In [1]:

```
# Amazon Top 50 Bestselling Books 2009-2022
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
data =pd.read_csv(r"C:\Users\ahmet\Downloads\bestsellers_with_categories_2022_03_27.csv")
df = pd.DataFrame(data)
```

In [3]:

```
df.head()
```

Out[3]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	Act Like a Lady, Think Like a Man: What Men Re...	Steve Harvey	4.6	5013	17	2009	Non Fiction
1	Arguing with Idiots: How to Stop Small Minds a...	Glenn Beck	4.6	798	5	2009	Non Fiction
2	Breaking Dawn (The Twilight Saga, Book 4)	Stephenie Meyer	4.6	9769	13	2009	Fiction
3	Crazy Love: Overwhelmed by a Relentless God	Francis Chan	4.7	1542	14	2009	Non Fiction
4	Dead And Gone: A Sookie Stackhouse Novel (Sook...	Charlaine Harris	4.6	1541	4	2009	Fiction

In [4]:

```
#Shows descriptive statistics data
df.describe()
```

Out[4]:

	User Rating	Reviews	Price	Year
count	700.000000	700.000000	700.000000	700.000000
mean	4.639857	19255.195714	12.700000	2015.500000
std	0.218586	23613.443875	9.915162	4.034011
min	3.300000	37.000000	0.000000	2009.000000
25%	4.500000	4987.250000	7.000000	2012.000000
50%	4.700000	10284.000000	11.000000	2015.500000

	User Rating	Reviews	Price	Year
75%	4.800000	23358.000000	15.000000	2019.000000
max	4.900000	208917.000000	105.000000	2022.000000

In [56]:

```
#cheking for null values
for i in df.columns:
  print(i,"\t-\t", df[i].isna().mean()*100)
```

```
Name    -        0.0
Author  -        0.0
User Rating     -        0.0
Reviews         -        0.0
Price   -        0.0
Year    -        0.0
Genre   -        0.0
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 700 entries, 0 to 699
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Name         700 non-null    object 
 1   Author       700 non-null    object 
 2   User Rating  700 non-null    float64
 3   Reviews      700 non-null    int64  
 4   Price        700 non-null    int64  
 5   Year         700 non-null    int64  
 6   Genre        700 non-null    object 
dtypes: float64(1), int64(3), object(3)
memory usage: 38.4+ KB
```

In [6]:

```
#sort the values of the 10 books in a ascending order from top to bottom by User Ratings.
top10=df.sort_values('User Rating',ascending=False)[:10]
```

In [32]:

```
top10
```

Out[32]:

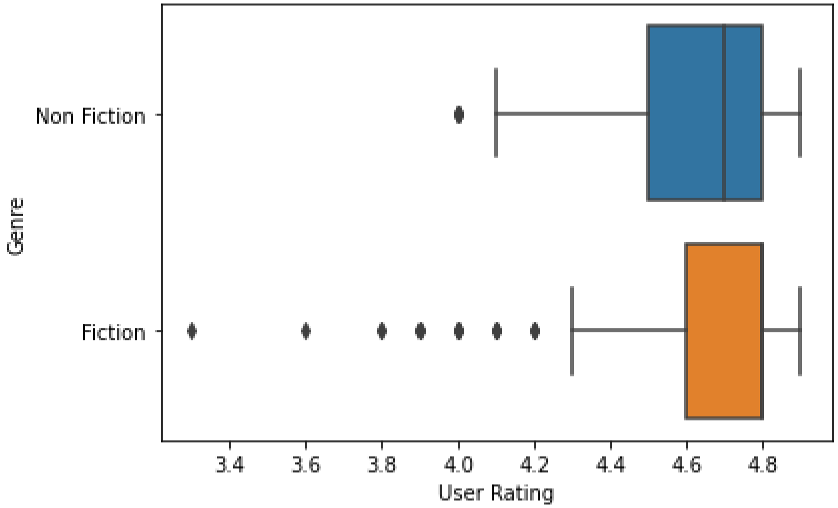
	Name	Author	User Rating	Reviews	Price	Year	Genre
605	Brown Bear, Brown Bear, What Do You See?	Bill Martin Jr.	4.9	38969	5	2021	Fiction
607	Call Us What We Carry: Poems	Amanda Gorman	4.9	2873	14	2021	Fiction
457	Dog Man: Brawl of the Wild: From the Creator o...	Dav Pilkey	4.9	7235	4	2018	Fiction
456	Dog Man and Cat Kid: From the Creator of Capta...	Dav Pilkey	4.9	5062	6	2018	Fiction
223	Oh, the Places You'll Go!	Dr. Seuss	4.9	21834	8	2013	Fiction
586	The Deep End (Diary of a Wimpy Kid Book 15)	Jeff Kinney	4.9	38674	7	2020	Fiction
227	Rush Revere and the Brave Pilgrims: Time-Trave...	Rush Limbaugh	4.9	7150	12	2013	Fiction
443	The Wonderful Things You Will Be	Emily Winfield Martin	4.9	8842	10	2017	Fiction
592	The Very Hungry Caterpillar	Eric Carle	4.9	47260	5	2020	Fiction
441	The Very Hungry Caterpillar	Eric Carle	4.9	19546	5	2017	Fiction

In [7]:

```
#Used seaborn to graph the genre and the user ratings.
#Acorrding to the graph  fiction is more popular than non-fiction on Amazon
sns.boxplot(x ='User Rating', y = 'Genre',data =df)
```

Out[7]:

```
<AxesSubplot:xlabel='User Rating', ylabel='Genre'>
```

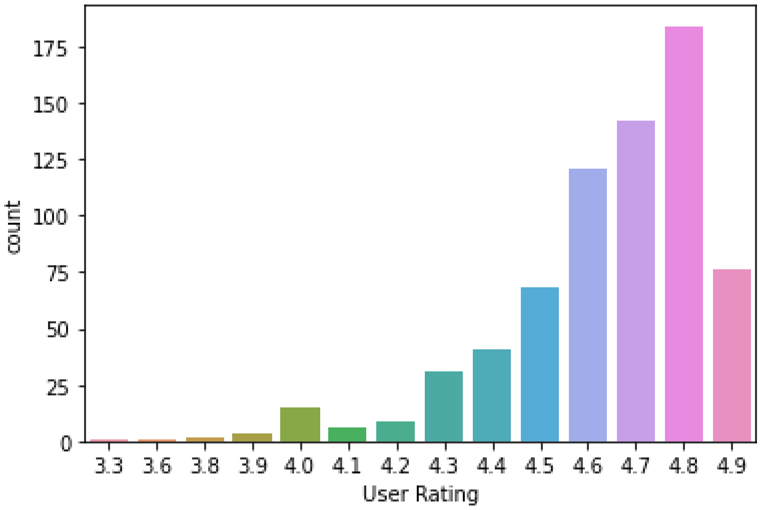


In [8]:

```
#shows a visual reaapresentation of user rating
sns.countplot(x = df['User Rating'])
```

Out[8]:

```
<AxesSubplot:xlabel='User Rating', ylabel='count'>
```



In [9]:

```
#Used pandas to extract data from the column user ratings that is equal to 4.9
#I used the groupby function to group author and user rating coulmn.
#Shows the top authors with the highest ratings.
bestsellers = df[df['User Rating']==5.0]
bestsellers = bestsellers.groupby('Author')['User Rating']
```

In [26]:

```
bestsellers
```

Out[26]:

```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x0000018E8443E8E0>
```

In [10]:

```
# I made the new year set to years from 2009-2022.
#I used the mean() function to give the average of the other numeric columns.
#Reset index to reset the index after making modifications to the column
pyear = df.groupby('Year').mean().reset_index()
pyear['Year'] = [ 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]
pyear
```

Out[10]:

	Year	User Rating	Reviews	Price

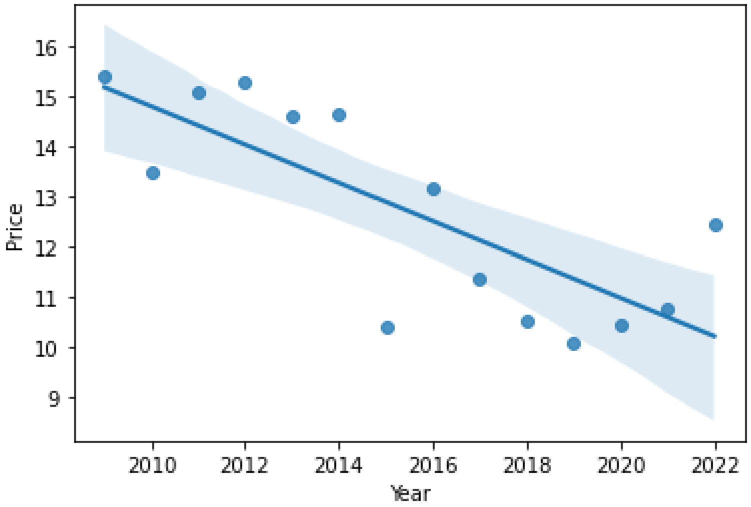
	Year	User Rating	Reviews	Price
0	2009	4.584	4710.12	15.40
1	2010	4.558	5479.62	13.48
2	2011	4.558	8100.82	15.10
3	2012	4.532	13090.92	15.30
4	2013	4.554	13098.14	14.60
5	2014	4.622	15859.94	14.64
6	2015	4.648	14233.38	10.42
7	2016	4.678	14196.00	13.18
8	2017	4.660	12888.40	11.38
9	2018	4.668	13930.42	10.52
10	2019	4.740	15898.34	10.08
11	2020	4.726	52349.94	10.46
12	2021	4.738	44859.48	10.78
13	2022	4.692	40877.22	12.46

In [11]:

```
#Performing EDA on the data.
#plots the linear regression model of the data from x and y.
#From the data the amazon price of books declined as the years went on.
sns.regplot(x="Year", y="Price",data=pyear)
```

Out[11]:

```
<AxesSubplot:xlabel='Year', ylabel='Price'>
```

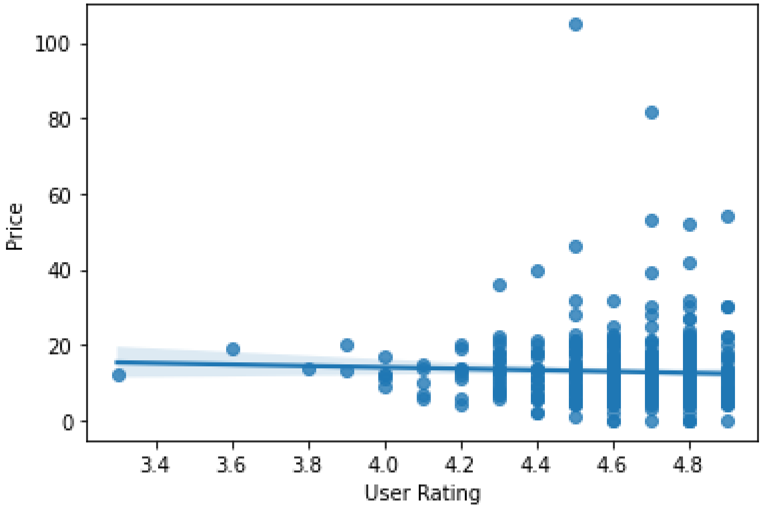


In [25]:

```
sns.regplot(x=data['User Rating'], y=data['Price'])
```

Out[25]:

```
<AxesSubplot:xlabel='User Rating', ylabel='Price'>
```

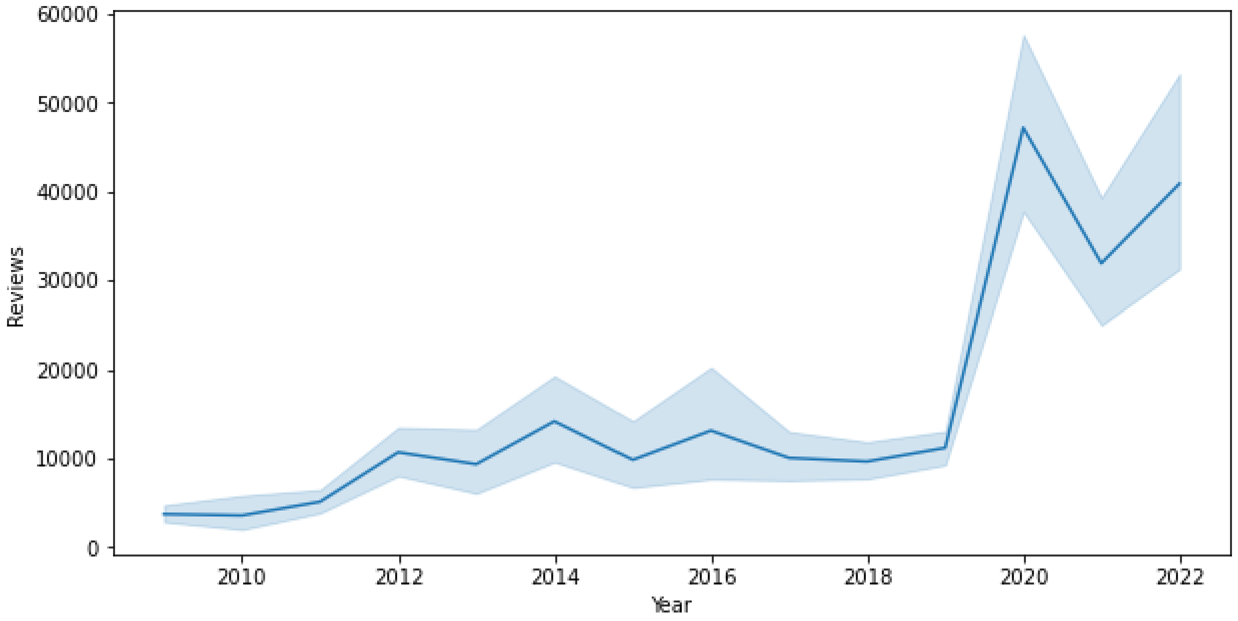


In [29]:

```
fig, ax = plt.subplots(figsize=(10, 5))
sns.lineplot(y='Reviews', x='Year', data=data, ax=ax)
```

Out[29]:

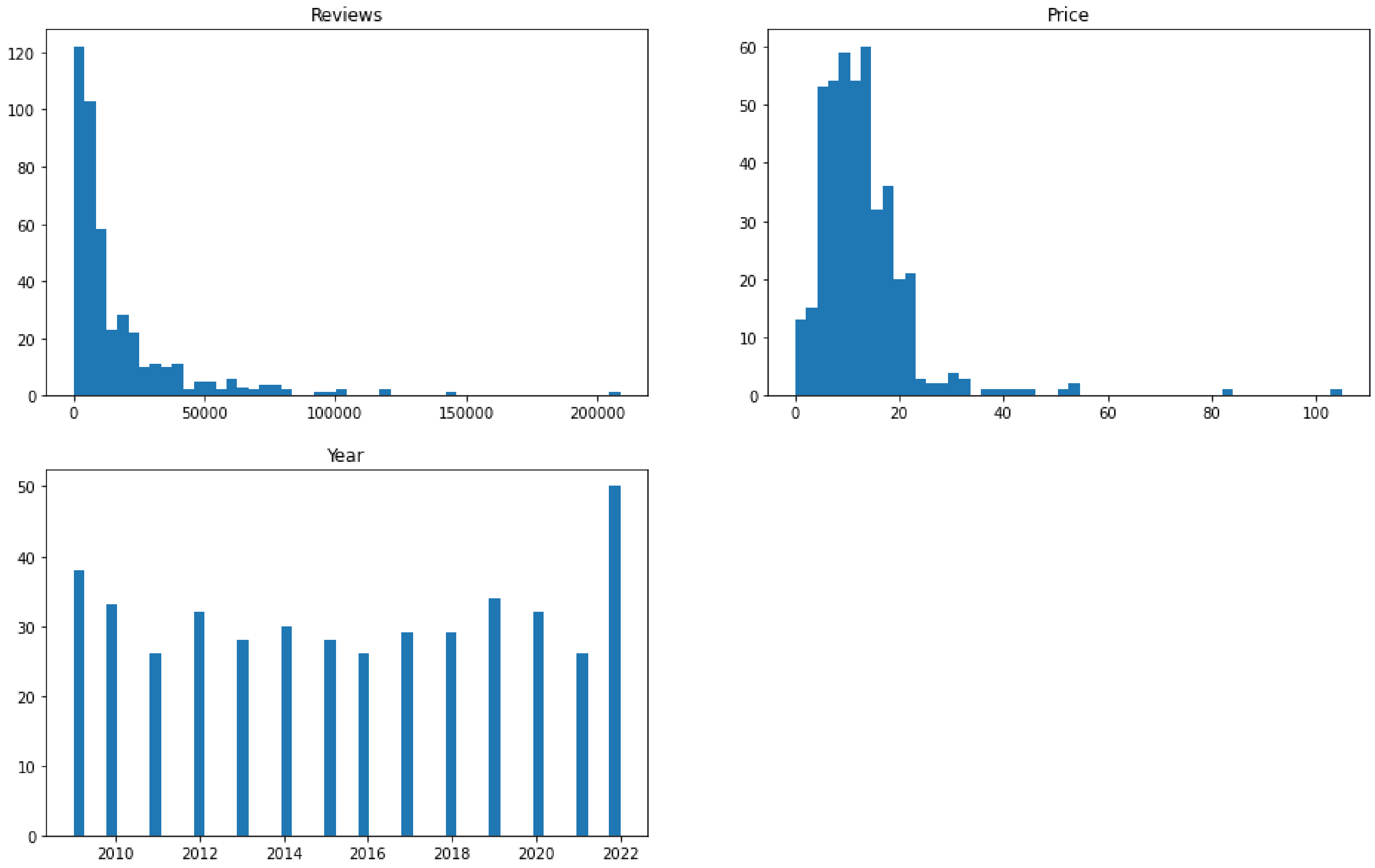
```
<AxesSubplot:xlabel='Year', ylabel='Reviews'>
```



In [33]:

```
fig, axs = plt.subplots(2, 2, figsize=(16,10))
fig.delaxes(axs[1,1])

axs[0,0].hist(data['Reviews'], bins=50)
axs[0,1].hist(data['Price'], bins=50)
axs[1,0].hist(data['Year'], bins=50)
axs[0,0].title.set_text('Reviews')
axs[0,1].title.set_text('Price')
axs[1,0].title.set_text('Year')
plt.show()
```

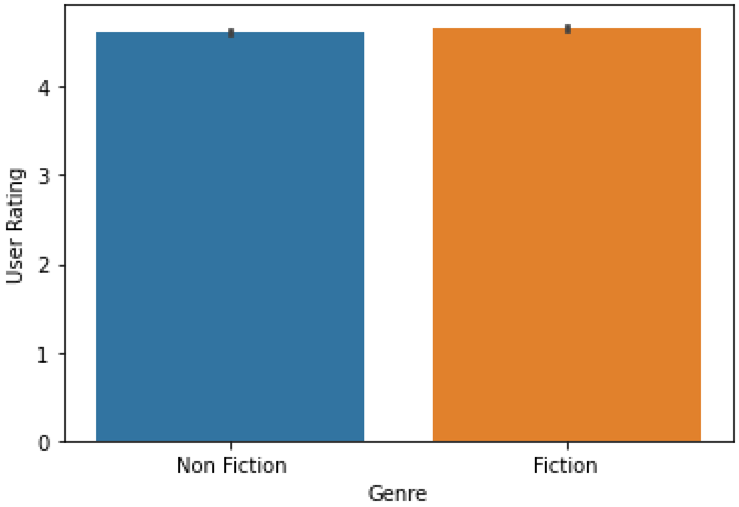


In [39]:

```
#Used seaborn to graph the genre and the user ratings.
#Acorrding to the graph  fiction is more popular than non-fiction on Amazon
sns.barplot(y='User Rating', x = 'Genre',data =df)
```

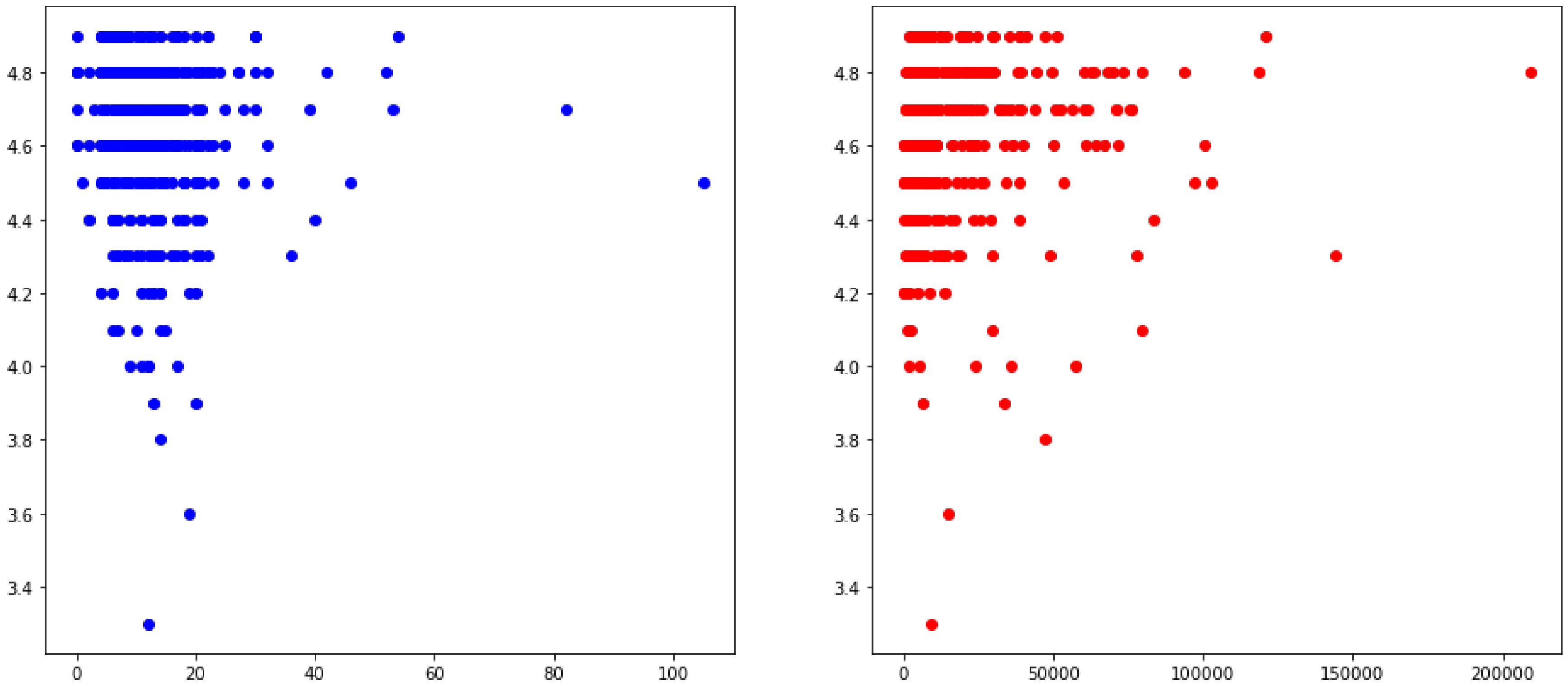
Out[39]:

```
<AxesSubplot:xlabel='Genre', ylabel='User Rating'>
```



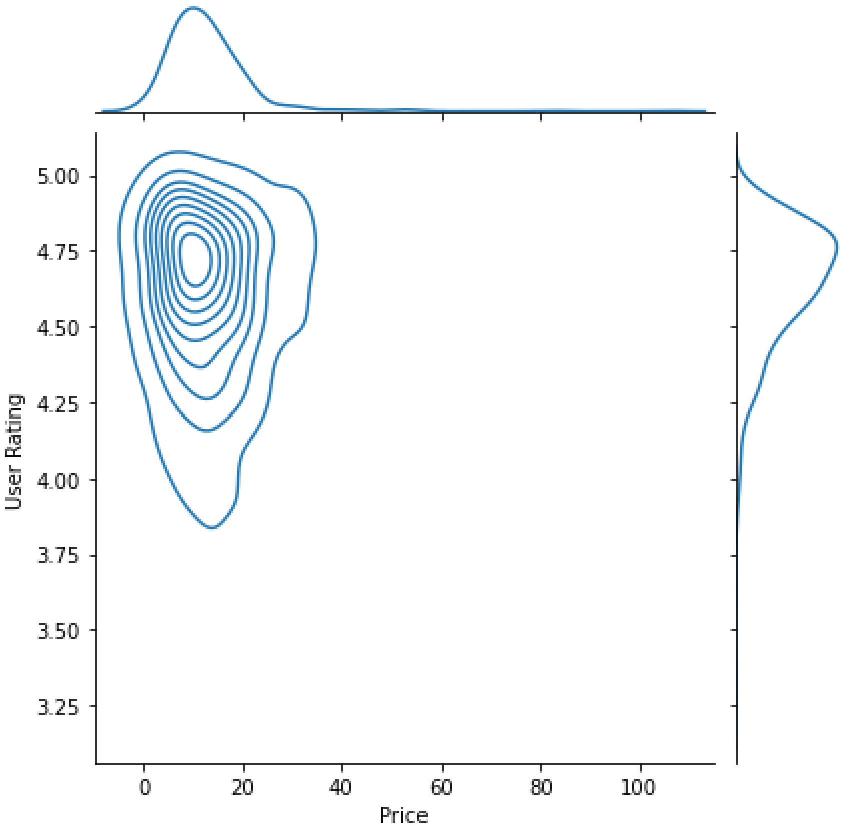
In [40]:

```
fig, ax = plt.subplots(1, 2, figsize=(16,7))
ax[0].scatter('Price', 'User Rating', data=data, color='b')
ax[1].scatter('Reviews', 'User Rating', data=data, color='r')
plt.show()
```



In [41]:
```
sns.jointplot(x=data['Price'], y=data['User Rating'], kind="kde")
```

Out[41]:
```
<seaborn.axisgrid.JointGrid at 0x18e8d876610>
```



In [42]:

```
#Creating new dataframe by copying the existing one so we can use it later without errors.
data_for_tree = data.copy(deep=True)
data_for_tree
```

Out[42]:

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	Act Like a Lady, Think Like a Man: What Men Re...	Steve Harvey	4.6	5013	17	2009	Non Fiction
1	Arguing with Idiots: How to Stop Small Minds a...	Glenn Beck	4.6	798	5	2009	Non Fiction
2	Breaking Dawn (The Twilight Saga, Book 4)	Stephenie Meyer	4.6	9769	13	2009	Fiction
4	Dead And Gone: A Sookie Stackhouse Novel (Sook...	Charlaine Harris	4.6	1541	4	2009	Fiction
5	Diary of a Wimpy Kid: The Last Straw (Book 3)	Jeff Kinney	4.8	3837	15	2009	Fiction
...	...	...	...	...	...	...	...

	Name	Author	User Rating	Reviews	Price	Year	Genre
695	The Wonderful Things You Will Be	Emily Winfield Martin	4.9	20920	9	2022	Fiction
696	Ugly Love: A Novel	Colleen Hoover	4.7	33929	10	2022	Fiction
697	Verity	Colleen Hoover	4.6	71826	11	2022	Fiction
698	What to Expect When You're Expecting	Heidi Murkoff	4.8	27052	13	2022	Non Fiction
699	Where the Crawdads Sing	Delia Owens	4.8	208917	10	2022	Fiction

441 rows × 7 columns

In [43]:

```
data=pd.get_dummies(data, drop_first=True, columns=['Year', 'Genre'])
data.head()
```

Out[43]:

	Name	Author	User Rating	Reviews	Price	Year_2010	Year_2011	Year_2012	Year_2013	Year_2014	Year_2015	Year_2016	Year_2017	Ye
0	Act Like a Lady, Think Like a Man: What Men Re...	Steve Harvey	4.6	5013	17	0	0	0	0	0	0	0	0	
1	Arguing with Idiots: How to Stop Small Minds a...	Glenn Beck	4.6	798	5	0	0	0	0	0	0	0	0	
2	Breaking Dawn (The Twilight Saga, Book 4)	Stephenie Meyer	4.6	9769	13	0	0	0	0	0	0	0	0	
4	Dead And Gone: A Sookie Stackhouse Novel (Sook...	Charlaine Harris	4.6	1541	4	0	0	0	0	0	0	0	0	

	Name	Author	User Rating	Reviews	Price	Year_2010	Year_2011	Year_2012	Year_2013	Year_2014	Year_2015	Year_2016	Year_2017	Ye
5	Diary of a Wimpy Kid: The Last Straw (Book 3)	Jeff Kinney	4.8	3837	15	0	0	0	0	0	0	0	0	

In [44]:

```
#Plotting Correlation
data.corr()
```

Out[44]:

	User Rating	Reviews	Price	Year_2010	Year_2011	Year_2012	Year_2013	Year_2014	Year_2015	Year_2016	Year_2017	Year_20
User Rating	1.000000	0.042372	-0.044217	-0.109122	-0.118072	-0.101660	-0.103623	-0.061572	0.014842	0.065889	-0.015649	-0.0073
Reviews	0.042372	1.000000	-0.071806	-0.163726	-0.127197	-0.073897	-0.084071	-0.030129	-0.078449	-0.039365	-0.077450	-0.0821
Price	-0.044217	-0.071806	1.000000	-0.031985	0.040620	0.064040	-0.030047	0.125271	-0.093514	0.072950	-0.018545	-0.0502
Year_2010	-0.109122	-0.163726	-0.031985	1.000000	-0.071185	-0.079550	-0.074051	-0.076836	-0.074051	-0.071185	-0.075453	-0.0754
Year_2011	-0.118072	-0.127197	0.040620	-0.071185	1.000000	-0.070013	-0.065173	-0.067624	-0.065173	-0.062651	-0.066407	-0.0664
Year_2012	-0.101660	-0.073897	0.064040	-0.079550	-0.070013	1.000000	-0.072831	-0.075571	-0.072831	-0.070013	-0.074210	-0.0742
Year_2013	-0.103623	-0.084071	-0.030047	-0.074051	-0.065173	-0.072831	1.000000	-0.070347	-0.067797	-0.065173	-0.069080	-0.0690
Year_2014	-0.061572	-0.030129	0.125271	-0.076836	-0.067624	-0.075571	-0.070347	1.000000	-0.070347	-0.067624	-0.071679	-0.0716
Year_2015	0.014842	-0.078449	-0.093514	-0.074051	-0.065173	-0.072831	-0.067797	-0.070347	1.000000	-0.065173	-0.069080	-0.0690
Year_2016	0.065889	-0.039365	0.072950	-0.071185	-0.062651	-0.070013	-0.065173	-0.067624	-0.065173	1.000000	-0.066407	-0.0664
Year_2017	-0.015649	-0.077450	-0.018545	-0.075453	-0.066407	-0.074210	-0.069080	-0.071679	-0.069080	-0.066407	1.000000	-0.0703
Year_2018	-0.007324	-0.082144	-0.050260	-0.075453	-0.066407	-0.074210	-0.069080	-0.071679	-0.069080	-0.066407	-0.070388	1.0000
Year_2019	0.127256	-0.070082	-0.075650	-0.082199	-0.072344	-0.080845	-0.075257	-0.078088	-0.075257	-0.072344	-0.076682	-0.0766
Year_2020	0.077302	0.373219	-0.043912	-0.079550	-0.070013	-0.078240	-0.072831	-0.075571	-0.072831	-0.070013	-0.074210	-0.0742
Year_2021	0.140349	0.166589	-0.035514	-0.071185	-0.062651	-0.070013	-0.065173	-0.067624	-0.065173	-0.062651	-0.066407	-0.0664

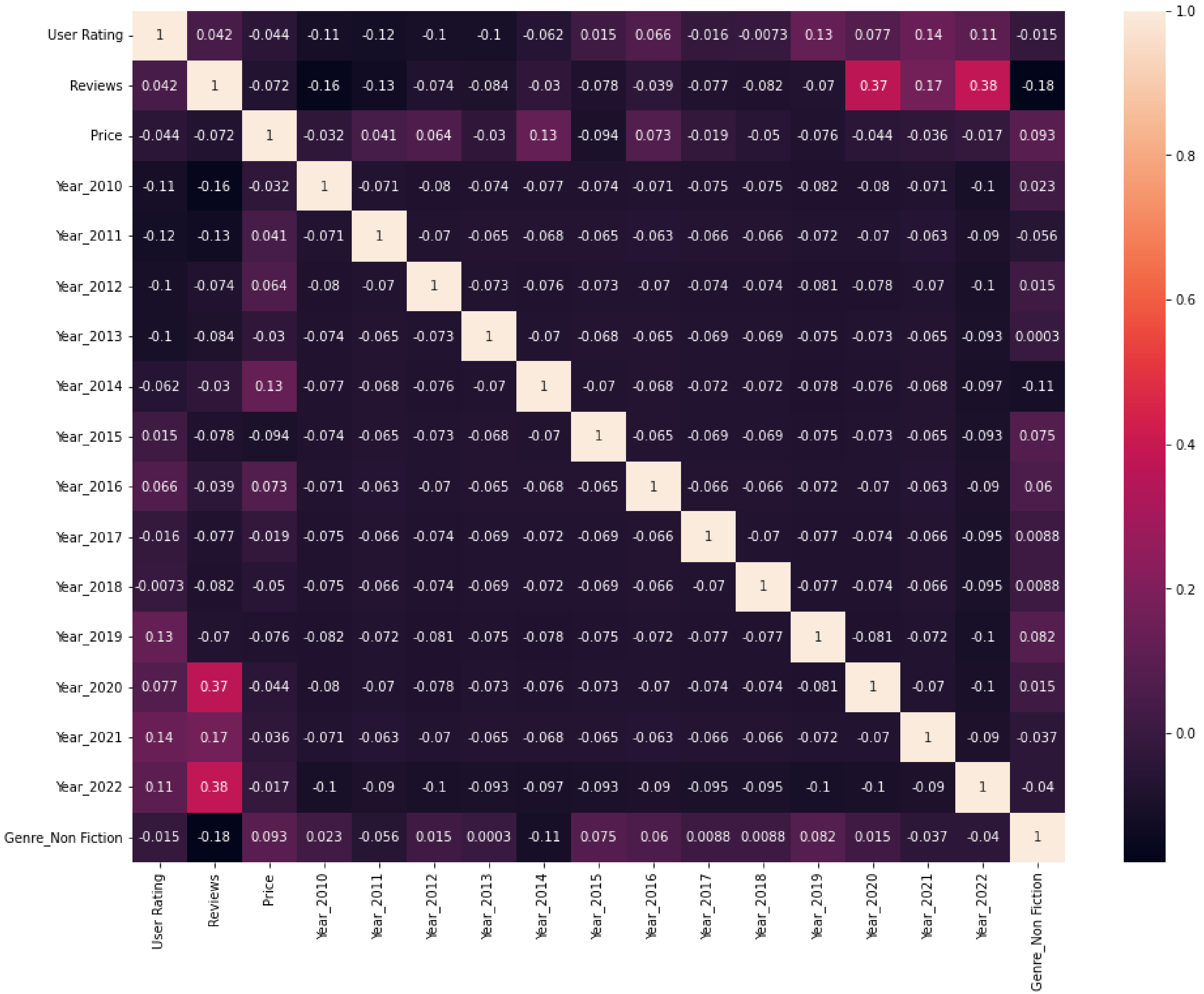
	User Rating	Reviews	Price	Year_2010	Year_2011	Year_2012	Year_2013	Year_2014	Year_2015	Year_2016	Year_2017	Year_201
Year_2022	0.106149	0.378272	-0.016527	-0.101701	-0.089507	-0.100025	-0.093111	-0.096613	-0.093111	-0.089507	-0.094874	-0.0948
Genre_Non Fiction	-0.014784	-0.179138	0.093460	0.023156	-0.056242	0.015340	0.000296	-0.109319	0.074874	0.059568	0.008818	0.0088

In [45]:

```
fig, ax = plt.subplots(figsize=(16, 12))
sns.heatmap(data.corr(),annot=True,ax=ax)
```

Out[45]:

```
<AxesSubplot:>
```


User Rating 1 0.042 -0.044 -0.11 -0.12 -0.1 -0.1 -0.062 0.015 0.066 -0.016 -0.0073 0.13 0.077 0.14 0.11 -0.015
Reviews 0.042 1 -0.072 -0.16 -0.13 -0.074 -0.084 -0.03 -0.078 -0.039 -0.077 -0.082 -0.07 0.37 0.17 0.38 -0.18
Price -0.044 -0.072 1 -0.032 0.041 0.064 -0.03 0.13 -0.094 0.073 -0.019 -0.05 -0.076 -0.044 -0.036 -0.017 0.093
Year_2010 -0.11 -0.16 -0.032 1 -0.071 -0.08 -0.074 -0.077 -0.074 -0.071 -0.075 -0.075 -0.082 -0.08 -0.071 -0.1 0.023
Year_2011 -0.12 -0.13 0.041 -0.071 1 -0.07 -0.065 -0.068 -0.065 -0.063 -0.066 -0.066 -0.072 -0.07 -0.063 -0.09 -0.056
Year_2012 -0.1 -0.074 0.064 -0.08 -0.07 1 -0.073 -0.076 -0.073 -0.07 -0.074 -0.074 -0.081 -0.078 -0.07 -0.1 0.015
Year_2013 -0.1 -0.084 -0.03 -0.074 -0.065 -0.073 1 -0.07 -0.068 -0.065 -0.069 -0.069 -0.075 -0.073 -0.065 -0.093 0.0003
Year_2014 -0.062 -0.03 0.13 -0.077 -0.068 -0.076 -0.07 1 -0.07 -0.068 -0.072 -0.072 -0.078 -0.076 -0.068 -0.097 -0.11
Year_2015 0.015 -0.078 -0.094 -0.074 -0.065 -0.073 -0.068 -0.07 1 -0.065 -0.069 -0.069 -0.075 -0.073 -0.065 -0.093 0.075
Year_2016 0.066 -0.039 0.073 -0.071 -0.063 -0.07 -0.065 -0.068 -0.065 1 -0.066 -0.066 -0.072 -0.07 -0.063 -0.09 0.06
Year_2017 -0.016 -0.077 -0.019 -0.075 -0.066 -0.074 -0.069 -0.072 -0.069 -0.066 1 -0.07 -0.077 -0.074 -0.066 -0.095 0.0088
Year_2018 -0.0073 -0.082 -0.05 -0.075 -0.066 -0.074 -0.069 -0.072 -0.069 -0.066 -0.07 1 -0.077 -0.074 -0.066 -0.095 0.0088
Year_2019 0.13 -0.07 -0.076 -0.082 -0.072 -0.081 -0.075 -0.078 -0.075 -0.072 -0.077 -0.077 1 -0.081 -0.072 -0.1 0.082
Year_2020 0.077 0.37 -0.044 -0.08 -0.07 -0.078 -0.073 -0.076 -0.073 -0.07 -0.074 -0.074 -0.081 1 -0.07 -0.1 0.015
Year_2021 0.14 0.17 -0.036 -0.071 -0.063 -0.07 -0.065 -0.068 -0.065 -0.063 -0.066 -0.066 -0.072 -0.07 1 -0.09 -0.037
Year_2022 0.11 0.38 -0.017 -0.1 -0.09 -0.1 -0.093 -0.097 -0.093 -0.09 -0.095 -0.095 -0.1 -0.1 -0.09 1 -0.04
Genre_Non Fiction -0.015 -0.18 0.093 0.023 -0.056 0.015 0.0003 -0.11 0.075 0.06 0.0088 0.0088 0.082 0.015 -0.037 -0.04 1
User Rating
Reviews
Price
Year_2010
Year_2011
Year_2012
Year_2013
Year_2014
Year_2015
Year_2016
Year_2017
Year_2018
Year_2019
Year_2020
Year_2021
Year_2022
Genre_Non Fiction
1.0
0.8
0.6
0.4
0.2
0.0

In [46]:

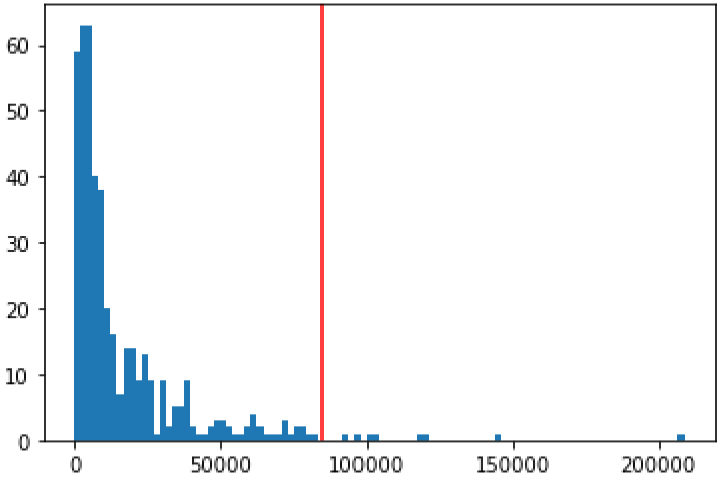
```
data.columns
```

Out[46]:

```
Index(['Name', 'Author', 'User Rating', 'Reviews', 'Price', 'Year_2010',
       'Year_2011', 'Year_2012', 'Year_2013', 'Year_2014', 'Year_2015',
       'Year_2016', 'Year_2017', 'Year_2018', 'Year_2019', 'Year_2020',
       'Year_2021', 'Year_2022', 'Genre_Non Fiction'],
      dtype='object')
```

In [47]:

```
# Lets remove Outliers
plt.hist(data['Reviews'], bins=100)
outlier_limit = (data['Reviews'].mean() + 3*data['Reviews'].std())
plt.axvline(x=outlier_limit, color='r')
plt.show()
```



In [ ]: