view(np.uint8), convert=convert
   1818     )

File lib.pyx:2926, in pandas._libs.lib.map_infer()

Cell In[7], line 2, in convert_size(size)
      1 def convert_size(size):
----> 2     if 'M' in size:
      3         return float(size.replace('M', ''))*1000
      4     elif 'K' in size:

TypeError: argument of type 'float' is not iterable
```

In [8]:
```
# Remove the '+' and ',' from 'Installs' and convert to numeric
data['Installs'] = data['Installs'].str.replace('+', '').str.replace(',', '')
data['Installs'] = pd.to_numeric(data['Installs'], errors='coerce')

# Remove the '$' sign from 'Price' and convert to numeric
data['Price'] = data['Price'].str.replace('$', '')
```

```
data['Price'] = pd.to_numeric(data['Price'], errors='coerce')

data.head()
```

Out[8]:

	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	19000.0	10000	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14000.0	500000	Free	0.0	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	8700.0	5000000	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25000.0	50000000	Free	0.0	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	2800.0	100000	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

Imputations

In [9]:
```
# Impute missing values in numeric columns with mean/median
data['Rating'].fillna(data['Rating'].mean(), inplace=True)
data['Reviews'].fillna(data['Reviews'].median(), inplace=True)
data['Size'].fillna(data['Size'].median(), inplace=True)
data['Installs'].fillna(data['Installs'].median(), inplace=True)
data['Price'].fillna(data['Price'].median(), inplace=True)
```

In [10]:
```
# Impute missing values in categorical columns with mode
data['Category'].fillna(data['Category'].mode()[0], inplace=True)
```

In [11]:
```
## Remove duplicates
data.drop_duplicates(inplace=True)
data.shape
```

Out[11]: (8886, 13)

In [12]:
```
#Convert the text data to lowercase and strip the whitespace
data['App']=data['App'].str.lower().str.strip()
data['Category']=data['Category'].str.lower().str.strip()
```

In [13]:
```
#Handle the outliers
data=data[(data['Rating']>=1)&(data['Rating']<=5)]
```

In [14]:
```
#display the cleaned data
data.head()
```

Out[14]:

	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	19000.0	10000	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	coloring book moana	art_and_design	3.9	967	14000.0	500000	Free	0.0	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	8700.0	5000000	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	sketch - draw & paint	art_and_design	4.5	215644	25000.0	50000000	Free	0.0	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	2800.0	100000	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up

EDA (Exploratory Data Analysis)

In [15]:
```
# Summary Stastics
data.describe()
```

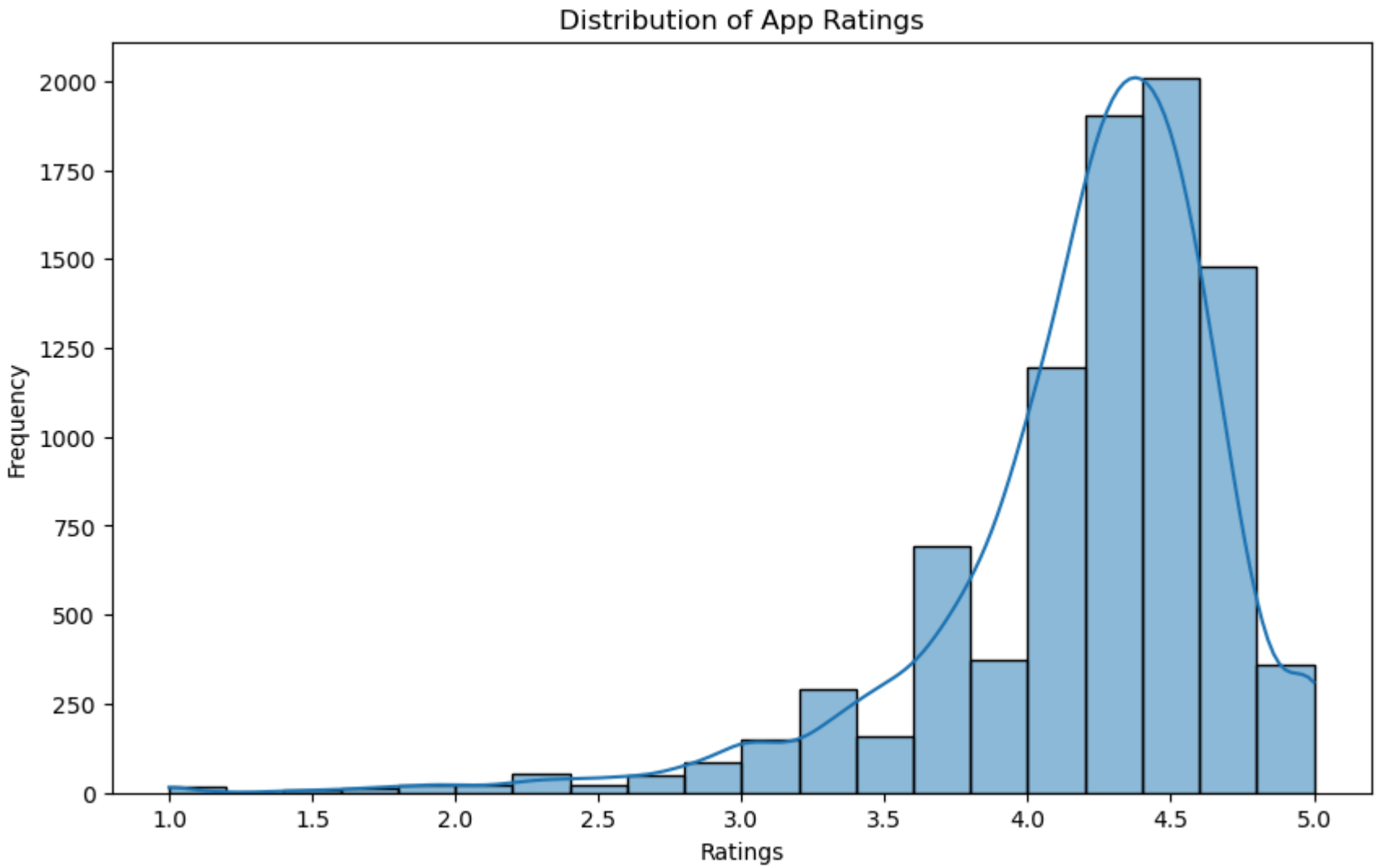
Out[15]:

	Rating	Reviews	Size	Installs	Price
count	8886.000000	8.886000e+03	8886.000000	8.886000e+03	8886.000000
mean	4.187959	4.730928e+05	22555.266019	1.650061e+07	0.963526
std	0.522428	2.906007e+06	21420.494024	8.640413e+07	16.194792
min	1.000000	1.000000e+00	8.500000	1.000000e+00	0.000000
25%	4.000000	1.640000e+02	6300.000000	1.000000e+04	0.000000
50%	4.300000	4.723000e+03	20000.000000	5.000000e+05	0.000000
75%	4.500000	7.131325e+04	27000.000000	5.000000e+06	0.000000
max	5.000000	7.815831e+07	100000.000000	1.000000e+09	400.000000

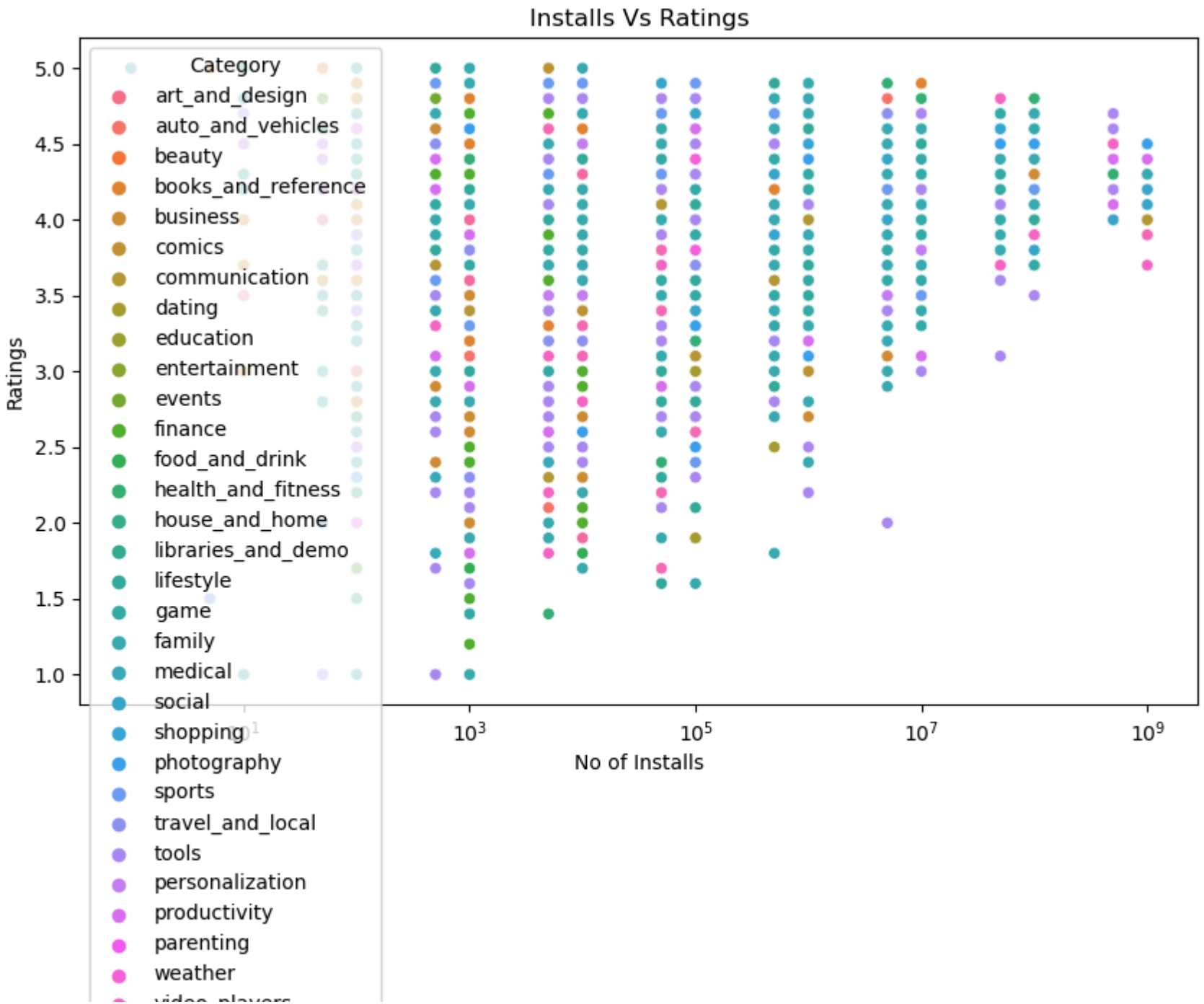
In [16]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

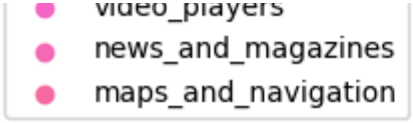
In [17]:
```
# Ratings Distribution
plt.figure(figsize=(10,6))
sns.histplot(data['Rating'].dropna(),bins=20, kde=True)
plt.title('Distribution of App Ratings')
plt.xlabel("Ratings")
plt.ylabel("Frequency")
plt.show()
```

```
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

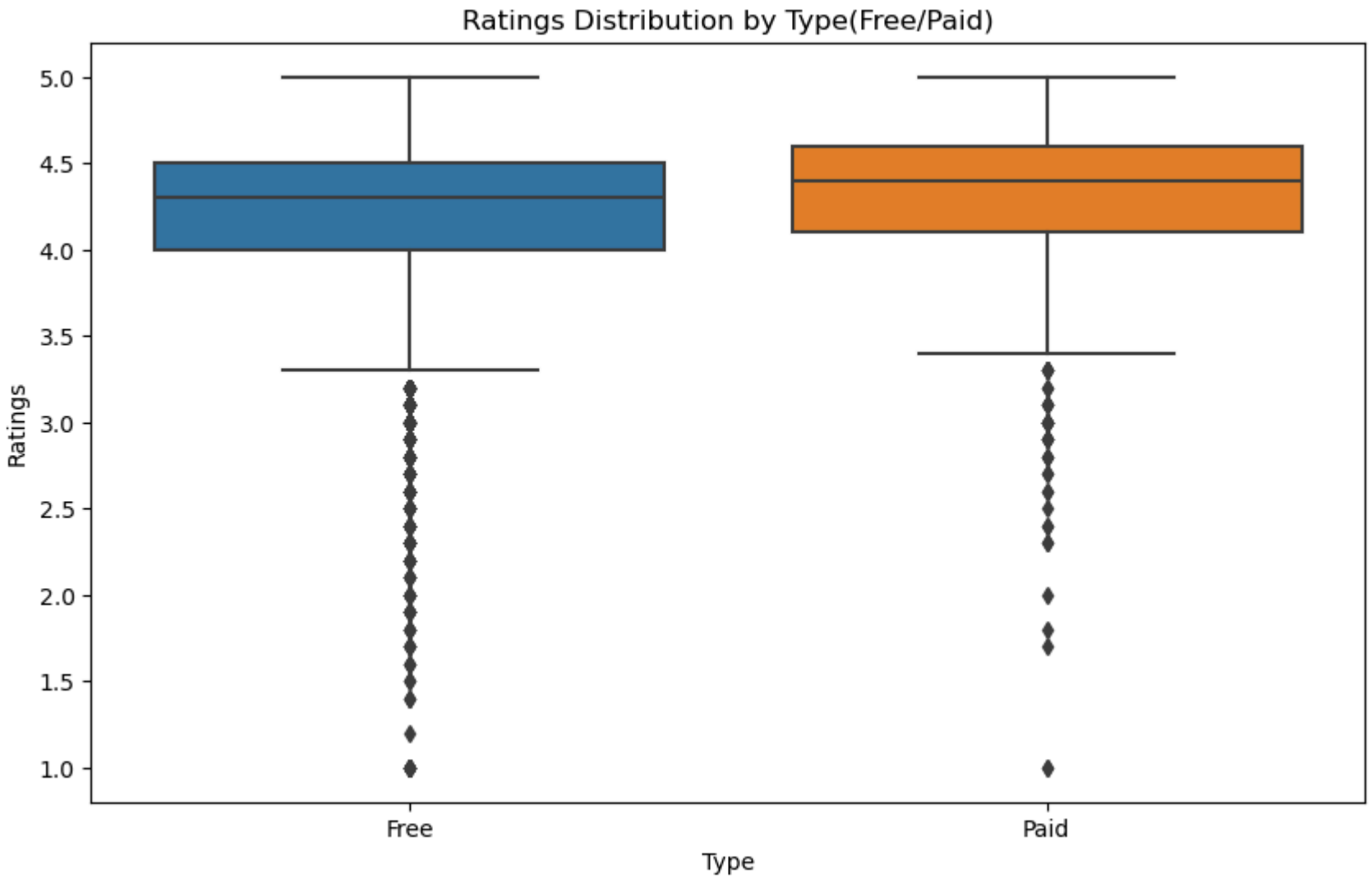


In [19]:
```
plt.figure(figsize=(10,6))
sns.scatterplot(data=data, x='Installs',y='Rating',hue='Category')
plt.title('Installs Vs Ratings')
plt.xlabel('No of Installs')
plt.ylabel('Ratings')
plt.xscale('log')
plt.show()
```

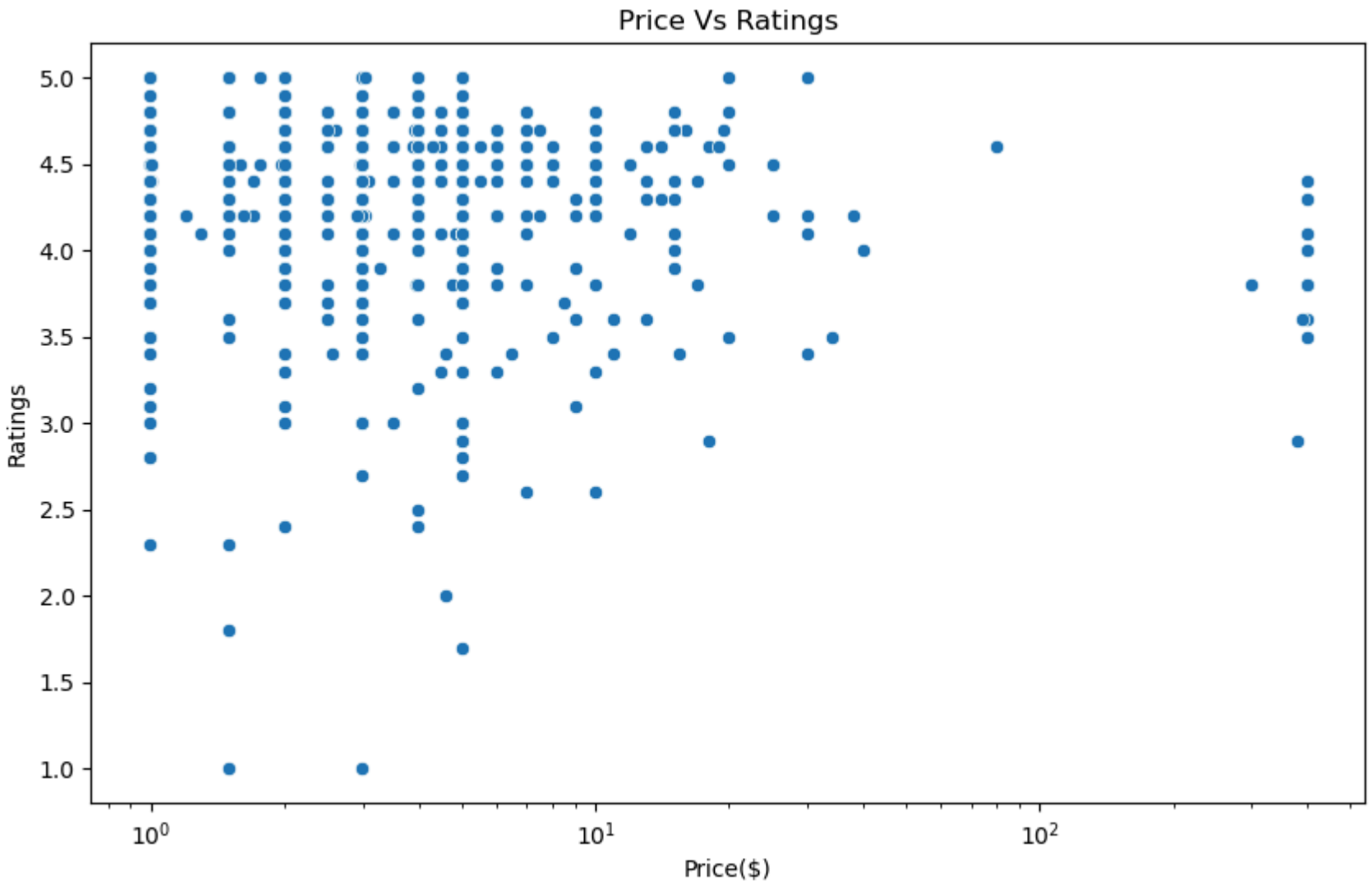

Installs Vs Ratings
Ratings
No of Installs
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
10^3
10^5
10^7
10^9
Category
art_and_design
auto_and_vehicles
beauty
books_and_reference
business
comics
communication
dating
education
entertainment
events
finance
food_and_drink
health_and_fitness
house_and_home
libraries_and_demo
lifestyle
game
family
medical
social
shopping
photography
sports
travel_and_local
tools
personalization
productivity
parenting
weather



In [21]:
```
# Box Plot
plt.figure(figsize=(10,6))
sns.boxplot(data=data,x='Type',y='Rating')
plt.title('Ratings Distribution by Type(Free/Paid)')
plt.xlabel('Type')
plt.ylabel('Ratings')
plt.show()
```

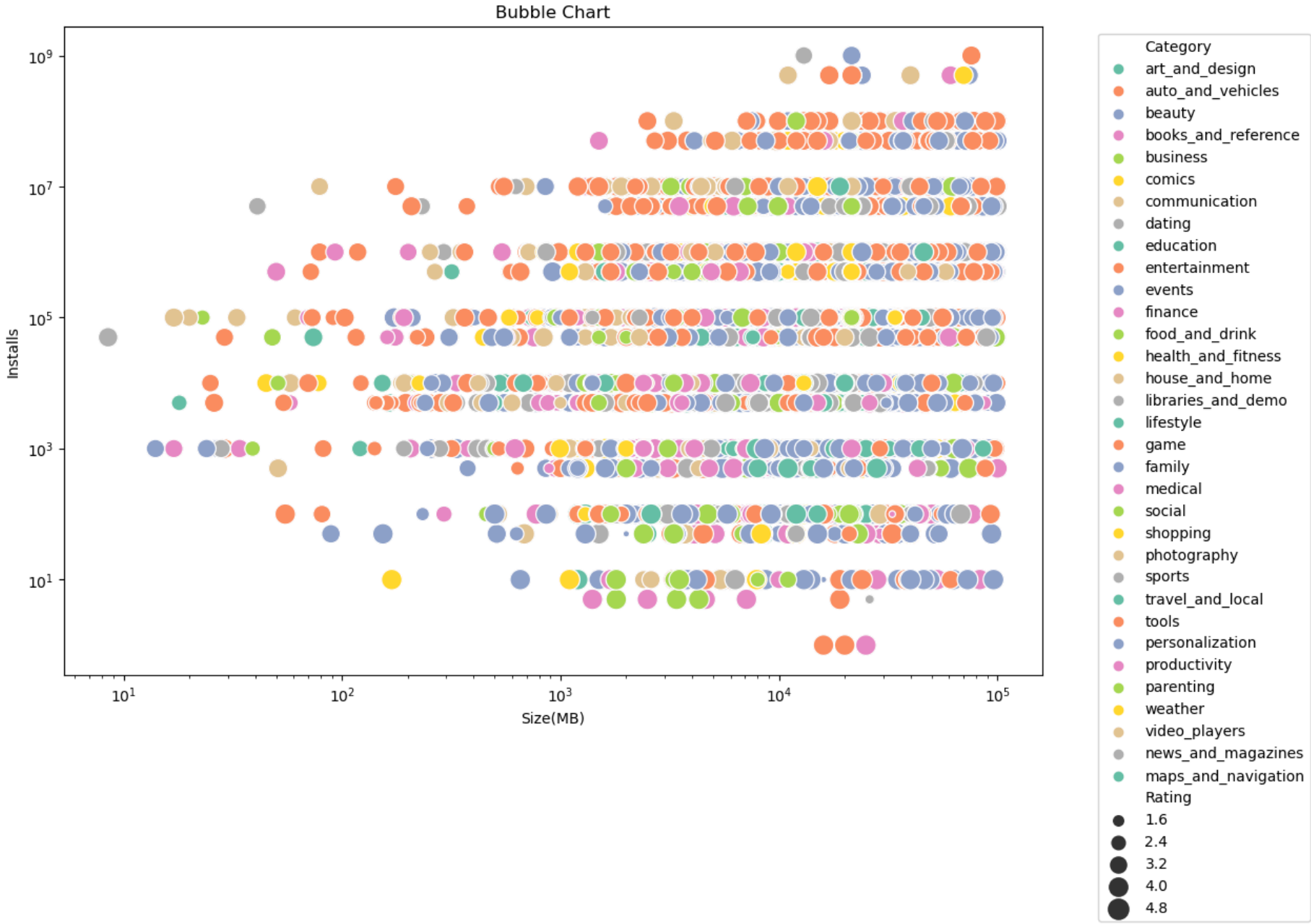


In [22]:
```
plt.figure(figsize=(10,6))
sns.scatterplot(data=data, x='Price',y='Rating')
plt.title('Price Vs Ratings')
plt.xlabel('Price($)')
plt.ylabel('Ratings')
plt.xscale('log')
plt.show()
```

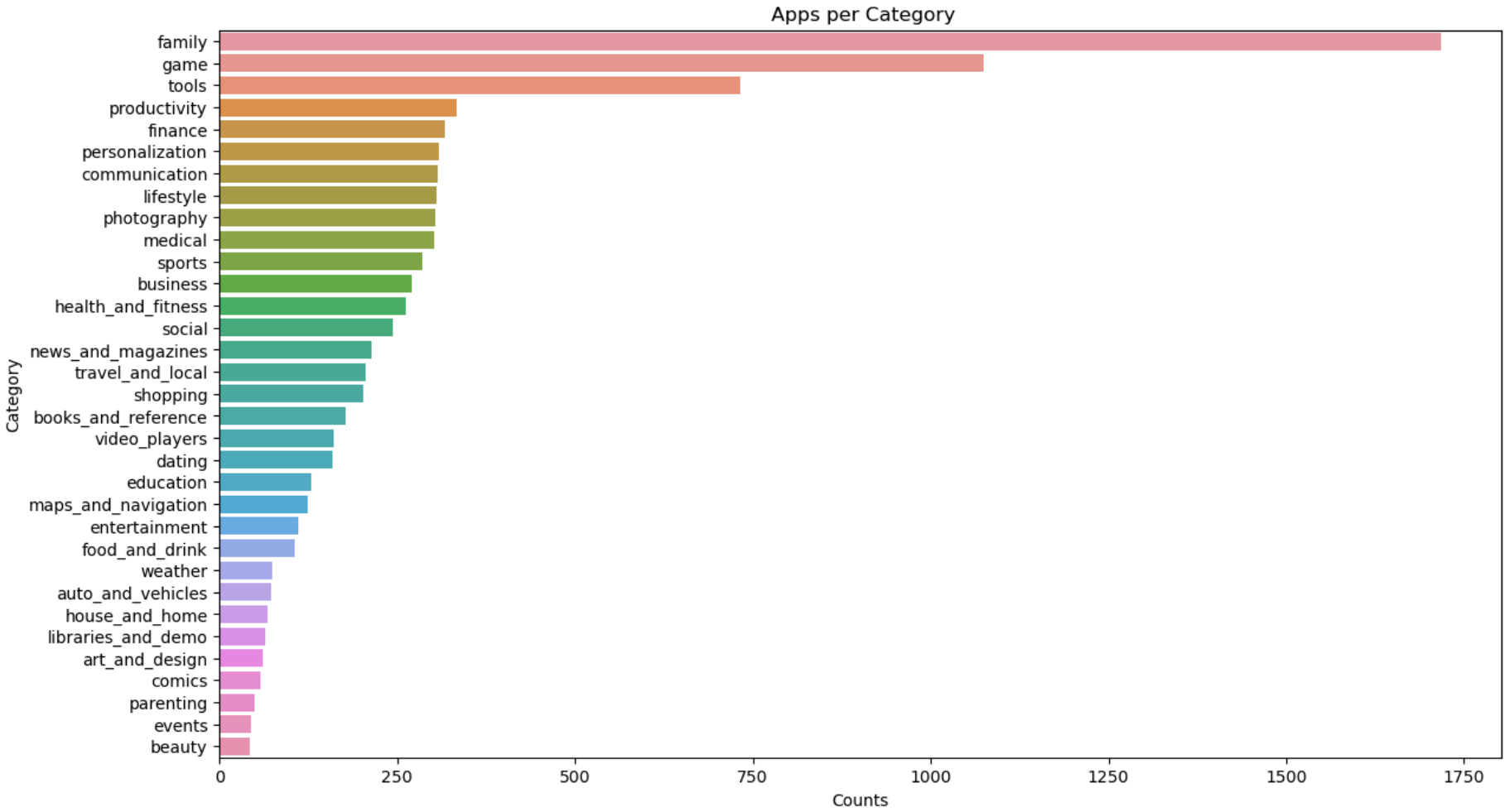


In [24]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=data, x='Size',y='Installs', size='Rating', hue='Category', sizes=(20,200),palette='Set2')
plt.title('Bubble Chart')
plt.xlabel('Size(MB)')
plt.ylabel('Installs')
plt.xscale('log')
plt.yscale('log')
```

```
plt.legend(bbox_to_anchor=(1.05,1), loc="upper left")
plt.show()
```

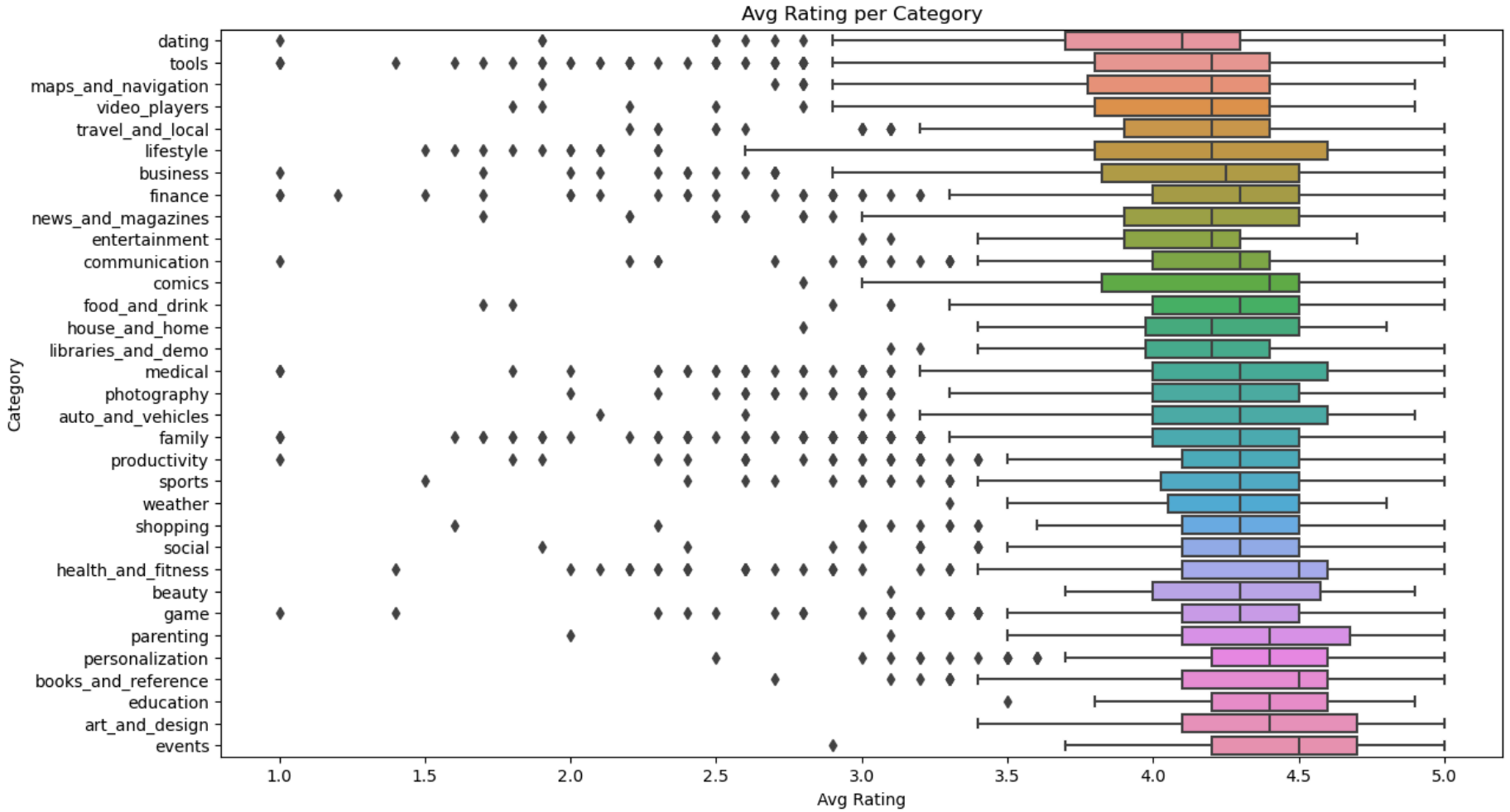

Bubble Chart
Installs
10^9
10^7
10^5
10^3
10^1
10^1
10^2
10^3
10^4
10^5
Size(MB)
Category
art_and_design
auto_and_vehicles
beauty
books_and_reference
business
comics
communication
dating
education
entertainment
events
finance
food_and_drink
health_and_fitness
house_and_home
libraries_and_demo
lifestyle
game
family
medical
social
shopping
photography
sports
travel_and_local
tools
personalization
productivity
parenting
weather
video_players
news_and_magazines
maps_and_navigation
Rating
1.6
2.4
3.2
4.0
4.8

In [32]:
```
# Number of Apps per CATEGORY
plt.figure(figsize=(14,8))
sns.countplot(y="Category", data=data,order=data['Category'].value_counts().index)
plt.title(' Apps per Category')
plt.xlabel("Counts")
plt.ylabel("Category")
plt.show()
```



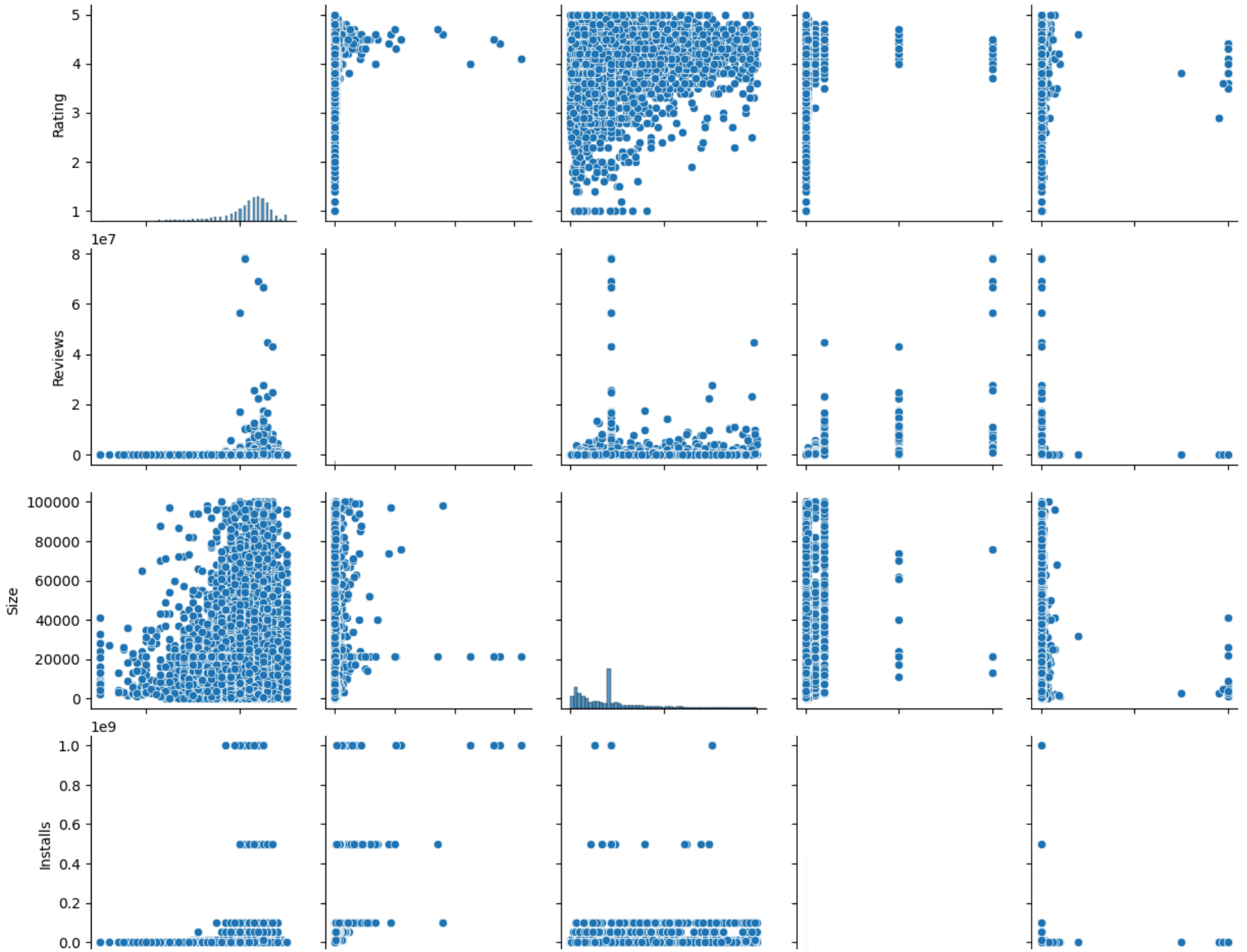
In [34]:
```
#Avg Rating per Category
plt.figure(figsize=(14,8))
```

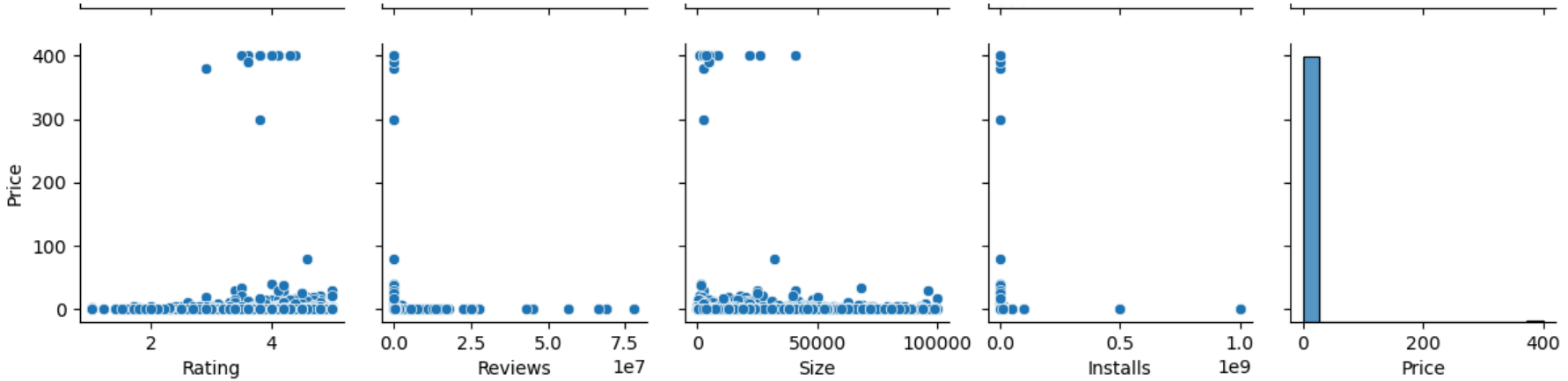
```
sns.boxplot(x='Rating',y='Category', data=data,order=data.groupby("Category")['Rating'].mean().sort_values().index)
plt.title('Avg Rating per Category')
plt.xlabel("Avg Rating")
plt.ylabel("Category")
plt.show()
```



In [37]:
```
# Pair Plot for slecetd features
sns.pairplot(data[['Rating','Reviews','Size','Installs','Price']].dropna())
plt.show()
```

```
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```


Rating
Reviews
1e7
Size
Installs
1e9



Feature Engineering

In [38]:
```
import numpy as np
```

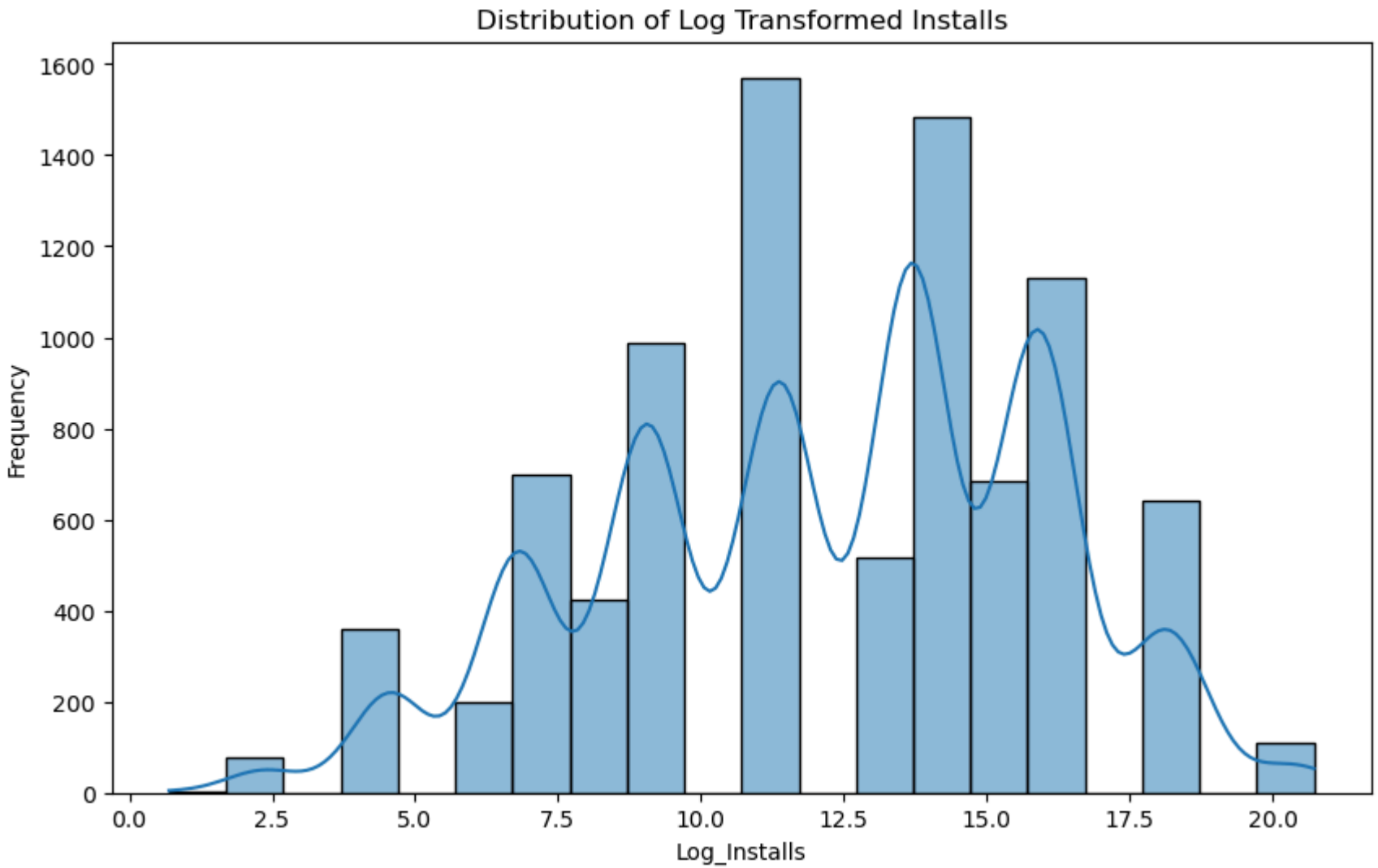
In [41]:
```
# Create a log-transformed 'Installs' feature
data['Log_Installs'] = np.log1p(data['Installs'])

# Create a categorical feature for 'Price'
data['Price_Category'] = pd.cut(data['Price'], bins=[0, 1, 5, 10, 50, 100, 400], labels=['Free', 'Cheap', 'Moderate', 'Expensi
```

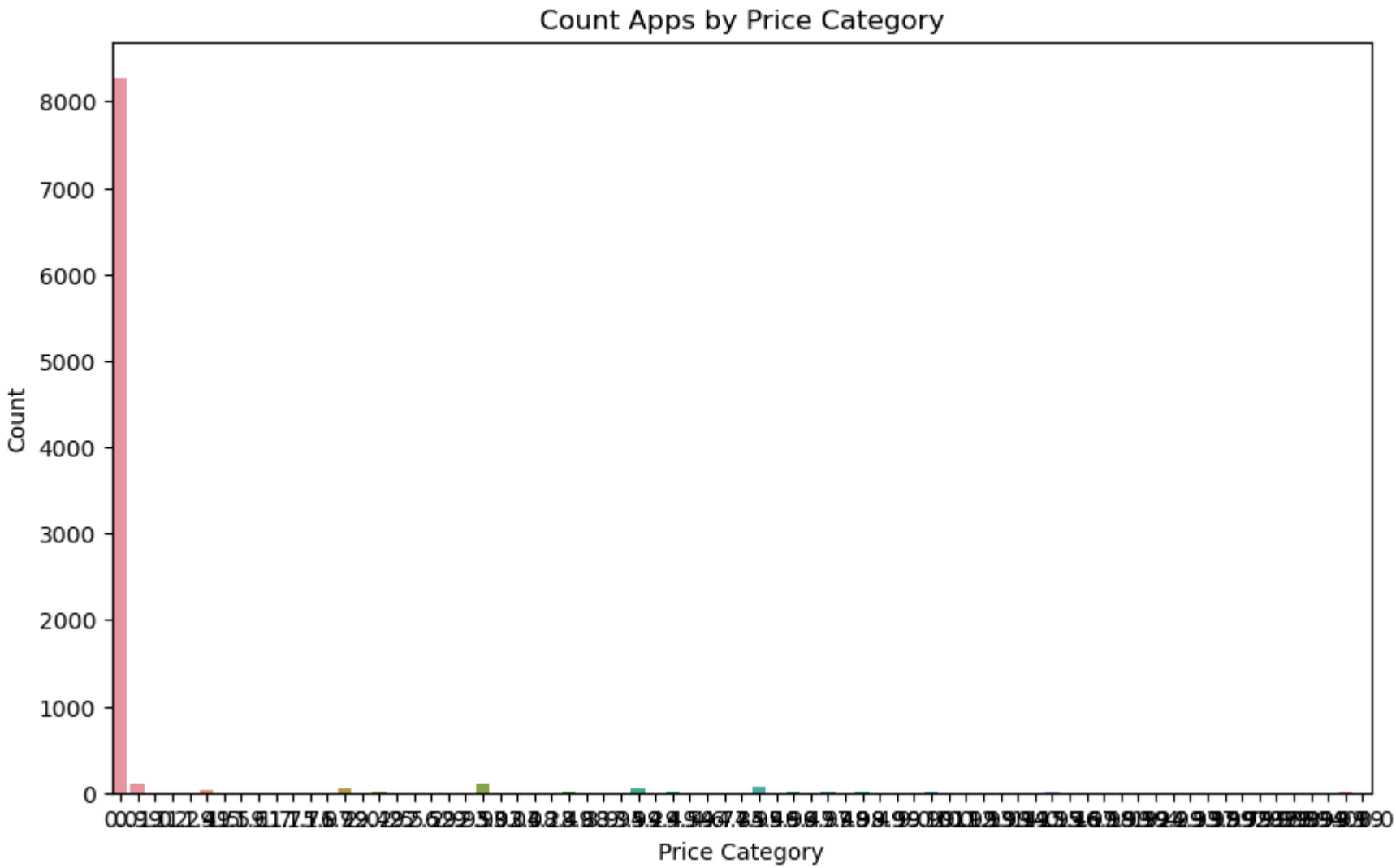
Visulaization of New Features

In [42]:
```
plt.figure(figsize=(10,6))
sns.histplot(data['Log_Installs'].dropna(),bins=20,kde=True)
plt.title('Distribution of Log Transformed Installs')
plt.xlabel("Log_Installs")
plt.ylabel("Frequency")
plt.show()
```

```
D:\New folder\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [43]:
```
plt.figure(figsize=(10,6))
sns.countplot(x='Price', data=data)
plt.title('Count Apps by Price Category')
plt.xlabel("Price Category")
plt.ylabel("Count")
plt.show()
```



In [44]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in d:\new folder\lib\site-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in d:\new folder\lib\site-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in d:\new folder\lib\site-packages (from wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in d:\new folder\lib\site-packages (from wordcloud) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (23.1)
Requirement already satisfied: pyparsing>=2.3.1 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in d:\new folder\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in d:\new folder\lib\site-packages (from python-dateutil>=2.7->matplotlib->wordcloud)
(1.16.0)
```

In [45]:
```
from wordcloud import WordCloud
```

In [46]:
```
# Word Cloud for App Description
if 'App' in data.columns:
    text="".join(app for app in data['App'].dropna())

wordcloud= WordCloud(background_color='White').generate(text)
plt.figure(figsize=(10,8))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of App Description')
plt.show()
```

@Lipika Sharma