In [84]:
```python
#Import Libraries
import numpy as np #linear algebra
import pandas as pd #data processing
import matplotlib.pyplot as plt
import seaborn as sns

import os
print(os.listdir("../input"))
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-84-f3b769404fed> in <module>
      6
      7 import os
----> 8 print(os.listdir("../input"))

FileNotFoundError: [WinError 3] The system cannot find the path specified: '../input'
```

In [85]:
```python
data=pd.read_csv(r"C:\Users\Mohd Abdullah\Desktop\googleplaystore.csv")
```

In [86]:
```python
data.head()
```

Out[86]:

	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	4.2 and up
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	4.4 and up

In [87]:
```python
data.describe()
```

Out[87]:

	Rating
count	9367.000000
mean	4.193338
std	0.537431
min	1.000000
25%	4.000000
50%	4.300000
75%	4.500000
max	19.000000

In [88]:
```python
data.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 [89]:
```python
data.Category.unique()
```

Out[89]:
```
array(['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',
       '1.9'], dtype=object)
```

In [90]:
```python
data.Category=data.Category.map({'ART_AND_DESIGN': 0, 'AUTO_AND_VEHICLES': 1, 'BEAUTY': 2,
       'BOOKS_AND_REFERENCE': 3, 'BUSINESS': 4, 'COMICS': 5, 'COMMUNICATION': 6,
       'DATING': 7, 'EDUCATION': 8, 'ENTERTAINMENT': 9, 'EVENTS': 10, 'FINANCE': 11,
       'FOOD_AND_DRINK': 12, 'HEALTH_AND_FITNESS': 13, 'HOUSE_AND_HOME': 14,
       'LIBRARIES_AND_DEMO': 15, 'LIFESTYLE': 16, 'GAME': 17, 'FAMILY': 18, 'MEDICAL': 18,
       'SOCIAL': 20, 'SHOPPING': 21, 'PHOTOGRAPHY': 22, 'SPORTS': 23, 'TRAVEL_AND_LOCAL': 24,
       'TOOLS': 25, 'PERSONALIZATION': 26, 'PRODUCTIVITY': 27, 'PARENTING': 28, 'WEATHER': 29,
       'VIDEO_PLAYERS': 30, 'NEWS_AND_MAGAZINES': 31, 'MAPS_AND_NAVIGATION': 32,
       '1.9': 33}).astype(float)
```

In [91]:
```python
data["Genres"].unique()
```

Out[91]:
```
array(['Art & Design', 'Art & Design;Pretend Play',
       'Art & Design;Creativity', 'Art & Design;Action & Adventure',
       'Auto & Vehicles', 'Beauty', 'Books & Reference', 'Business',
       'Comics', 'Comics;Creativity', 'Communication', 'Dating',
       'Education;Education', 'Education', 'Education;Creativity',
       'Education;Music & Video', 'Education;Action & Adventure',
       'Education;Pretend Play', 'Education;Brain Games', 'Entertainment',
       'Entertainment;Music & Video', 'Entertainment;Brain Games',
       'Entertainment;Creativity', 'Events', 'Finance', 'Food & Drink',
       'Health & Fitness', 'House & Home', 'Libraries & Demo',
       'Lifestyle', 'Lifestyle;Pretend Play',
       'Adventure;Action & Adventure', 'Arcade', 'Casual', 'Card',
       'Casual;Pretend Play', 'Action', 'Strategy', 'Puzzle', 'Sports',
       'Music', 'Word', 'Racing', 'Casual;Creativity',
       'Casual;Action & Adventure', 'Simulation', 'Adventure', 'Board',
       'Trivia', 'Role Playing', 'Simulation;Education',
       'Action;Action & Adventure', 'Casual;Brain Games',
       'Simulation;Action & Adventure', 'Educational;Creativity',
       'Puzzle;Brain Games', 'Educational;Education', 'Card;Brain Games',
       'Educational;Brain Games', 'Educational;Pretend Play',
       'Entertainment;Education', 'Casual;Education',
       'Music;Music & Video', 'Racing;Action & Adventure',
       'Arcade;Pretend Play', 'Role Playing;Action & Adventure',
       'Simulation;Pretend Play', 'Puzzle;Creativity',
       'Sports;Action & Adventure', 'Educational;Action & Adventure',
       'Arcade;Action & Adventure', 'Entertainment;Action & Adventure',
       'Puzzle;Action & Adventure', 'Strategy;Action & Adventure',
       'Music & Audio;Music & Video', 'Health & Fitness;Education',
       'Adventure;Education', 'Board;Brain Games',
       'Board;Action & Adventure', 'Board;Pretend Play',
       'Casual;Music & Video', 'Role Playing;Pretend Play',
       'Entertainment;Pretend Play', 'Video Players & Editors;Creativity',
       'Card;Action & Adventure', 'Medical', 'Social', 'Shopping',
       'Photography', 'Travel & Local',
       'Travel & Local;Action & Adventure', 'Tools', 'Tools;Education',
       'Personalization', 'Productivity', 'Parenting',
       'Parenting;Music & Video', 'Parenting;Education',
       'Parenting;Brain Games', 'Weather', 'Video Players & Editors',
       'Video Players & Editors;Music & Video', 'News & Magazines',
       'Maps & Navigation', 'Health & Fitness;Action & Adventure',
       'Educational', 'Casino', 'Adventure;Brain Games',
       'Trivia;Education', 'Lifestyle;Education',
       'Books & Reference;Creativity', 'Books & Reference;Education',
       'Puzzle;Education', 'Role Playing;Education',
       'Role Playing;Brain Games', 'Strategy;Education',
       'Racing;Pretend Play', 'Communication;Creativity',
       'February 11, 2018', 'Strategy;Creativity'], dtype=object)
```

In [92]:
```python
genresVal= data["Genres"].unique()
genresValCount =len(genresVal)
genres_dict ={}
for i in range(0,genresValCount):
    genres_dict[genresVal[i]]=i
data["Genres"] = data["Genres"].map(genres_dict).astype(int)
```

In [93]:
```python
data['Content Rating'].unique()
```

Out[93]:
```
array(['Everyone', 'Teen', 'Everyone 10+', 'Mature 17+',
       'Adults only 18+', 'Unrated', nan], dtype=object)
```

In [94]:
```python
data['Content Rating']=data['Content Rating'].map({'Everyone': 0, 'Teen': 1, 'Everyone 10+': 2, 'Mature 17+': 3,
       'Adults only 18+': 4}).astype(float)
```

In [95]:
```python
data['Reviews']= [float(i.split('M')[0])if 'M' in i else float(i) for i in data ['Reviews']]
```

In [96]:
```python
data["Size"]= [float(i.split('M')[0])if 'M' in i else float(0) for i in data ["Size"]]
```

In [97]:
```python
data["Price"]= [float(i.split('$')[1])if '$' in i else float(0) for i in data ["Price"]]
```

In [98]:
```python
data.Installs.unique()
```

Out[98]:
```
array(['10,000+', '500,000+', '5,000,000+', '50,000,000+', '100,000+',
       '50,000+', '1,000,000+', '10,000,000+', '5,000+', '100,000,000+',
       '1,000,000,000+', '1,000+', '500,000,000+', '50+', '100+', '500+',
       '10+', '1+', '5+', '0+', '0', 'Free'], dtype=object)
```

In [99]:
```python
data["Installs"]= [float(i.replace('+','').replace(',',''))if '+' in i or ',' in i else float(0) for i in data["Installs"]]
```

In [100…
```python
data.drop(["Last Updated","Current Ver","Android Ver","App","Type"],axis=1,inplace=True)
```

In [101…
```python
data["Rating"]=data.groupby("Category")["Rating"].transform(lambda x:x.fillna(x.mean()))
data["Content Rating"]= data[["Content Rating"]].fillna(method="ffill")
```

In [102…
```python
data.head()
```

Out[102…

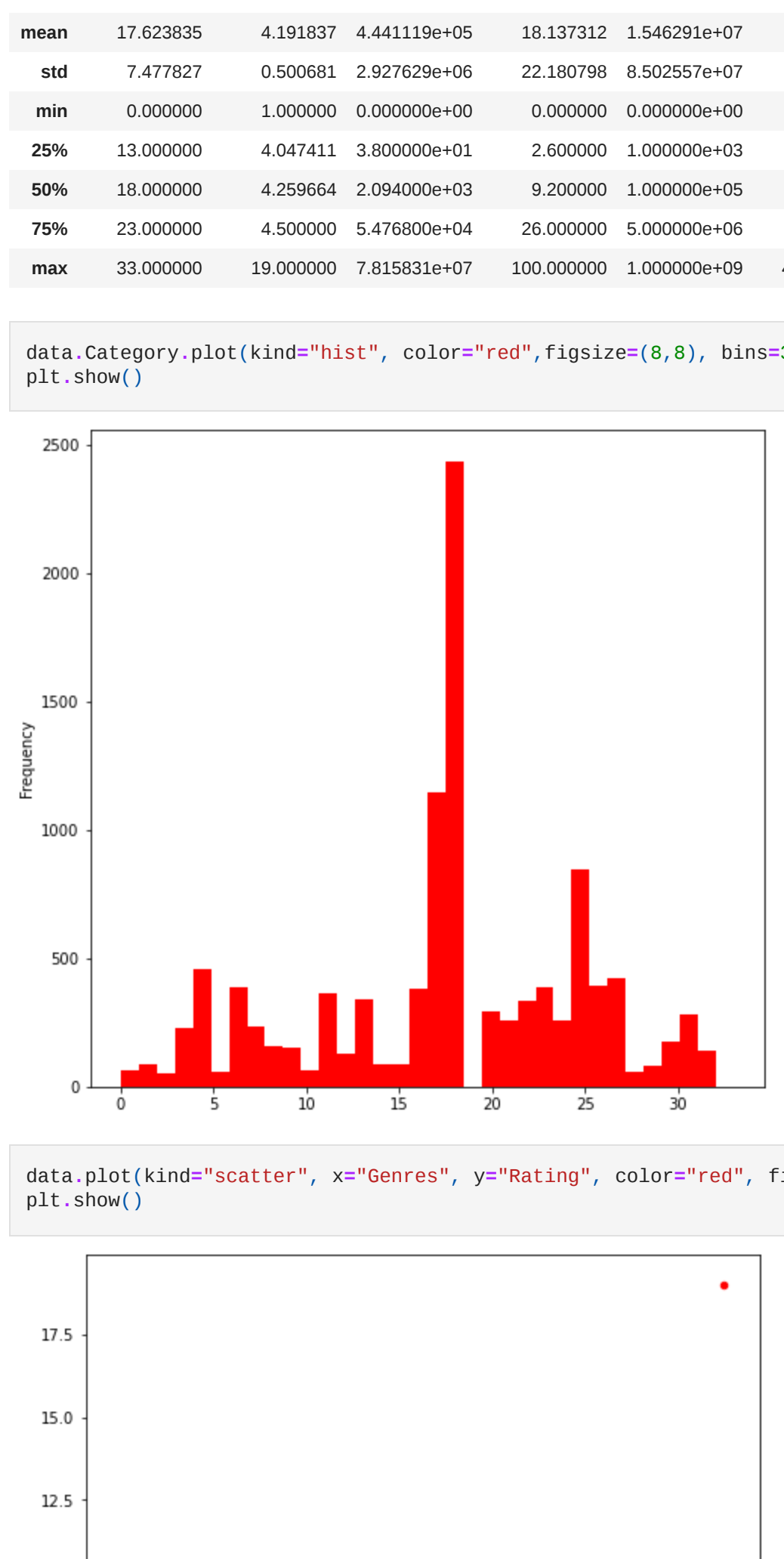
	Category	Rating	Reviews	Size	Installs	Price	Content Rating	Genres
0	0.0	4.1	159.0	19.0	10000.0	0.0	0.0	0
1	0.0	3.9	967.0	14.0	500000.0	0.0	0.0	1
2	0.0	4.7	87510.0	8.7	5000000.0	0.0	0.0	0
3	0.0	4.5	215644.0	25.0	50000000.0	0.0	1.0	0
4	0.0	4.3	967.0	2.8	100000.0	0.0	0.0	2

In [103…
```python
data.describe()
```

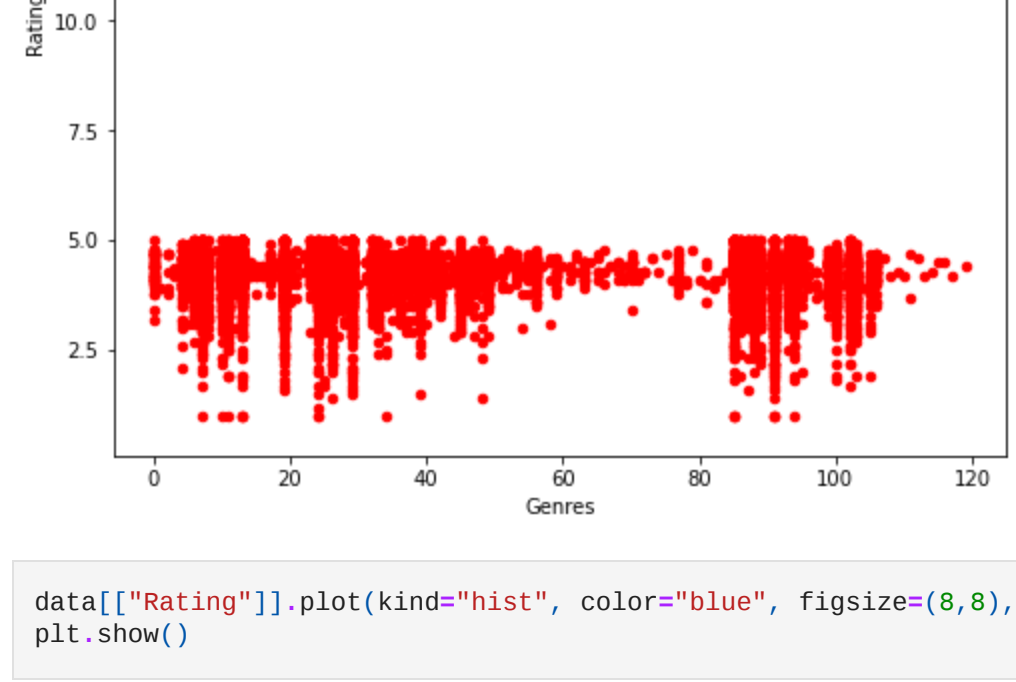
Out[103…

	Category	Rating	Reviews	Size	Installs	Price	Content Rating	Genres
count	10841.000000	10841.000000	1.084100e+04	10841.000000	1.084100e+04	10841.000000	10841.000000	10841.000000
mean	17.623835	4.191837	4.441119e+05	18.137312	1.546291e+07	1.027273	0.327092	50.468315
std	7.477827	0.500681	2.927629e+06	22.180798	8.502557e+07	15.948971	0.758964	34.495916
min	0.000000	1.000000	0.000000e+00	0.000000	0.000000e+00	0.000000	0.000000	0.000000
25%	13.000000	4.047411	3.800000e+01	2.600000	1.000000e+03	0.000000	0.000000	19.000000
50%	18.000000	4.259664	2.094000e+03	9.200000	1.000000e+05	0.000000	0.000000	38.000000
75%	23.000000	4.500000	5.476800e+04	26.000000	5.000000e+06	0.000000	0.000000	89.000000
max	33.000000	19.000000	7.815831e+07	100.000000	1.000000e+09	400.000000	4.000000	119.000000

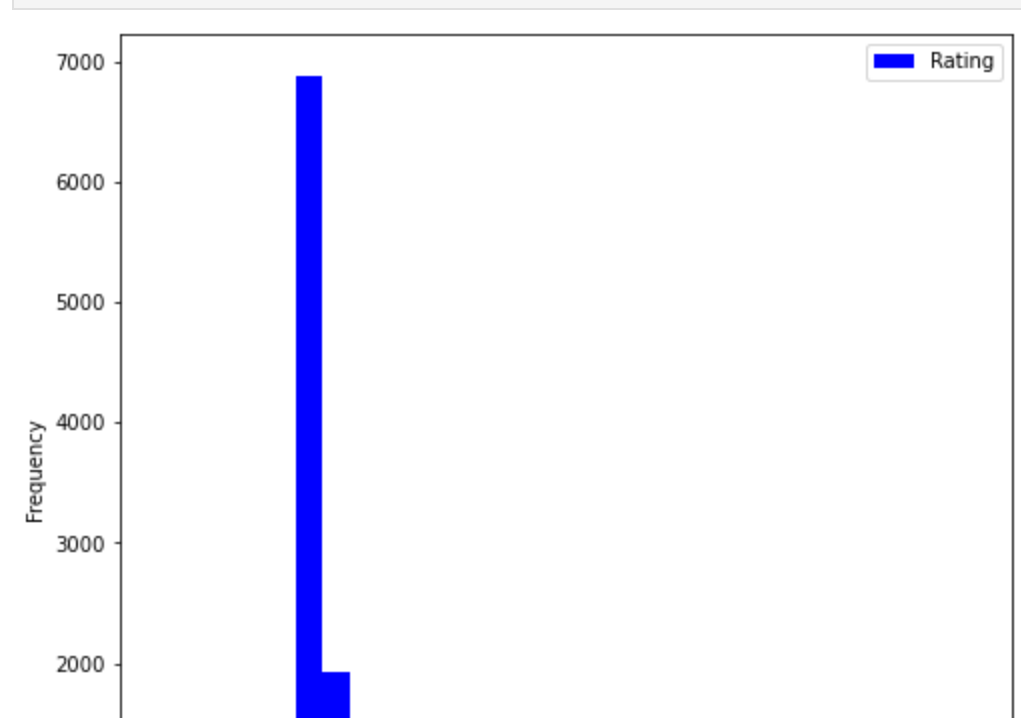
In [104…
```python
data.Category.plot(kind="hist", color="red",figsize=(8,8), bins=34)
plt.show()
```



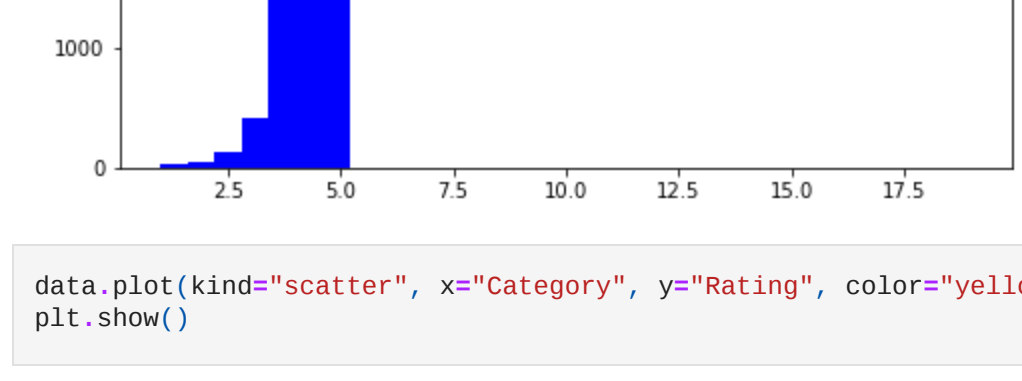
In [105…
```python
data.plot(kind="scatter", x="Genres", y="Rating", color="red", figsize=(8,8))
plt.show()
```



In [106…
```python
data[["Rating"]].plot(kind="hist", color="blue", figsize=(8,8), bins=30)
plt.show()
```



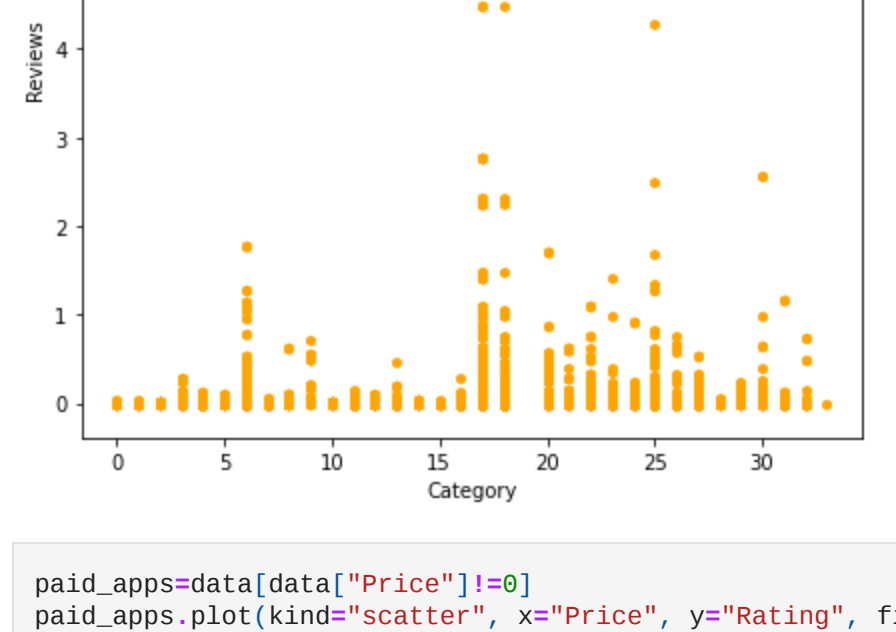
In [119…
```python
data.plot(kind="scatter", x="Category", y="Rating", color="yellow", figsize=(8,4))
plt.show()
```



In [108…
```python
beuty_and_whather_data = data[(data['Category']==2) | (data['Category']==29)]
beuty_and_whather_data["Rating"].min()
```

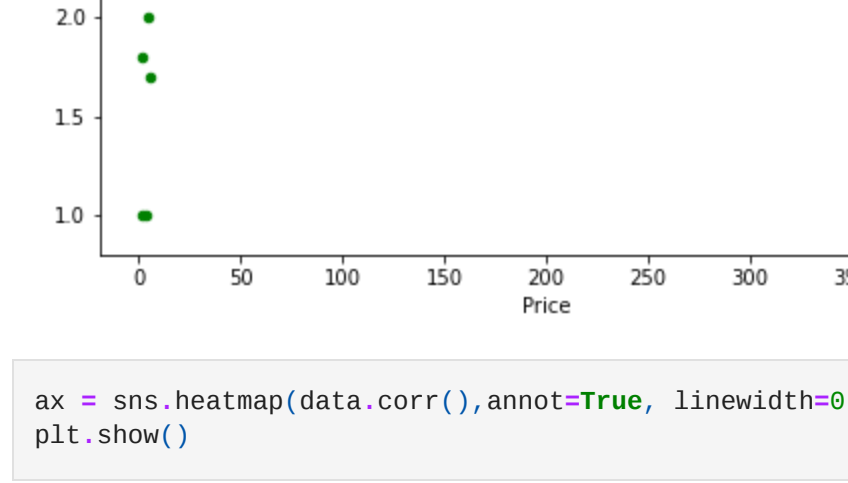
Out[108… 3.1

In [109…
```python
data.plot(kind="scatter", x="Category", y="Reviews", color="orange", marker="h", figsize=(7,7))
plt.show()
```



In [110…
```python
paid_apps=data[data["Price"]!=0]
paid_apps.plot(kind="scatter", x="Price", y="Rating", figsize=(8,8), color="green")
plt.show()
```

In [111…
```python
ax = sns.heatmap(data.corr(),annot=True, linewidth=0.5, fmt='.1f')
plt.show()
```



In [112…
```python
data.describe()
```

Out[112…

	Category	Rating	Reviews	Size	Installs	Price	Content Rating	Genres
count	10841.000000	10841.000000	1.084100e+04	10841.000000	1.084100e+04	10841.000000	10841.000000	10841.000000
mean	17.623835	4.191837	4.441119e+05	18.137312	1.546291e+07	1.027273	0.327092	50.468315
std	7.477827	0.500681	2.927629e+06	22.180798	8.502557e+07	15.948971	0.758964	34.495916
min	0.000000	1.000000	0.000000e+00	0.000000	0.000000e+00	0.000000	0.000000	0.000000
25%	13.000000	4.047411	3.800000e+01	2.600000	1.000000e+03	0.000000	0.000000	19.000000
50%	18.000000	4.259664	2.094000e+03	9.200000	1.000000e+05	0.000000	0.000000	38.000000
75%	23.000000	4.500000	5.476800e+04	26.000000	5.000000e+06	0.000000	0.000000	89.000000
max	33.000000	19.000000	7.815831e+07	100.000000	1.000000e+09	400.000000	4.000000	119.000000

In [113…
```python
from sklearn.model_selection import train_test_split
X = data.drop(["Rating"], axis=1)
Y = data.Rating
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.30)
```

In [114…
```python
from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
X_train_scaled = min_max_scaler.fit_transform(X_train)
X_test_scaled = min_max_scaler.fit_transform(X_test)
```

In [115…
```python
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

In [116…
```python
neigh = KNeighborsRegressor(n_neighbors=1, metric= 'chebyshev')
neigh.fit(X_train_scaled, Y_train)
knn_pred = neigh.predict(X_test_scaled)
r2_score(Y_test, knn_pred)
```

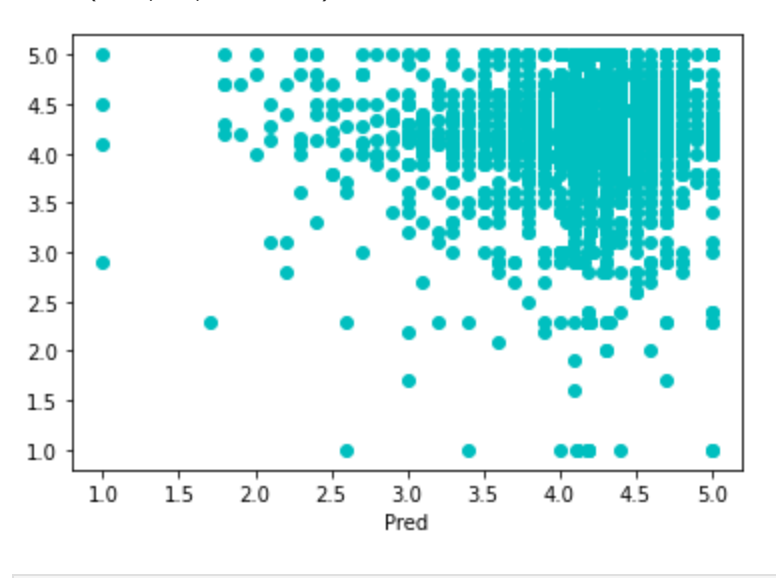
Out[116… -0.9609555082912618

In [117…
```python
mean_squared_error(Y_test, knn_pred)
```

Out[117… 0.44026153063783907

In [118…
```python
plt.scatter(x=Y_test, y=knn_pred, color='c')
plt.xlabel("Pred")
```

Out[118… Text(0.5, 0, 'Pred')



In [ ]: