Simplilearn GooglePlay store Project

March 3, 2022

[1]:
```
%config Completer.use_jedi = True
```

1 Load data files using Pandas

[3]:
```
import pandas as pd
```

[4]:
```
dataset=pd.read_csv('googleplaystore.csv')
```

[5]:
```
dataset.head()
```

[5]:

	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	19M	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	14M	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	8.7M	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	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	

```
3    4.2 and up
4    4.4 and up
```

2 Check for null values

[7]:
```
dataset.isnull().sum()
```

[7]:
```
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
```

3 Drop records with any null values

[9]:
```
dataset=dataset.dropna()
dataset=dataset.reset_index(drop=True)
```

[10]:
```
dataset.isnull().sum()
```

[10]:
```
App             0
Category        0
Rating          0
Reviews         0
Size            0
Installs        0
Type            0
Price           0
Content Rating  0
Genres          0
Last Updated    0
Current Ver     0
Android Ver     0
dtype: int64
```

4 Fixing variables

[12]:
```
dataset["Size"].unique()
```

[12]:
```
array(['19M', '14M', '8.7M', '25M', '2.8M', '5.6M', '29M', '33M', '3.1M',
       '28M', '12M', '20M', '21M', '37M', '5.5M', '17M', '39M', '31M',
       '4.2M', '23M', '6.0M', '6.1M', '4.6M', '9.2M', '5.2M', '11M',
       '24M', 'Varies with device', '9.4M', '15M', '10M', '1.2M', '26M',
       '8.0M', '7.9M', '56M', '57M', '35M', '54M', '201k', '3.6M', '5.7M',
       '8.6M', '2.4M', '27M', '2.7M', '2.5M', '7.0M', '16M', '3.4M',
       '8.9M', '3.9M', '2.9M', '38M', '32M', '5.4M', '18M', '1.1M',
       '2.2M', '4.5M', '9.8M', '52M', '9.0M', '6.7M', '30M', '2.6M',
       '7.1M', '22M', '6.4M', '3.2M', '8.2M', '4.9M', '9.5M', '5.0M',
       '5.9M', '13M', '73M', '6.8M', '3.5M', '4.0M', '2.3M', '2.1M',
       '42M', '9.1M', '55M', '23k', '7.3M', '6.5M', '1.5M', '7.5M', '51M',
       '41M', '48M', '8.5M', '46M', '8.3M', '4.3M', '4.7M', '3.3M', '40M',
       '7.8M', '8.8M', '6.6M', '5.1M', '61M', '66M', '79k', '8.4M',
       '3.7M', '118k', '44M', '695k', '1.6M', '6.2M', '53M', '1.4M',
       '3.0M', '7.2M', '5.8M', '3.8M', '9.6M', '45M', '63M', '49M', '77M',
       '4.4M', '70M', '9.3M', '8.1M', '36M', '6.9M', '7.4M', '84M', '97M',
       '2.0M', '1.9M', '1.8M', '5.3M', '47M', '556k', '526k', '76M',
       '7.6M', '59M', '9.7M', '78M', '72M', '43M', '7.7M', '6.3M', '334k',
       '93M', '65M', '79M', '100M', '58M', '50M', '68M', '64M', '34M',
       '67M', '60M', '94M', '9.9M', '232k', '99M', '624k', '95M', '8.5k',
       '41k', '292k', '80M', '1.7M', '10.0M', '74M', '62M', '69M', '75M',
       '98M', '85M', '82M', '96M', '87M', '71M', '86M', '91M', '81M',
       '92M', '83M', '88M', '704k', '862k', '899k', '378k', '4.8M',
       '266k', '375k', '1.3M', '975k', '980k', '4.1M', '89M', '696k',
       '544k', '525k', '920k', '779k', '853k', '720k', '713k', '772k',
       '318k', '58k', '241k', '196k', '857k', '51k', '953k', '865k',
       '251k', '930k', '540k', '313k', '746k', '203k', '26k', '314k',
       '239k', '371k', '220k', '730k', '756k', '91k', '293k', '17k',
       '74k', '14k', '317k', '78k', '924k', '818k', '81k', '939k', '169k',
       '45k', '965k', '90M', '545k', '61k', '283k', '655k', '714k', '93k',
       '872k', '121k', '322k', '976k', '206k', '954k', '444k', '717k',
       '210k', '609k', '308k', '306k', '175k', '350k', '383k', '454k',
       '1.0M', '70k', '812k', '442k', '842k', '417k', '412k', '459k',
       '478k', '335k', '782k', '721k', '430k', '429k', '192k', '460k',
       '728k', '496k', '816k', '414k', '506k', '887k', '613k', '778k',
       '683k', '592k', '186k', '840k', '647k', '373k', '437k', '598k',
       '716k', '585k', '982k', '219k', '55k', '323k', '691k', '511k',
       '951k', '963k', '25k', '554k', '351k', '27k', '82k', '208k',
       '551k', '29k', '103k', '116k', '153k', '209k', '499k', '173k',
       '597k', '809k', '122k', '411k', '400k', '801k', '787k', '50k',
       '643k', '986k', '516k', '837k', '780k', '20k', '498k', '600k',
       '656k', '221k', '228k', '176k', '34k', '259k', '164k', '458k',
       '629k', '28k', '288k', '775k', '785k', '636k', '916k', '994k',
```

```
       '309k', '485k', '914k', '903k', '608k', '500k', '54k', '562k',
       '847k', '948k', '811k', '270k', '48k', '523k', '784k', '280k',
       '24k', '892k', '154k', '18k', '33k', '860k', '364k', '387k',
       '626k', '161k', '879k', '39k', '170k', '141k', '160k', '144k',
       '143k', '190k', '376k', '193k', '473k', '246k', '73k', '253k',
       '957k', '420k', '72k', '404k', '470k', '226k', '240k', '89k',
       '234k', '257k', '861k', '467k', '676k', '552k', '582k', '619k'],
      dtype=object)
```

[13]:
```
def mb_to_kb(b):
  if b.endswith("M"):
    return float(b[:-1])*1000
  elif b.endswith("k"):
    return float(b[:-1])
  else:
    return b
```

[14]:
```
dataset["Size"]=dataset["Size"].apply(lambda b:mb_to_kb(b))
```

Need to deal with varies with device

[16]:
```
dataset[dataset["Size"]=="Varies with device"]
```

[16]:

	App	Category	Rating	Reviews	Size	Installs	Type	Price
35	Floor Plan Creator	ART_AND_DESIGN	4.1	36639	Varies with device	5,000,000+	Free	0
40	Textgram - write on photos	ART_AND_DESIGN	4.4	295221	Varies with device	10,000,000+	Free	0
50	Used Cars and Trucks for Sale	AUTO_AND_VEHICLES	4.6	17057	Varies with device	1,000,000+	Free	0
65	Ulysse Speedometer	AUTO_AND_VEHICLES	4.3	40211	Varies with device	5,000,000+	Free	0
66	REPUVE	AUTO_AND_VEHICLES	3.9	356	Varies with device	100,000+	Free	0
…	…	…	…	…	…	…	…	…
9267	My Earthquake Alerts - US & Worldwide Earthquakes	WEATHER	4.4	3471	Varies with device	100,000+	Free	0
9279	Posta App	MAPS_AND_NAVIGATION	3.6	8	Varies with device	1,000+	Free	0
9307	Chat For Strangers - Video Chat	SOCIAL	3.4	622	Varies with device	100,000+	Free	0
9348	Frim: get new friends on local chat rooms	SOCIAL	4.0	88486	Varies with device	5,000,000+	Free	0
9358	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE						

```
9358     4.5     114  Varies with device       1,000+  Free     0

     Content Rating             Genres        Last Updated  \
35         Everyone       Art & Design       July 14, 2018
40         Everyone       Art & Design       July 30, 2018
50         Everyone    Auto & Vehicles       July 30, 2018
65         Everyone    Auto & Vehicles       July 30, 2018
66         Everyone    Auto & Vehicles        May 25, 2018
…             …                …                …
9267       Everyone            Weather       July 24, 2018
9279       Everyone  Maps & Navigation  September 27, 2017
9307     Mature 17+             Social        May 23, 2018
9348     Mature 17+             Social      March 23, 2018
9358     Mature 17+  Books & Reference    January 19, 2015

              Current Ver         Android Ver
35    Varies with device        2.3.3 and up
40    Varies with device  Varies with device
50    Varies with device  Varies with device
65    Varies with device  Varies with device
66    Varies with device  Varies with device
…                    …                 …
9267  Varies with device  Varies with device
9279  Varies with device          4.4 and up
9307  Varies with device  Varies with device
9348  Varies with device  Varies with device
9358  Varies with device  Varies with device

[1637 rows x 13 columns]
```

[17]:
```
rows=dataset[dataset["Size"]=="Varies with device"].index
dataset.drop(rows,inplace=True)
```

Convert reviews to numeric

[19]:
```
dataset['Reviews']=dataset["Reviews"].astype(int)
```

Change installs

[21]:
```
dataset["Installs"].value_counts()
```

[21]:
```
1,000,000+        1301
100,000+          1037
10,000+            968
10,000,000+        825
1,000+             689
5,000,000+         535
```

```
500,000+            490
50,000+             436
5,000+              419
100+                303
100,000,000+        201
500+                197
50,000,000+         147
10+                  67
50+                  56
500,000,000+         30
1,000,000,000+       10
5+                    9
1+                    3
Name: Installs, dtype: int64
```

[22]:
```
dataset["Installs"]=dataset["Installs"].str[:-1]
dataset["Installs"]=dataset["Installs"].apply(lambda x:x.replace(",",""))
```

[23]:
```
dataset["Installs"]=dataset["Installs"].astype(int)
```

Change Price

[25]:
```
dataset["Price"].unique()
```

[25]:
```
array(['0', '$4.99', '$6.99', '$7.99', '$3.99', '$5.99', '$2.99', '$1.99',
       '$9.99', '$0.99', '$9.00', '$5.49', '$10.00', '$24.99', '$11.99',
       '$79.99', '$16.99', '$14.99', '$29.99', '$12.99', '$3.49',
       '$10.99', '$7.49', '$1.50', '$19.99', '$15.99', '$33.99', '$39.99',
       '$2.49', '$4.49', '$1.70', '$1.49', '$3.88', '$399.99', '$17.99',
       '$400.00', '$3.02', '$1.76', '$4.84', '$4.77', '$1.61', '$1.59',
       '$6.49', '$1.29', '$299.99', '$379.99', '$37.99', '$18.99',
       '$389.99', '$8.49', '$1.75', '$14.00', '$2.00', '$3.08', '$2.59',
       '$19.40', '$15.46', '$8.99', '$3.04', '$13.99', '$4.29', '$3.28',
       '$4.60', '$1.00', '$2.90', '$1.97', '$2.56', '$1.20'], dtype=object)
```

[26]:
```
dataset["Price"]=dataset["Price"].apply(lambda x:x.replace("$",""))
dataset["Price"]=dataset["Price"].astype(float)
```

[27]:
```
dataset["Price"].unique()
```

[27]:
```
array([  0.  ,   4.99,   6.99,   7.99,   3.99,   5.99,   2.99,   1.99,
         9.99,   0.99,   9.  ,   5.49,  10.  ,  24.99,  11.99,  79.99,
        16.99,  14.99,  29.99,  12.99,   3.49,  10.99,   7.49,   1.5 ,
        19.99,  15.99,  33.99,  39.99,   2.49,   4.49,   1.7 ,   1.49,
         3.88, 399.99,  17.99, 400.  ,   3.02,   1.76,   4.84,   4.77,
         1.61,   1.59,   6.49,   1.29, 299.99, 379.99,  37.99,  18.99,
       389.99,   8.49,   1.75,  14.  ,   2.  ,   3.08,   2.59,  19.4 ,
```

```
       15.46,   8.99,   3.04,  13.99,   4.29,   3.28,   4.6 ,   1.  ,
        2.9 ,   1.97,   2.56,   1.2 ])
```

5 Sanity checks

Average rating should be between 1 and 5

[29]:
```
dataset=dataset.drop(dataset[(dataset.Rating < 1) & (dataset.Rating > 5)].index)
```

[30]:
```
dataset.Rating<1
```

[30]:
```
0       False
1       False
2       False
3       False
4       False
        …
9354    False
9355    False
9356    False
9357    False
9359    False
Name: Rating, Length: 7723, dtype: bool
```

Dropping records with more reviews than installs

[32]:
```
rows=dataset[dataset["Installs"]<dataset["Reviews"]].index
dataset.drop(rows,inplace=True)
```

Dropping free apps, where price>0

[34]:
```
free = dataset.loc[dataset['Type'] == 'Free'].index
```

[35]:
```
import numpy as np
```

[36]:
```
np.sum(dataset.loc[free,'Price'] > 0)
```

[36]:
```
0
```

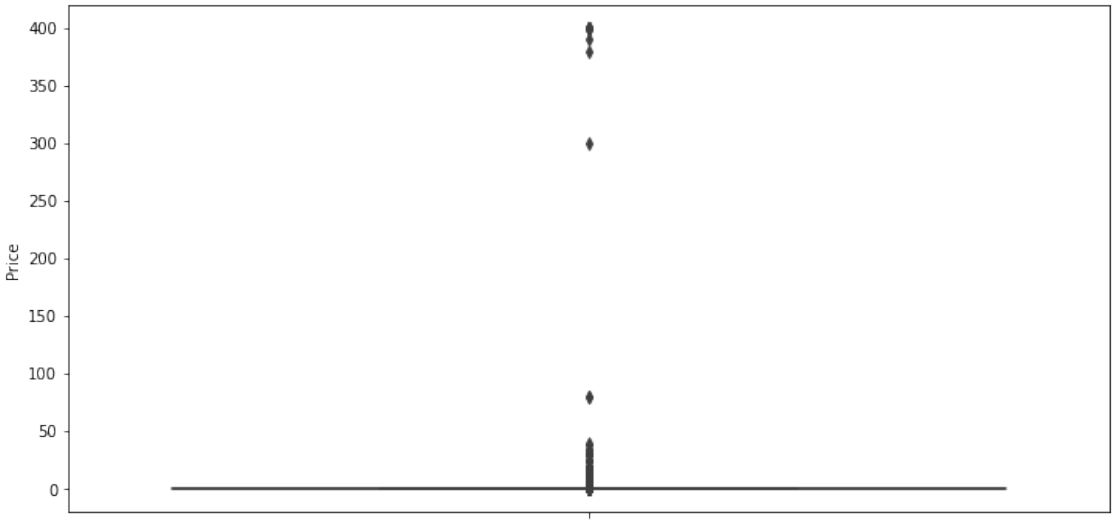
[112]:
```

```

6 Perform Univariate Analysis

[38]:
```
import matplotlib.pyplot as plt
```

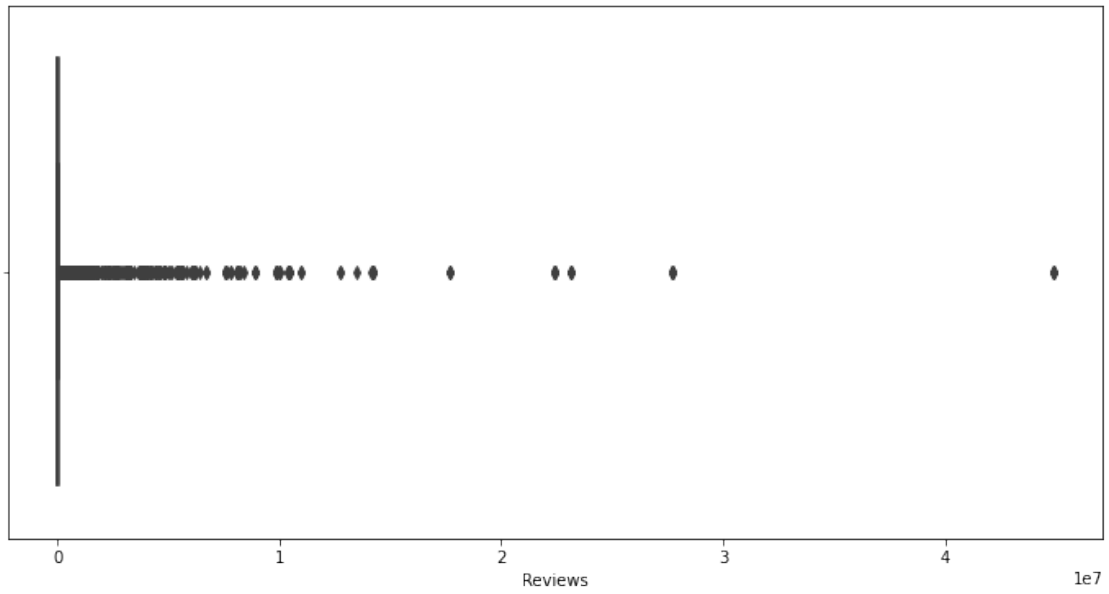
[39]:
```
import seaborn as sns
```

[40]:
```
plt.figure(figsize=(12,6))
sns.boxplot(y='Price',data=dataset)
plt.show()
```



There are apps with very high prices, shown by the wide spread, with the most expensive on the northern end of the graph

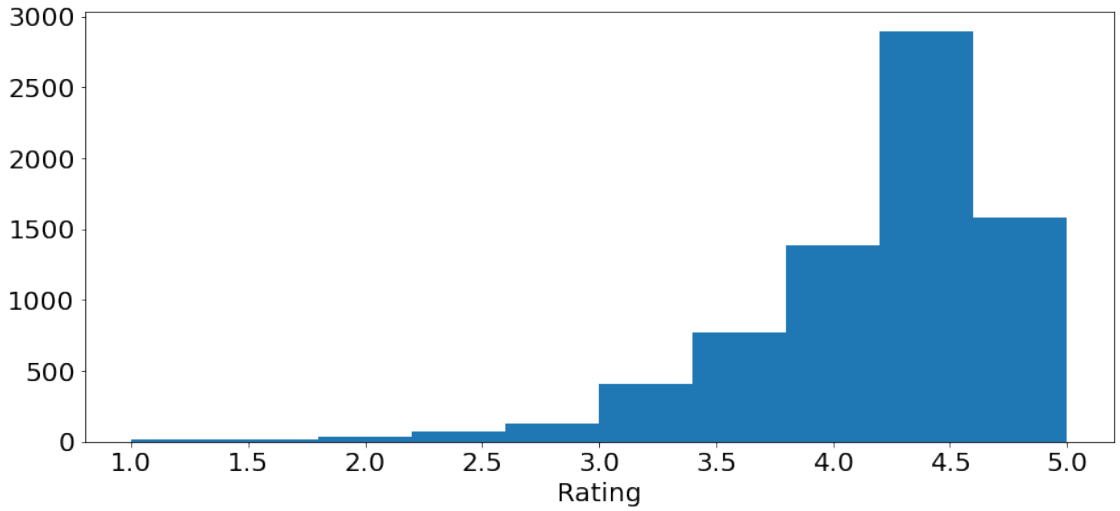
[42]:
```
plt.figure(figsize=(12,6))
sns.boxplot(x='Reviews', data=dataset)
plt.show()
```



There are vey few apps with a high rating, which seem odd

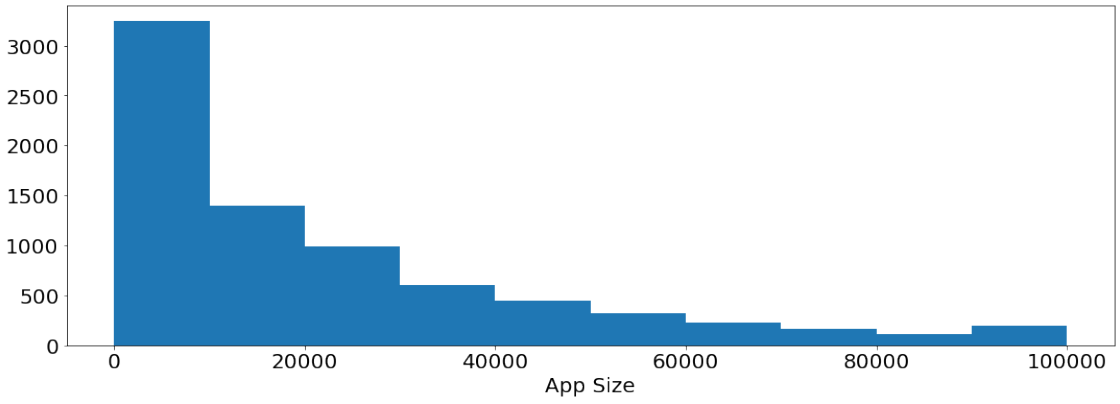
6.1 Histogram

[109]:
```
plt.figure(figsize=(14,6))
plt.hist(dataset.Rating)
plt.rc('xtick', labelsize=10)
plt.rc('ytick', labelsize=10)
plt.xlabel('Rating')
plt.rc('axes', labelsize=20)
plt.show()
```



Ratings are distributed to more of the higher end, particualrly from 4.0 onwards

[47]:
```
plt.figure(figsize=(18,6))
plt.hist(dataset.Size)
plt.rc('xtick', labelsize=15)
plt.rc('ytick', labelsize=15)
plt.xlabel('App Size')
plt.rc('axes',labelsize=22)
plt.show()
```



Higher frequency of apps with lower size.This decreases signficantly as app size increases

7 Outlier treatment

200 is a high price and should be dropped

[51]:
```
rows=dataset[dataset["Price"]>200].index
dataset.drop(rows,inplace=True)
```

7.0.1 Dropping reviews

[53]:
```
rows=dataset[dataset["Reviews"]>2000000].index
dataset.drop(rows,inplace=True)
```

7.0.2 Installs

[55]:
```
dataset.Installs.quantile([0.10, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99])
```

[55]:
```
0.10        1000.0
0.25       10000.0
0.50      100000.0
0.75     1000000.0
0.90    10000000.0
0.95    10000000.0
0.99    50000000.0
Name: Installs, dtype: float64
```

7.0.3 Therefore, a reasonable threshold, should be from 95th percentile

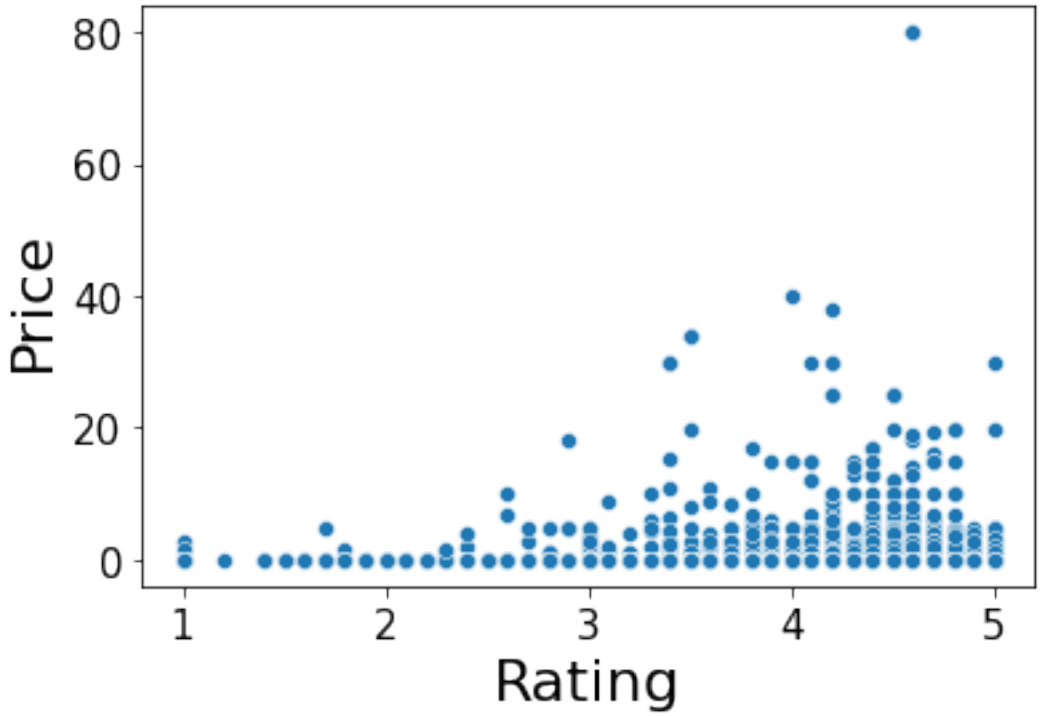
[57]:
```
rows=dataset[dataset["Installs"]>10000000].index
dataset.drop(rows,inplace=True)
```

[58]:
```
dataset.Installs>10000000
```

[58]:
```
0       False
1       False
2       False
4       False
5       False
        ...
9354    False
9355    False
9356    False
9357    False
9359    False
Name: Installs, Length: 7307, dtype: bool
```

8 Bivariate Analysis

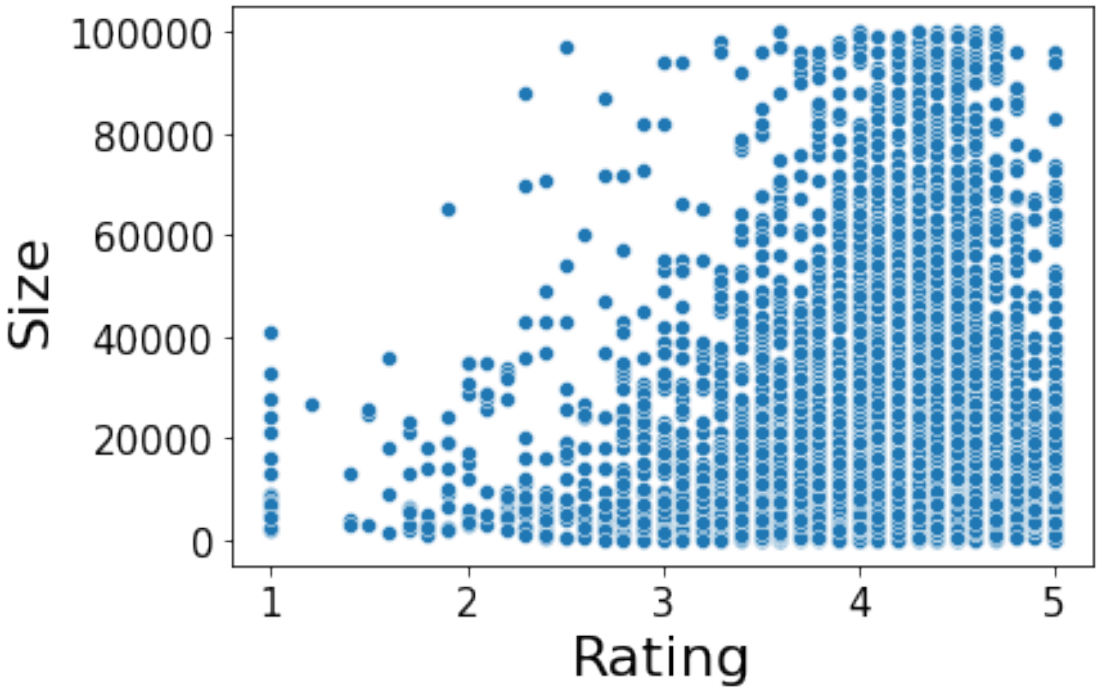
[60]:
```
sns.scatterplot(x="Rating", y="Price", data=dataset)
plt.show()
```



From 2.5 rating onwards, there is a somewhat clear positive correlation between price and rating

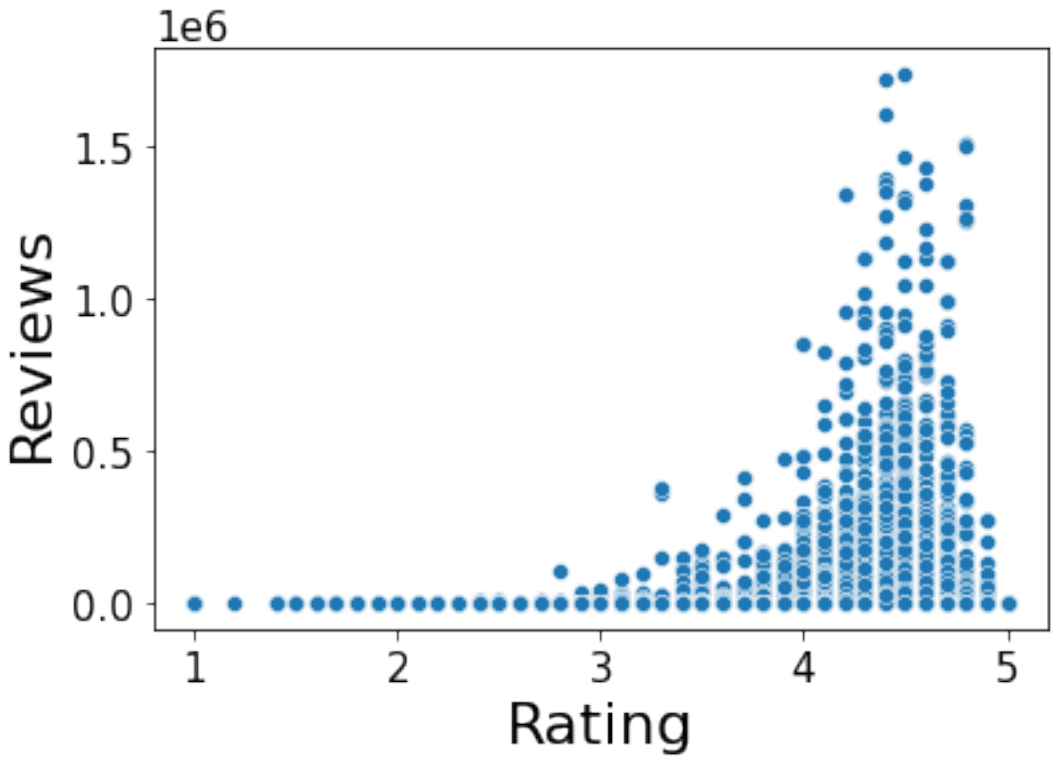
8.0.1 Rating vs Size

[63]:
```
sns.scatterplot(x="Rating", y="Size", data=dataset)
plt.show()
```



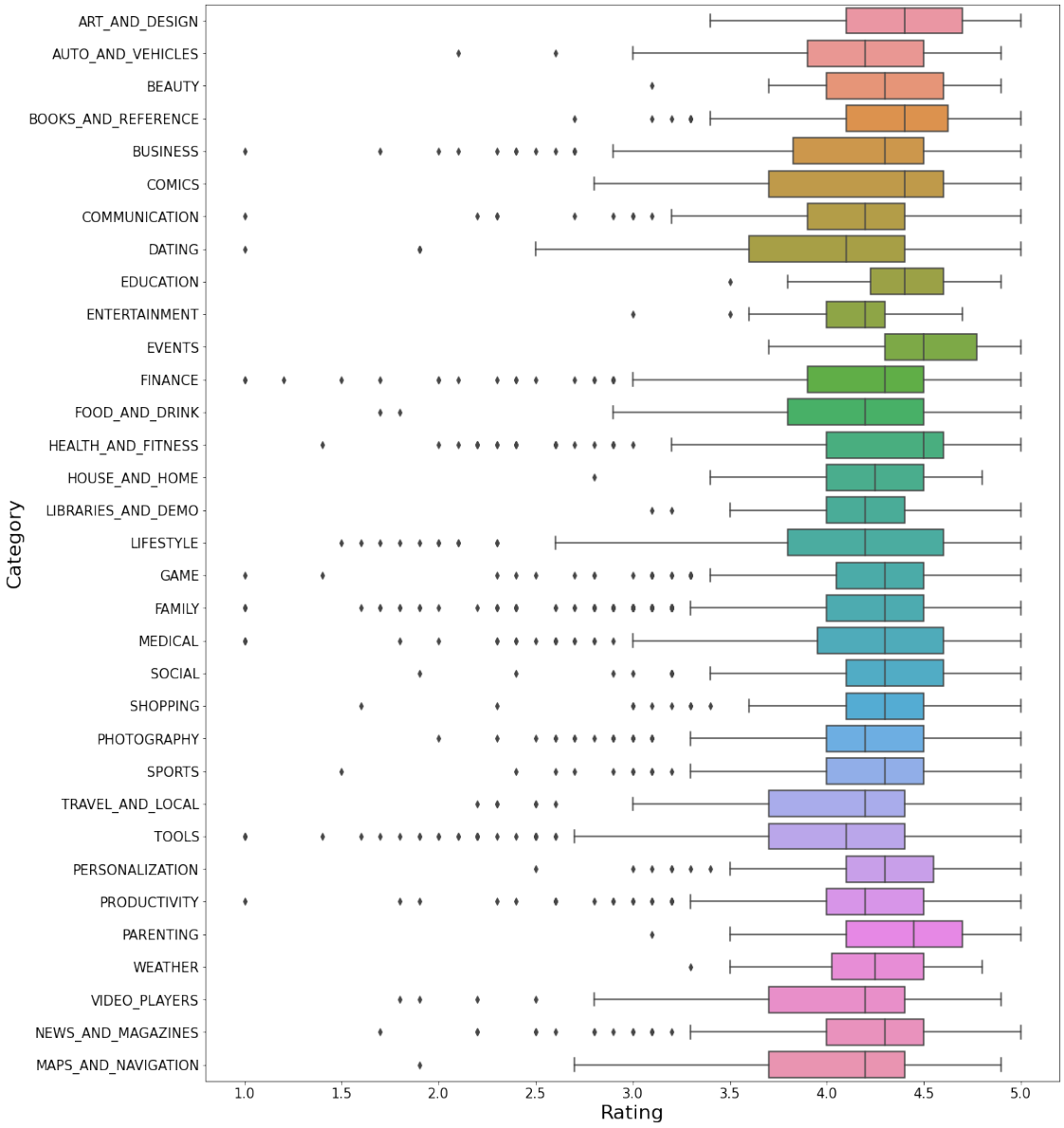
An increase in size does not always mean a higher rating, but heavy apps tend to be rated better than lighter apps

[65]:
```
sns.scatterplot(x="Rating", y="Reviews", data=dataset)
plt.show()
```



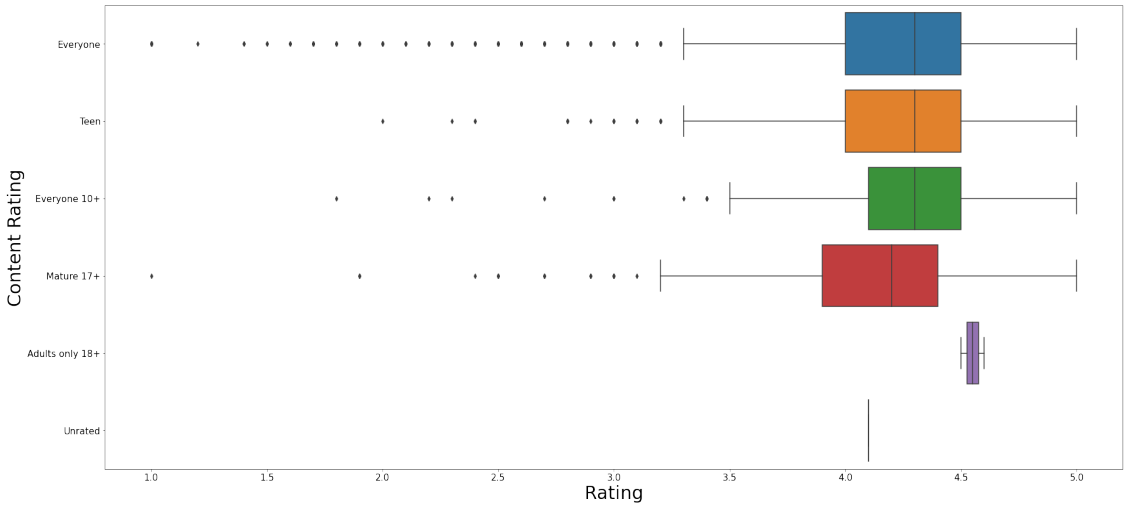
There is some correlation after 3.0 rating that a higher rating is linked to a higher review but it is weak

[67]:
```
plt.figure(figsize=(17,22))
sns.boxplot(x='Rating',y='Category',data=dataset)
plt.rc('font', size=30)
plt.show()
```



Perhaps not the easiest to tell, but I would say Events had the highest rating

[102]:
```
from matplotlib import rcParams
plt.figure(figsize=(30,14))
sns.boxplot(x='Rating',y='Content Rating',data=dataset)
plt.rc('axes', labelsize=30)
plt.rc('xtick', labelsize=8)
plt.rc('ytick', labelsize=8)
plt.show()
```



Whle it is close, I would say Adults only 18+ is the most favourable in terms of rating in terms of different categories

9 Data Preprocessing

[72]:
```
inp1=dataset.copy()
```

9.0.1 Apply log transformation to Reviews and Installs

[74]:
```
inp1['Reviews'] = np.log1p(inp1['Reviews'])
```

[75]:
```
inp1['Installs'] = np.log1p(inp1['Installs'])
```

9.0.2 Drop unwanted columns

[ ]:
```
inp1.drop(columns = { 'App','Last Updated','Current Ver','Android Ver'},
inplace=True)
```

9.0.3 Get dummy columns for Category, Genres, and Content Rating

[78]:
```
dum_cols = ['Category','Genres','Content Rating']
inp2 = pd.get_dummies(inp1,columns=dum_cols,drop_first=True)
inp2
```

[78]:

	Rating	Reviews	Size	Installs	Type	Price	\
0	4.1	5.075174	19000	9.210440	Free	0.0	
1	3.9	6.875232	14000	13.122365	Free	0.0	
2	4.7	11.379520	8700	15.424949	Free	0.0	
4	4.3	6.875232	2800	11.512935	Free	0.0	
5	4.4	5.123964	5600	10.819798	Free	0.0	
…	…	…	…	…	…	…	
9354	4.8	3.806662	619	6.908755	Free	0.0	
9355	4.0	2.079442	2600	6.216606	Free	0.0	
9356	4.5	3.663562	53000	8.517393	Free	0.0	
9357	5.0	1.609438	3600	4.615121	Free	0.0	
9359	4.5	12.894981	19000	16.118096	Free	0.0	

	Category_AUTO_AND_VEHICLES	Category_BEAUTY	\
0	0	0	
1	0	0	
2	0	0	
4	0	0	
5	0	0	
…	…	…	
9354	0	0	
9355	0	0	
9356	0	0	
9357	0	0	
9359	0	0	

	Category_BOOKS_AND_REFERENCE	Category_BUSINESS	…	\
0	0	0	…	
1	0	0	…	
2	0	0	…	
4	0	0	…	
5	0	0	…	
…	…	…	…	
9354	1	0	…	
9355	0	0	…	
9356	0	0	…	
9357	0	0	…	
9359	0	0	…	

	Genres_Video Players & Editors	\
0	0	
1	0	
2	0	
4	0	
5	0	
…	…	
9354	0	

```
9355                                0
9356                                0
9357                                0
9359                                0

      Genres_Video Players & Editors;Creativity  \
0                                             0
1                                             0
2                                             0
4                                             0
5                                             0
…                                           …
9354                                          0
9355                                          0
9356                                          0
9357                                          0
9359                                          0

      Genres_Video Players & Editors;Music & Video  Genres_Weather  \
0                                                0               0
1                                                0               0
2                                                0               0
4                                                0               0
5                                                0               0
…                                              …               …
9354                                             0               0
9355                                             0               0
9356                                             0               0
9357                                             0               0
9359                                             0               0

      Genres_Word  Content Rating_Everyone  Content Rating_Everyone 10+  \
0               0                        1                            0
1               0                        1                            0
2               0                        1                            0
4               0                        1                            0
5               0                        1                            0
…             …                        …                            …
9354            0                        1                            0
9355            0                        1                            0
9356            0                        1                            0
9357            0                        1                            0
9359            0                        1                            0

      Content Rating_Mature 17+  Content Rating_Teen  Content Rating_Unrated
0                             0                    0                       0
1                             0                    0                       0
```

```
2                              0                    0                    0
4                              0                    0                    0
5                              0                    0                    0
…                              …                    …                    …
9354                           0                    0                    0
9355                           0                    0                    0
9356                           0                    0                    0
9357                           0                    0                    0
9359                           0                    0                    0

[7307 rows x 154 columns]
```

10 Train and Test split and apply 70-30 split

[104]:
```
from sklearn.model_selection import train_test_split
df_train, df_test = train_test_split(inp2, train_size = 0.7, random_state = 100)
y_train = df_train.Rating
X_train = df_train
y_test = df_test.Rating
X_test = df_test
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

[104]:
```
LinearRegression()
```

[103]:
```
inp2.pop('Type')
```

[103]:
```
0       Free
1       Free
2       Free
4       Free
5       Free
        …
9354    Free
9355    Free
9356    Free
9357    Free
9359    Free
Name: Type, Length: 7307, dtype: object
```

11 Model building

[105]:
```
from sklearn.model_selection import train_test_split
df_train, df_test = train_test_split(inp2, train_size = 0.7, random_state = 100)
y_train = df_train.Rating
X_train = df_train
y_test = df_test.Rating
X_test = df_test
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

[105]:
```
LinearRegression()
```

[106]:
```
from sklearn.metrics import r2_score
y_train_pred= lr.predict(X_train)
r2_score(y_train, y_train_pred)
```

[106]:
```
1.0
```

11.1 R2 of 1 means regression predictions perfectly fit the data

[115]:
```

```

[ ]:
```

```