

```
In [2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

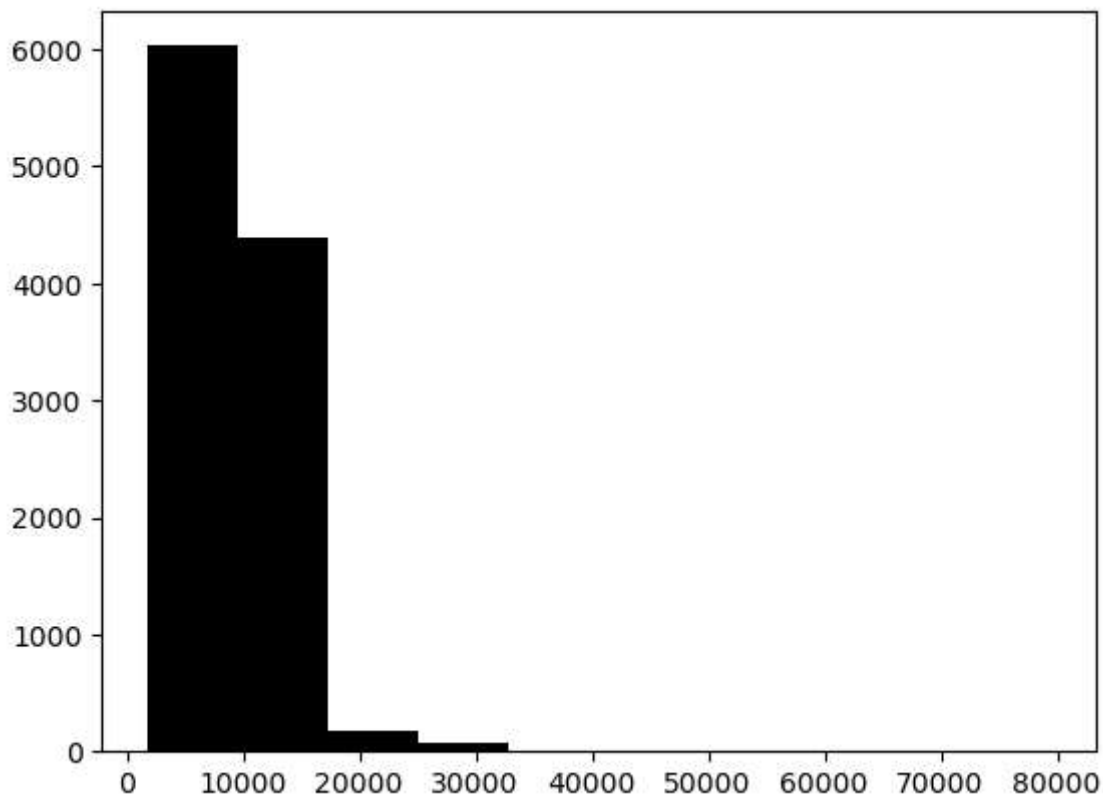
```
In [3]: df=pd.read_excel("flight_price.xlsx")
```

```
In [8]: #Q1
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Airline                10683 non-null  object
1   Date_of_Journey        10683 non-null  object
2   Source                 10683 non-null  object
3   Destination            10683 non-null  object
4   Route                 10682 non-null  object
5   Dep_Time              10683 non-null  object
6   Arrival_Time          10683 non-null  object
7   Duration               10683 non-null  object
8   Total_Stops            10682 non-null  object
9   Additional_Info        10683 non-null  object
10  Price                 10683 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

```
In [15]: #Q2
plt.hist(data=df,x='Price',color='Black')
```

```
Out[15]: (array([6.029e+03, 4.390e+03, 1.820e+02, 6.600e+01, 7.000e+00, 1.000e+00,
5.000e+00, 2.000e+00, 0.000e+00, 1.000e+00]),
array([ 1759. , 9534.3, 17309.6, 25084.9, 32860.2, 40635.5, 48410.8,
56186.1, 63961.4, 71736.7, 79512. ]),
<BarContainer object of 10 artists>)
```



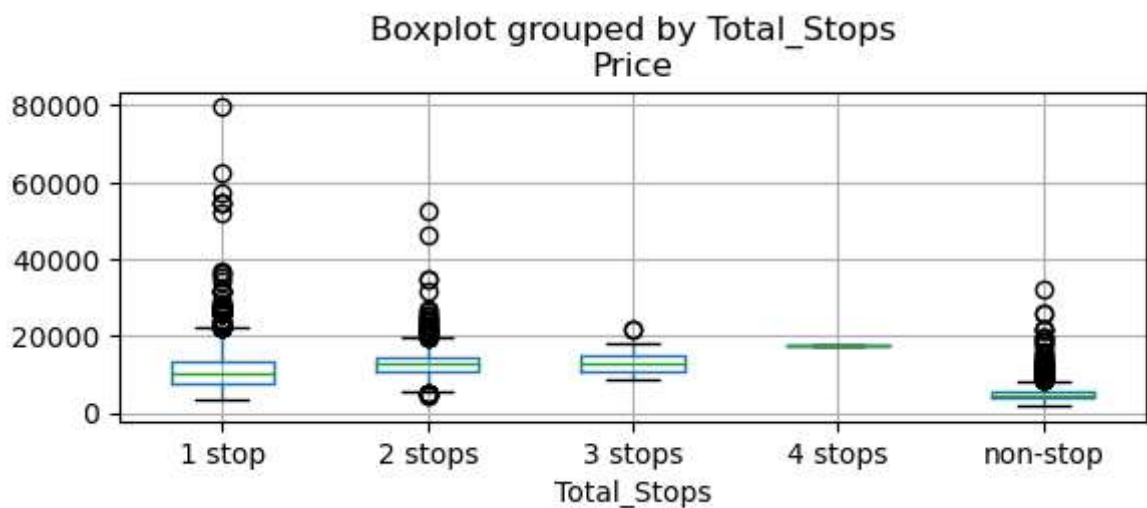
```
In [16]: #Q3
df.describe().T
```

```
Out[16]:
```

	count	mean	std	min	25%	50%	75%	max
Price	10683.0	9087.064121	4611.359167	1759.0	5277.0	8372.0	12373.0	79512.0

```
In [18]: #Q4
df.boxplot(column=['Price'],by='Total_Stops',layout=(2,1))
```

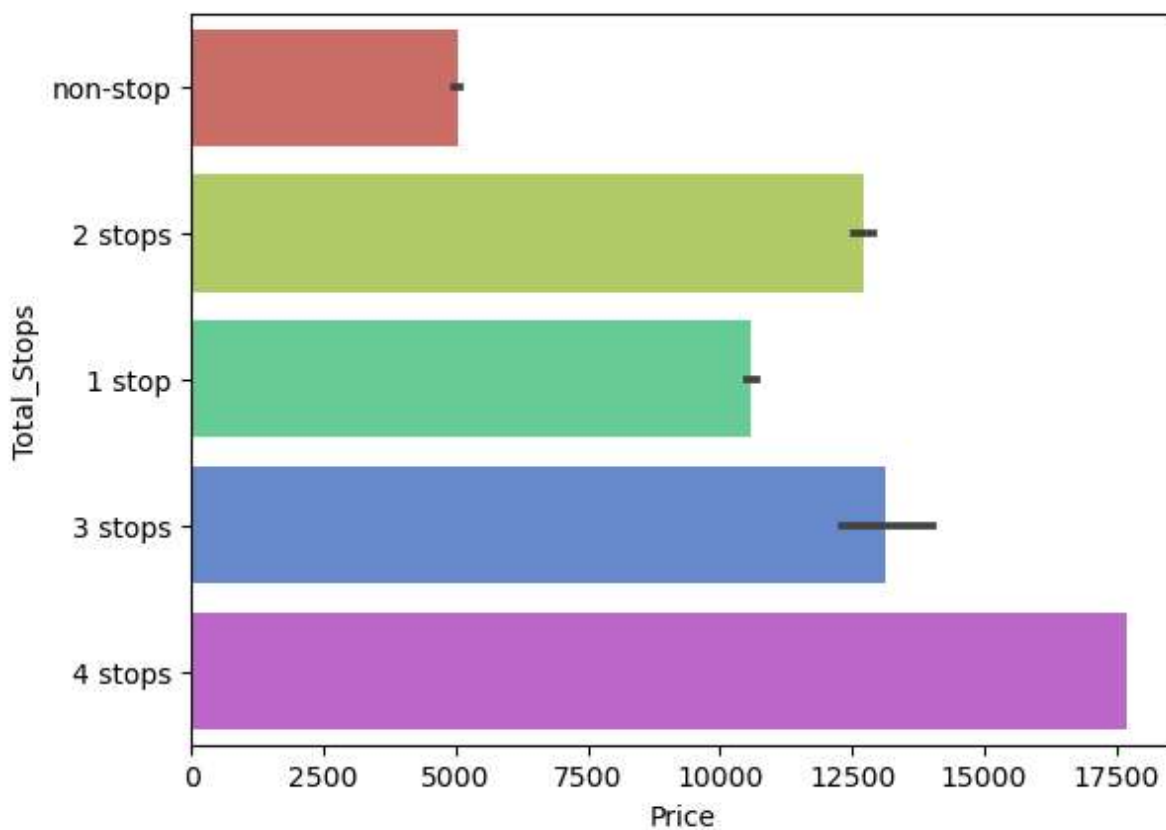
```
Out[18]: array([<AxesSubplot: title={'center': 'Price'}, xlabel='Total_Stops'>,
<AxesSubplot: >], dtype=object)
```



Q5 Yes above Plot has a Outliers,when we compare price with Total\_Stops.Outliers can significantly increase the variance ,making it appear that the data is more spread out than it really is.

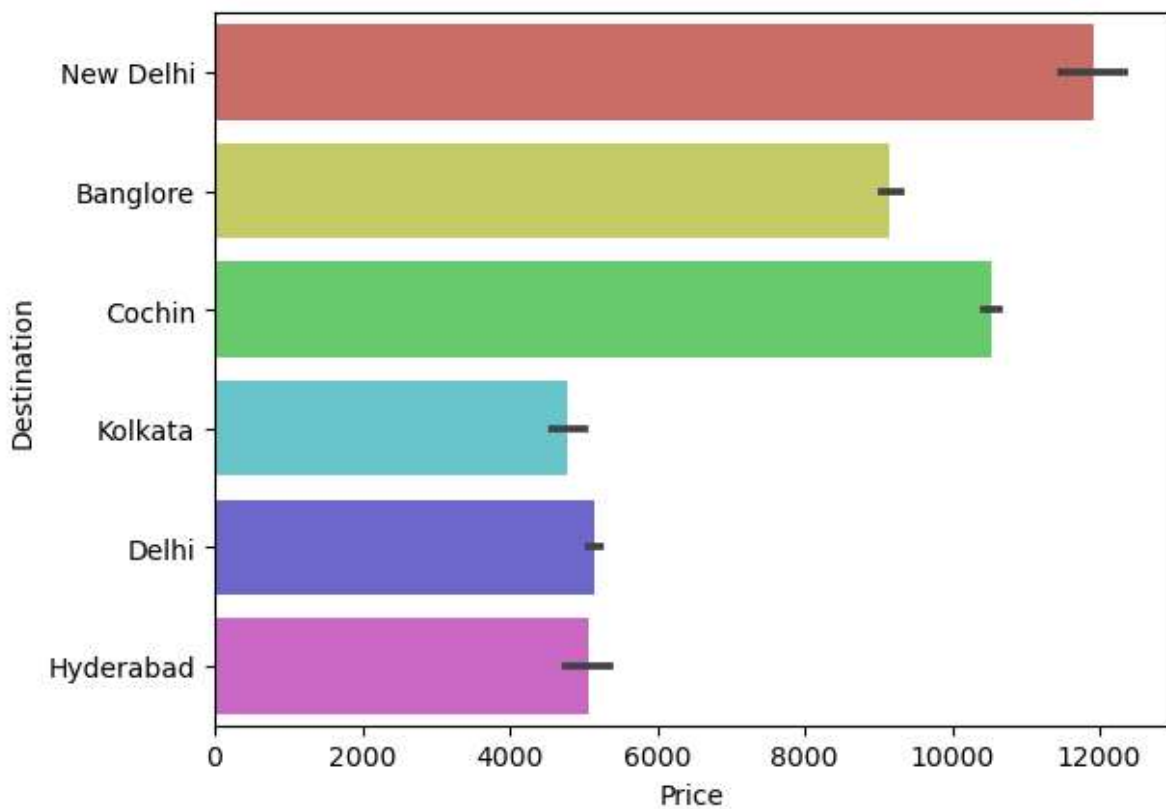
```
In [22]: #Q6
sns.barplot(x=df['Price'],y=df['Total_Stops'],palette='hls')
```

```
Out[22]: <AxesSubplot: xlabel='Price', ylabel='Total_Stops'>
```



```
In [24]: #Q7
sns.barplot(x=df['Price'],y=df['Destination'],palette='hls')
```

```
Out[24]: <AxesSubplot: xlabel='Price', ylabel='Destination'>
```



In [25]: `#Q8`  
`df.head(2)`

Out[25]:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	3 stops

In [4]: `#Q9`

```
df1=pd.read_csv('https://raw.githubusercontent.com/krishnaik06/playstore-Dataset/main/df1.csv')
df1.head(2)
```

Out[4]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Ge
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & D
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Design;Pre

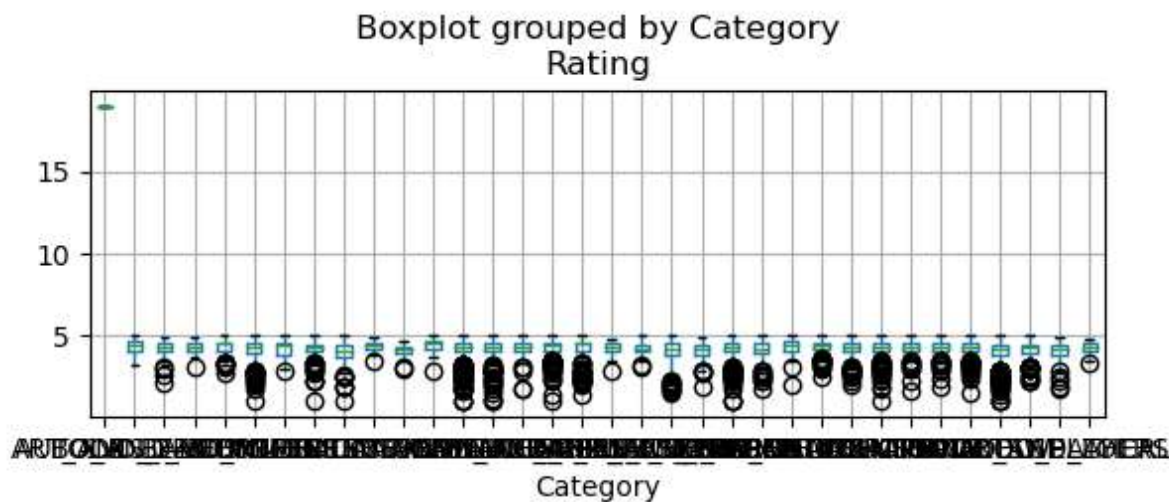
In [5]: df1.info()

```
<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 [4]: #Q10

```
df1.boxplot(column=['Rating'],by='Category',layout=(2,1))
```

```
Out[4]: array([<AxesSubplot: title={'center': 'Rating'}, xlabel='Category'>,
               <AxesSubplot: >], dtype=object)
```

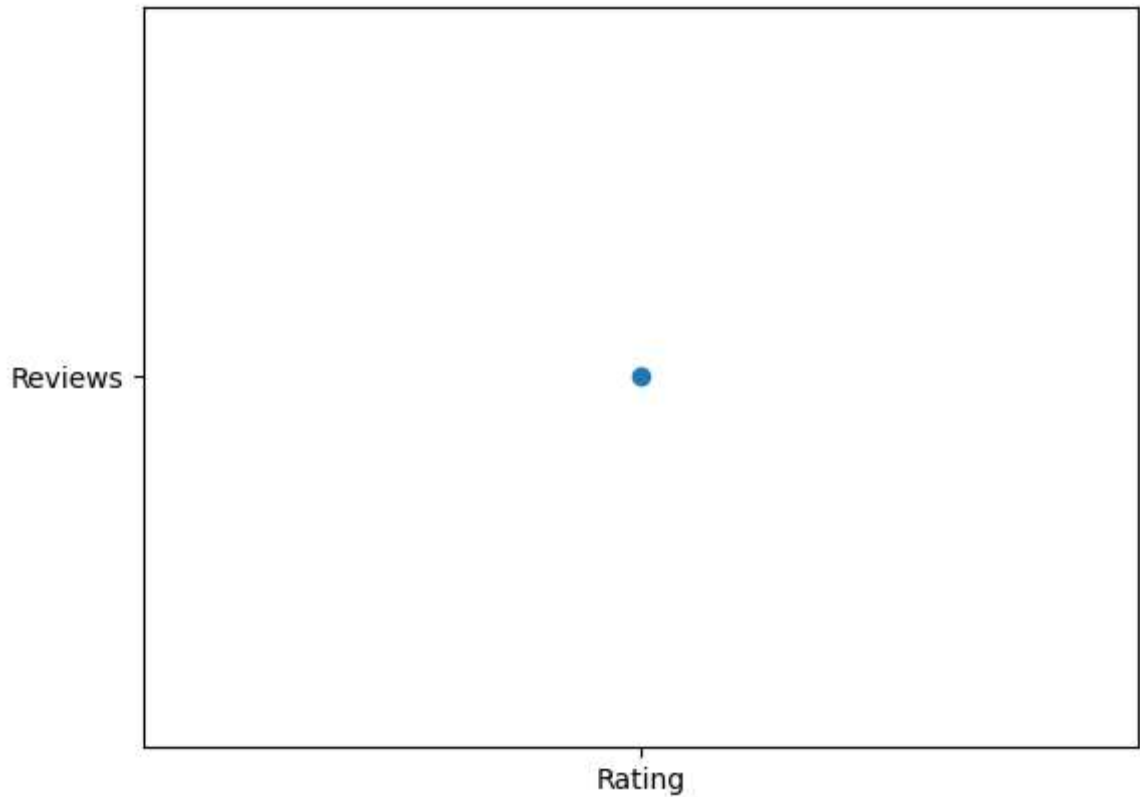


```
In [5]: #Q11
df1.isnull().sum()
```

```
Out[5]: 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
```

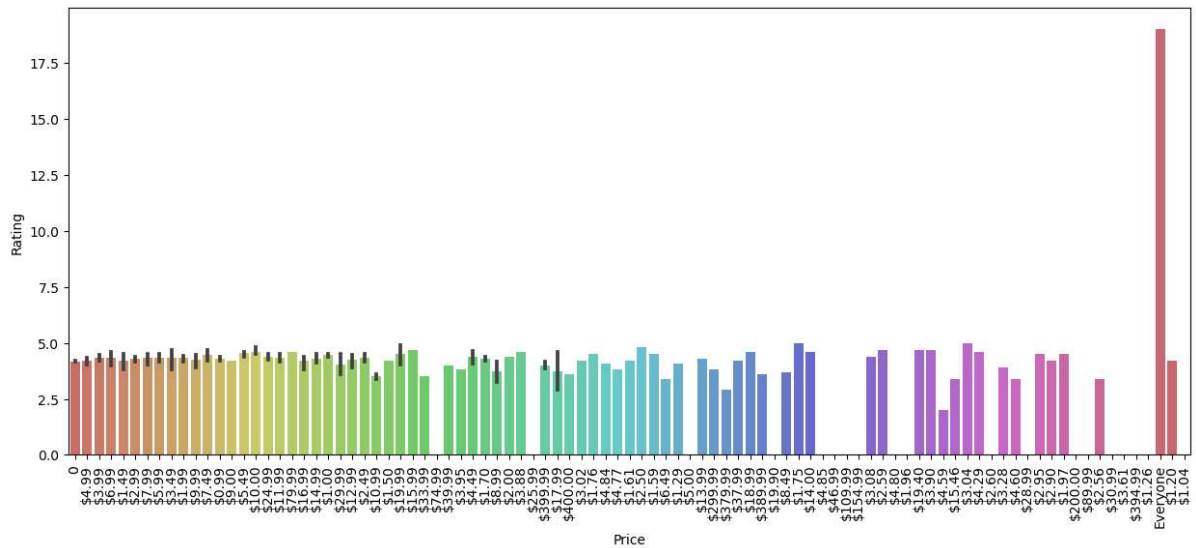
```
In [14]: #Q12
plt.scatter(x='Rating',y='Reviews')
```

```
Out[14]: <matplotlib.collections.PathCollection at 0x7f0209ff5870>
```



```
In [16]: #Q13
plt.figure(figsize=(15,6))
sns.barplot(x='Price',y='Rating',data=df1,palette='hls')
#plt.title('Top 10 App categories')
plt.xticks(rotation=90)
plt.show()
```

```
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
/opt/conda/lib/python3.10/site-packages/seaborn/algorithms.py:98: RuntimeWarning:
Mean of empty slice
  boot_dist.append(f(*sample, **func_kwargs))
```



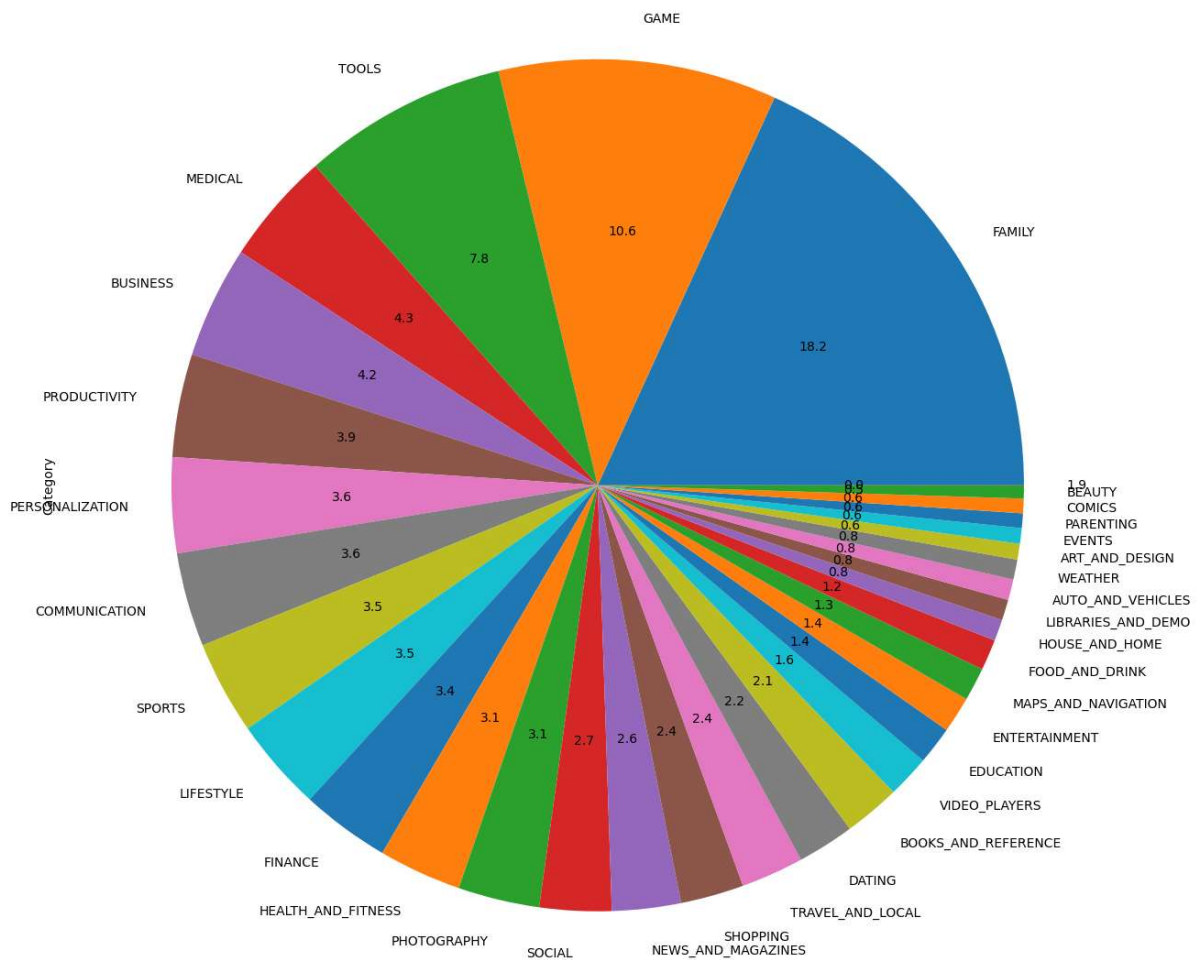
```
In [19]: #Q14
df1['Installs'].describe()
```

```
Out[19]: count      10841
         unique        22
         top    1,000,000+
         freq       1579
         Name: Installs, dtype: object
```

```
In [20]: #Q15
df1['Category'].value_counts().plot.pie(y=df1['Category'],figsize=(15,16),autopct='')
```

```
Out[20]: <AxesSubplot: ylabel='Category'>
```





In [26]: #Q16

```
category = pd.DataFrame(df1['Category'].value_counts()) #Dataframe of apps o
category.rename(columns = {'Category':'Count'},inplace=True)
```

In [27]: category

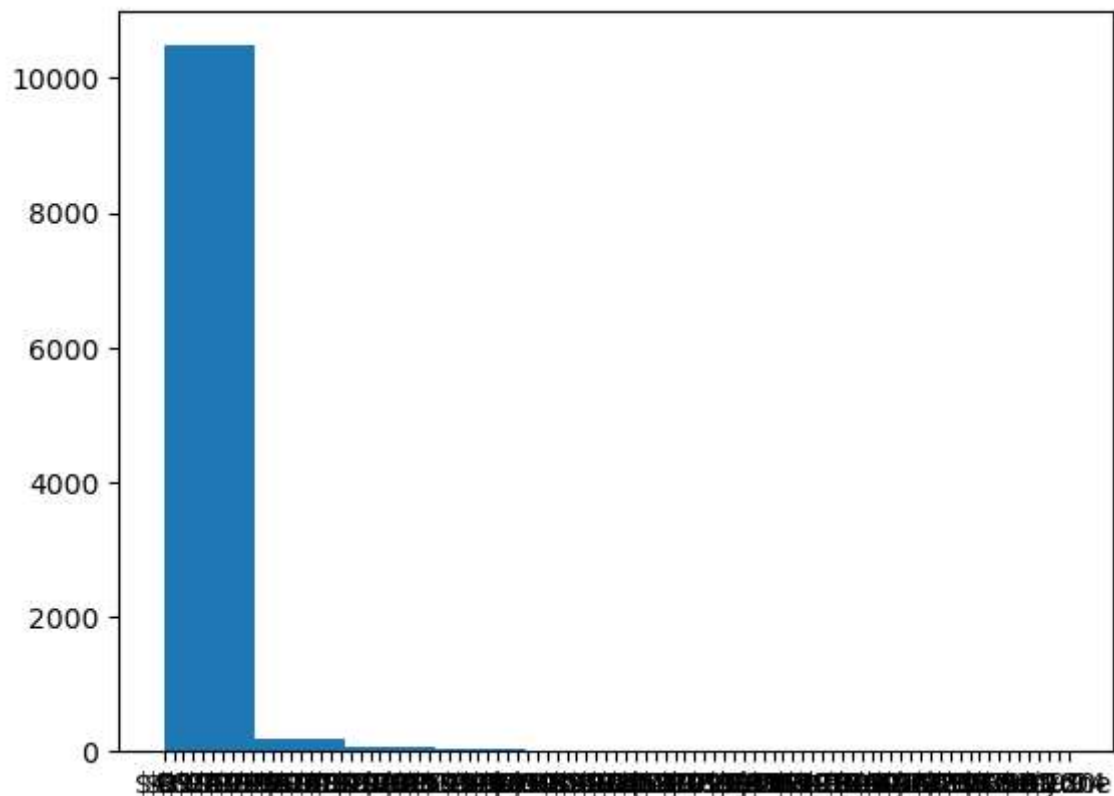
Out[27]:

	Count
FAMILY	1972
GAME	1144
TOOLS	843
MEDICAL	463
BUSINESS	460
PRODUCTIVITY	424
PERSONALIZATION	392
COMMUNICATION	387
SPORTS	384
LIFESTYLE	382
FINANCE	366
HEALTH_AND_FITNESS	341
PHOTOGRAPHY	335
SOCIAL	295
NEWS_AND_MAGAZINES	283
SHOPPING	260
TRAVEL_AND_LOCAL	258
DATING	234
BOOKS_AND_REFERENCE	231
VIDEO_PLAYERS	175
EDUCATION	156
ENTERTAINMENT	149
MAPS_AND_NAVIGATION	137
FOOD_AND_DRINK	127
HOUSE_AND_HOME	88
LIBRARIES_AND_DEMO	85
AUTO_AND_VEHICLES	85
WEATHER	82
ART_AND_DESIGN	65
EVENTS	64
PARENTING	60
COMICS	60
BEAUTY	53

Count	
1.9	1

```
In [14]: #Q17  
plt.hist(x='Price',data=df1)
```

```
Out[14]: (array([1.048e+04, 1.960e+02, 6.400e+01, 2.700e+01, 2.100e+01, 1.500e+01,  
                1.000e+01, 9.000e+00, 9.000e+00, 1.000e+01]),  
array([ 0. ,  9.2, 18.4, 27.6, 36.8, 46. , 55.2, 64.4, 73.6, 82.8, 92. ]),  
<BarContainer object of 10 artists>)
```



```
In [ ]:
```