EDA Project ON Book Sales & Rating Dataset

In [58]:
```
## Import the required librabries for EDA using python
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

DataSet Link: https://www.kaggle.com/datasets/thedevastator/books-sales-and-ratings

In [2]:
```
df = pd.read_csv("Books_Data_Clean.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	index	Publishing Year	Book Name	Author	language_code	Author_Rating	Book_average_rating	Book_ratings_count	genre	gross sales	p
0	0	1975.0	Beowulf	Unknown, Seamus Heaney	en-US	Novice	3.42	155903	genre fiction	34160.0	
1	1	1987.0	Batman: Year One	Frank Miller, David Mazzucchelli, Richmond Lew...	eng	Intermediate	4.23	145267	genre fiction	12437.5	
2	2	2015.0	Go Set a Watchman	Harper Lee	eng	Novice	3.31	138669	genre fiction	47795.0	
3	3	2008.0	When You Are Engulfed in Flames	David Sedaris	en-US	Intermediate	4.04	150898	fiction	41250.0	
4	4	2011.0	Daughter of Smoke & Bone	Laini Taylor	eng	Intermediate	4.04	198283	genre fiction	37952.5	

In [4]:
```
df.tail()
```

Out[4]:

	index	Publishing Year	Book Name	Author	language_code	Author_Rating	Book_average_rating	Book_ratings_count	genre	gross sales
1065	1065	2014.0	Gray Mountain	John Grisham	eng	Intermediate	3.52	37379	nonfiction	104.94
1066	1066	1989.0	The Power of One	Bryce Courtenay	eng	Excellent	4.34	57312	genre fiction	846.94
1067	1067	1930.0	The Maltese Falcon	Dashiell Hammett	eng	Intermediate	3.92	58742	genre fiction	846.94
1068	1068	2011.0	Night Road	Kristin Hannah	en-US	Excellent	4.17	58028	genre fiction	104.94
1069	1069	1999.0	Tripwire	Lee Child	eng	Excellent	4.07	55251	genre fiction	316.94

In [5]:
```
# Get the shape of dataframe
df.shape
```

Out[5]:
```
(1070, 15)
```

In [6]:
```
#  Get the Size of DataFrame
df.size
```

Out[6]:
```
16050
```

In [7]:
```
# Get the Dimension of DataFrame
df.ndim
```

Out[7]: 2

In [8]:
```
# Get the type of each columns within the dataframe
df.dtypes
```

Out[8]:
```
index                   int64
Publishing Year       float64
Book Name              object
Author                 object
language_code          object
Author_Rating          object
Book_average_rating   float64
Book_ratings_count      int64
genre                  object
gross sales           float64
publisher revenue     float64
sale price            float64
sales rank              int64
Publisher              object
units sold              int64
dtype: object
```

In [9]:
```
# Stastical Summary

df.describe()
```

Out[9]:

	index	Publishing Year	Book_average_rating	Book_ratings_count	gross sales	publisher revenue	sale price	sales rank	units
count	1070.000000	1069.000000	1070.000000	1070.000000	1070.000000	1070.000000	1070.000000	1070.000000	1070.00
mean	534.500000	1971.377923	4.007000	94909.913084	1856.622944	843.281030	4.869561	611.652336	9676.98
std	309.026698	185.080257	0.247244	31513.242518	3936.924240	2257.596743	3.559919	369.849830	15370.57
min	0.000000	-560.000000	2.970000	27308.000000	104.940000	0.000000	0.990000	1.000000	106.00
25%	267.250000	1985.000000	3.850000	70398.000000	372.465000	0.000000	1.990000	287.500000	551.25
50%	534.500000	2003.000000	4.015000	89309.000000	809.745000	273.078000	3.990000	595.500000	3924.00
75%	801.750000	2010.000000	4.170000	113906.500000	1487.957500	721.180500	6.990000	932.500000	5312.25
max	1069.000000	2016.000000	4.770000	206792.000000	47795.000000	28677.000000	33.860000	1273.000000	61560.00

In [10]:
```
# Fix the publishing year as it is having negative value

df=df[df['Publishing Year']>1900]
```

In [11]:
```
df.describe()
```

Out[11]:

	index	Publishing Year	Book_average_rating	Book_ratings_count	gross sales	publisher revenue	sale price	sales rank	units
count	1009.000000	1009.000000	1009.000000	1009.000000	1009.000000	1009.000000	1009.000000	1009.000000	1009.00
mean	535.926660	1994.730426	4.012230	94817.793855	1832.644985	841.360638	4.844311	613.314172	9744.48
std	308.769358	23.204719	0.246492	31473.890412	3947.885096	2279.579848	3.561712	369.628663	15350.02
min	0.000000	1901.000000	2.970000	27308.000000	104.940000	0.000000	0.990000	1.000000	106.00
25%	271.000000	1989.000000	3.860000	70701.000000	366.300000	0.000000	1.990000	291.000000	570.00
50%	535.000000	2003.000000	4.030000	89204.000000	792.000000	273.240000	3.990000	596.000000	3942.00
75%	802.000000	2010.000000	4.180000	113400.000000	1470.260000	714.756000	6.990000	933.000000	5427.00
max	1069.000000	2016.000000	4.770000	206792.000000	47795.000000	28677.000000	33.860000	1273.000000	61560.00

In [12]:

```
# Check for na data

df.isna().sum()
```

Out[12]:

```
index                  0
Publishing Year        0
Book Name             21
Author                 0
language_code         49
Author_Rating          0
Book_average_rating    0
Book_ratings_count     0
genre                  0
gross sales            0
publisher revenue      0
sale price             0
sales rank             0
Publisher              0
units sold             0
dtype: int64
```

In [13]:
```
# Drop the book name that have NA data

df.dropna(subset = 'Book Name')
```

Out[13]:

	index	Publishing Year	Book Name	Author	language_code	Author_Rating	Book_average_rating	Book_ratings_count	genre	
0	0	1975.0	Beowulf	Unknown, Seamus Heaney	en-US	Novice	3.42	155903	genre fiction	341
1	1	1987.0	Batman: Year One	Frank Miller, David Mazzucchelli, Richmond Lew...	eng	Intermediate	4.23	145267	genre fiction	124
2	2	2015.0	Go Set a Watchman	Harper Lee	eng	Novice	3.31	138669	genre fiction	477
3	3	2008.0	When You Are Engulfed in Flames	David Sedaris	en-US	Intermediate	4.04	150898	fiction	412
4	4	2011.0	Daughter of Smoke & Bone	Laini Taylor	eng	Intermediate	4.04	198283	genre fiction	379
...	...	...	...	...	...	...	...	...	...	
1065	1065	2014.0	Gray Mountain	John Grisham	eng	Intermediate	3.52	37379	nonfiction	1
1066	1066	1989.0	The Power of One	Bryce Courtenay	eng	Excellent	4.34	57312	genre fiction	8
1067	1067	1930.0	The Maltese Falcon	Dashiell Hammett	eng	Intermediate	3.92	58742	genre fiction	8
1068	1068	2011.0	Night Road	Kristin Hannah	en-US	Excellent	4.17	58028	genre fiction	1

	index	Publishing Year	Book Name	Author	language_code	Author_Rating	Book_average_rating	Book_ratings_count	genre	
1069	1069	1999.0	Tripwire	Lee Child	eng	Excellent	4.07	55251	genre fiction	3

988 rows × 15 columns

In [14]:
```
# Check for NA Value within the Dataframe
df.isna().sum()
```

Out[14]:
```
index                  0
Publishing Year        0
Book Name             21
Author                 0
language_code         49
Author_Rating          0
Book_average_rating    0
Book_ratings_count     0
genre                  0
gross sales            0
publisher revenue      0
sale price             0
sales rank             0
Publisher              0
units sold             0
dtype: int64
```

In [15]:
```
# Drop the NA Values
df.dropna(subset = 'Book Name', inplace =True)
```

In [16]:
```
df.isna().sum()
```

Out[16]:
```
index                  0
Publishing Year        0
Book Name              0
Author                 0
language_code         47
Author_Rating          0
Book_average_rating    0
Book_ratings_count     0
genre                  0
gross sales            0
publisher revenue      0
sale price             0
sales rank             0
Publisher              0
units sold             0
dtype: int64
```

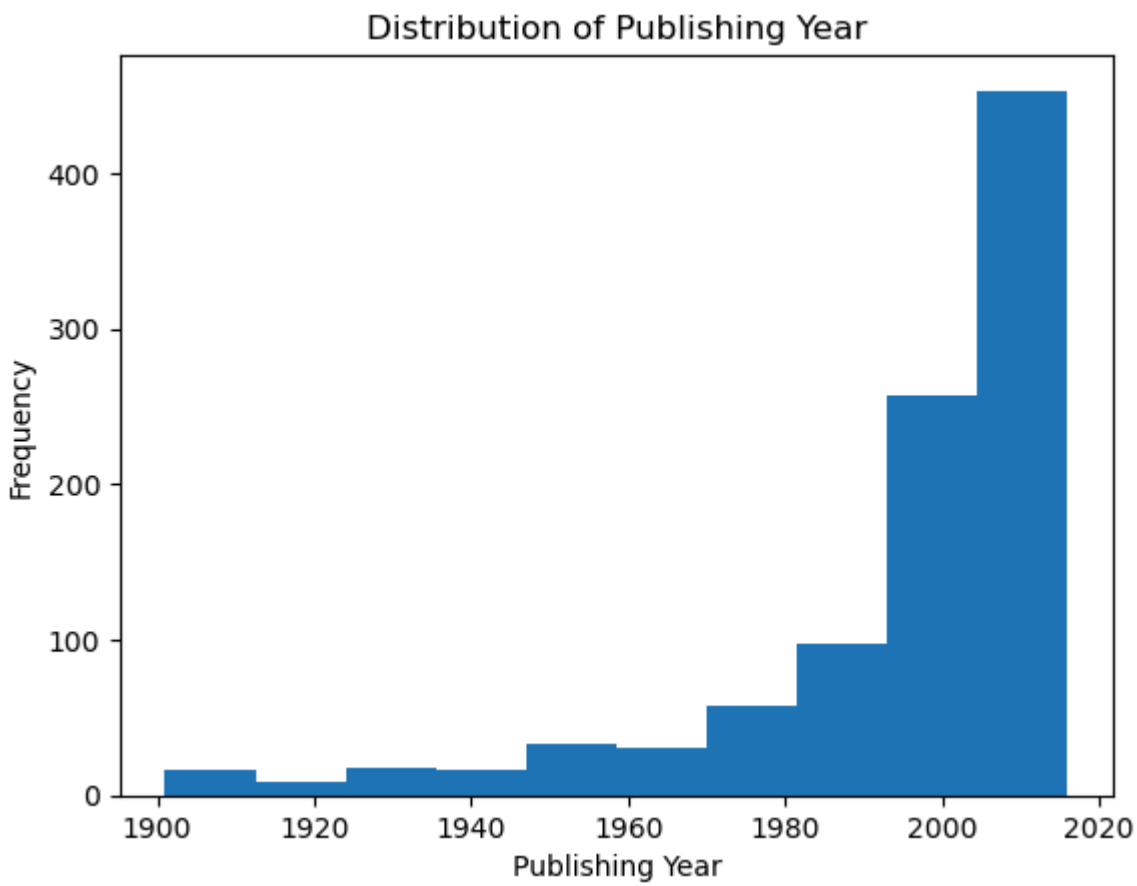
In [17]:
```
#  Check for duplicated data

df.duplicated().sum()
```

Out[17]:
```
0
```

In [18]:
```
# Check the unique entries

df.nunique()
```

Out[18]:
```
index                  988
Publishing Year        101
Book Name              987
Author                 669
language_code            8
Author_Rating            4
Book_average_rating    133
Book_ratings_count     983
genre                    4
gross sales            774
publisher revenue      570
sale price             143
sales rank             818
Publisher                9
units sold             470
dtype: int64
```

In [19]:
```
# Create a histogram for the publishing year

plt.hist(df["Publishing Year"])
plt.xlabel("Publishing Year")
plt.ylabel("Frequency")
plt.title("Distribution of Publishing Year")
plt.show()
```

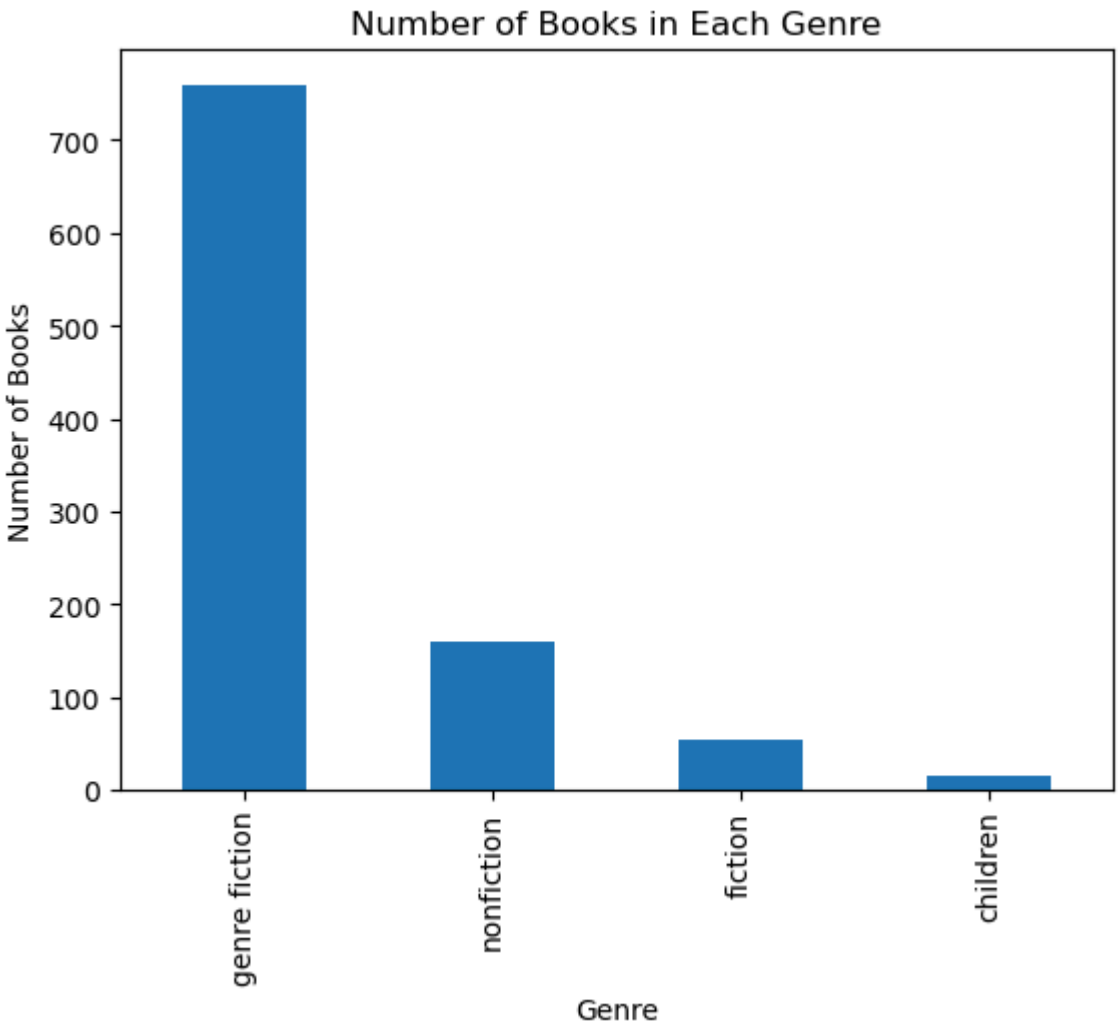


In [20]:
```
# Create a Bar chart of book in each genre

df["genre"].value_counts()
```

Out[20]:
```
genre
genre fiction    759
nonfiction       160
fiction           54
children          15
Name: count, dtype: int64
```

In [21]:
```
# Get the number of books in each genre
df["genre"].value_counts().plot(kind = 'bar')
```

```
plt.xlabel("Genre")
plt.ylabel("Number of Books")
plt.title("Number of Books in Each Genre")
plt.show()
```



In [23]:
```
# Perform the GroupBy

df.groupby("Author")["Book_average_rating"].mean()
```

Out[23]:
```
Author
A.A. Milne, Ernest H. Shepard                        4.36
A.S.A. Harrison                                      3.30
Abbi Glines                                          4.21
Adam Johnson                                         4.06
Adam Mansbach, Ricardo CortÃ©s                       4.26
                                                     ...
William Styron                                       4.18
Yana Toboso, Tomo Kimura                             4.37
Zadie Smith                                          3.75
Ø£ØÙ„Ø§Ù… Ù…Ø³ØªØºØ§Ù†Ù…ÙŠ, Ahlam Mosteghanemi    3.72
Ø£ØÙ…Ø¯ Ù…Ø±Ø§Ø¯                                     3.80
Name: Book_average_rating, Length: 669, dtype: float64
```

In [26]:
```
type(df.groupby("Author")["Book_average_rating"].mean())
```

Out[26]:
```
pandas.core.series.Series
```

In [27]:
```
df.groupby("Author")["Book_average_rating"].mean().reset_index()
```

Out[27]:

	Author	Book_average_rating
0	A.A. Milne, Ernest H. Shepard	4.36
1	A.S.A. Harrison	3.30
2	Abbi Glines	4.21
3	Adam Johnson	4.06
4	Adam Mansbach, Ricardo CortÃ©s	4.26
...	...	...
664	William Styron	4.18
665	Yana Toboso, Tomo Kimura	4.37
666	Zadie Smith	3.75
667	Ø£ØÙ„Ø§Ù… Ù…Ø³Øªø°Ø§Ù†Ù…ÙŠ, Ahlam Mosteghanemi	3.72
668	Ø£ØÙ…Ø¯ Ù…Ø±Ø§Ø¯	3.80

669 rows × 2 columns

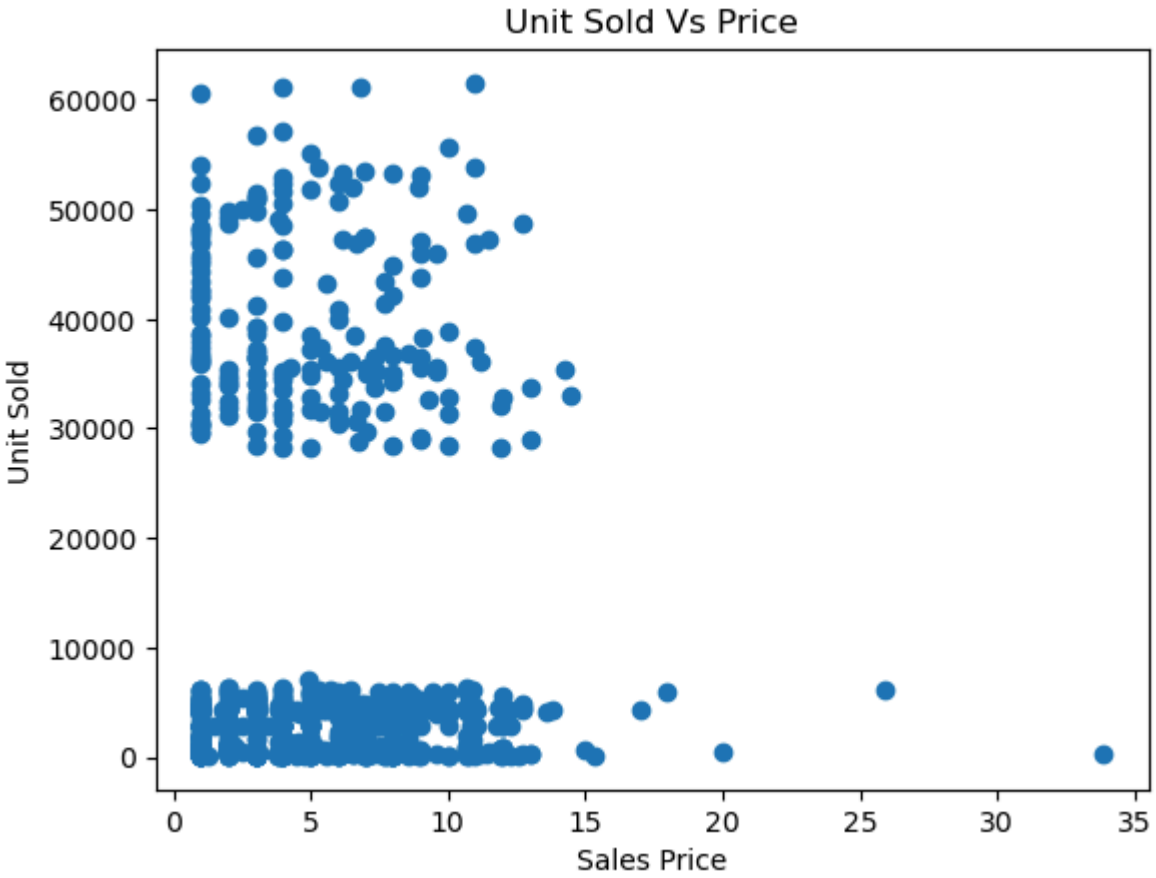
In [29]:
```
df.groupby("Author")["Book_average_rating"].mean().sort_values(ascending = False)
```

Out[29]:
```
Author
Bill Watterson                  4.650000
Bill Watterson, G.B. Trudeau    4.610000
J.R.R. Tolkien                  4.590000
George R.R. Martin              4.560000
Sarah J. Maas                   4.526000
                                   ...
Chetan Bhagat                   3.273333
Audrey Niffenegger              3.230000
Herman Koch, Sam Garrett        3.220000
P.D. James                      3.210000
Sue Monk Kidd                   3.100000
Name: Book_average_rating, Length: 669, dtype: float64
```

In [32]:
```
# Book ratings in each genre
sns.boxplot(x="genre", y = "Book_ratings_count", data = df, color = "red")
plt.xlabel("Genre")
plt.ylabel("Books Rating Count")
plt.title("Box Plot for book rating in each genre")
plt.show()
```

In [36]:
```
# Scatter plot showing the relation between Unit sold and price of the book
plt.scatter(df["sale price"], df["units sold"])
plt.xlabel("Sales Price")
plt.ylabel("Unit Sold")
plt.title("Unit Sold Vs Price")
```

Out[36]: Text(0.5, 1.0, 'Unit Sold Vs Price')



In [37]:
```
df.columns
```

Out[37]:
```
Index(['index', 'Publishing Year', 'Book Name', 'Author', 'language_code',
       'Author_Rating', 'Book_average_rating', 'Book_ratings_count', 'genre',
       'gross sales', 'publisher revenue', 'sale price', 'sales rank',
       'Publisher ', 'units sold'],
      dtype='object')
```

In [22]:
```
df.head(2)
```

Out[22]:

	index	Publishing Year	Book Name	Author	language_code	Author_Rating	Book_average_rating	Book_ratings_count	genre	gross sales	pub re
0	0	1975.0	Beowulf	Unknown, Seamus Heaney	en-US	Novice	3.42	155903	genre fiction	34160.0	2
1	1	1987.0	Batman: Year One	Frank Miller, David Mazzucchelli, Richmond Lew...	eng	Intermediate	4.23	145267	genre fiction	12437.5	

In [38]:

```
# Get the total number of languages in which book is written

language_counts = df["language_code"].value_counts()
```

In [39]:

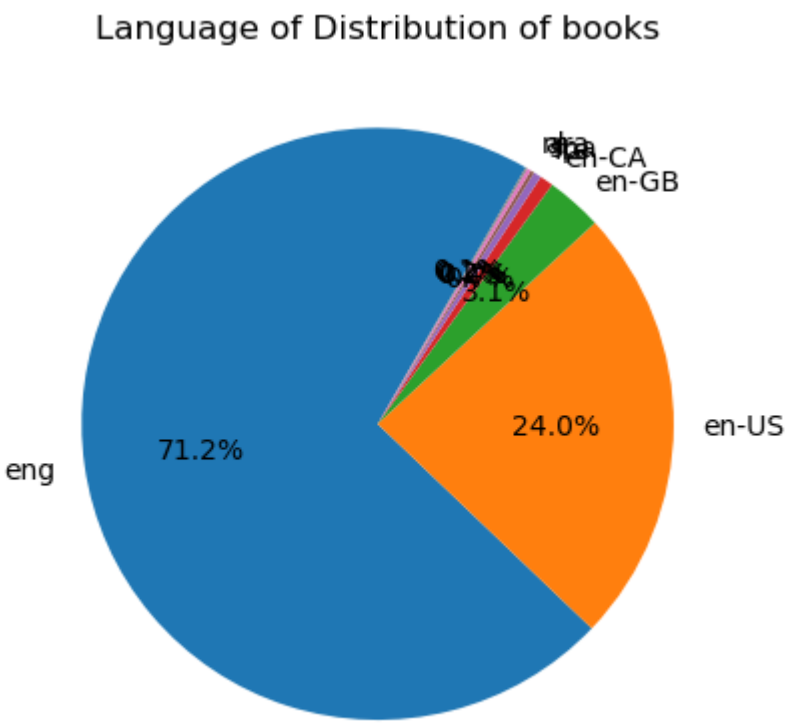
```
language_counts
```

Out[39]:

```
language_code
eng      670
en-US    226
en-GB     29
en-CA      7
fre        4
spa        2
ara        2
nl         1
Name: count, dtype: int64
```

In [43]:

```
# Represent the Distribution of books with respect to their language

plt.pie(language_counts,  labels= language_counts.index,
        startangle= 60, autopct="%1.1f%%")
plt.title('Language of Distribution of books')
plt.show()
```



In [46]:
```
# Get the Publisher's revenue in Descending order

df.groupby('Publisher ')['publisher revenue'].sum().sort_values(ascending= False)
```

Out[46]:
```
Publisher
Penguin Group (USA) LLC                191581.104
Random House LLC                       174956.244
Amazon Digital Services,  Inc.         141767.772
HarperCollins Publishers               121769.814
Hachette Book Group                    107410.968
Simon and Schuster Digital Sales Inc    46858.206
Macmillan                               31249.830
HarperCollins Publishing                 2830.806
HarperCollins Christian Publishing       2135.670
Name: publisher revenue, dtype: float64
```

In [47]:
```
# get the max rating

df.groupby('Author_Rating')['Book_ratings_count'].mean().max()
```

Out[47]:
```
101400.27256944444
```

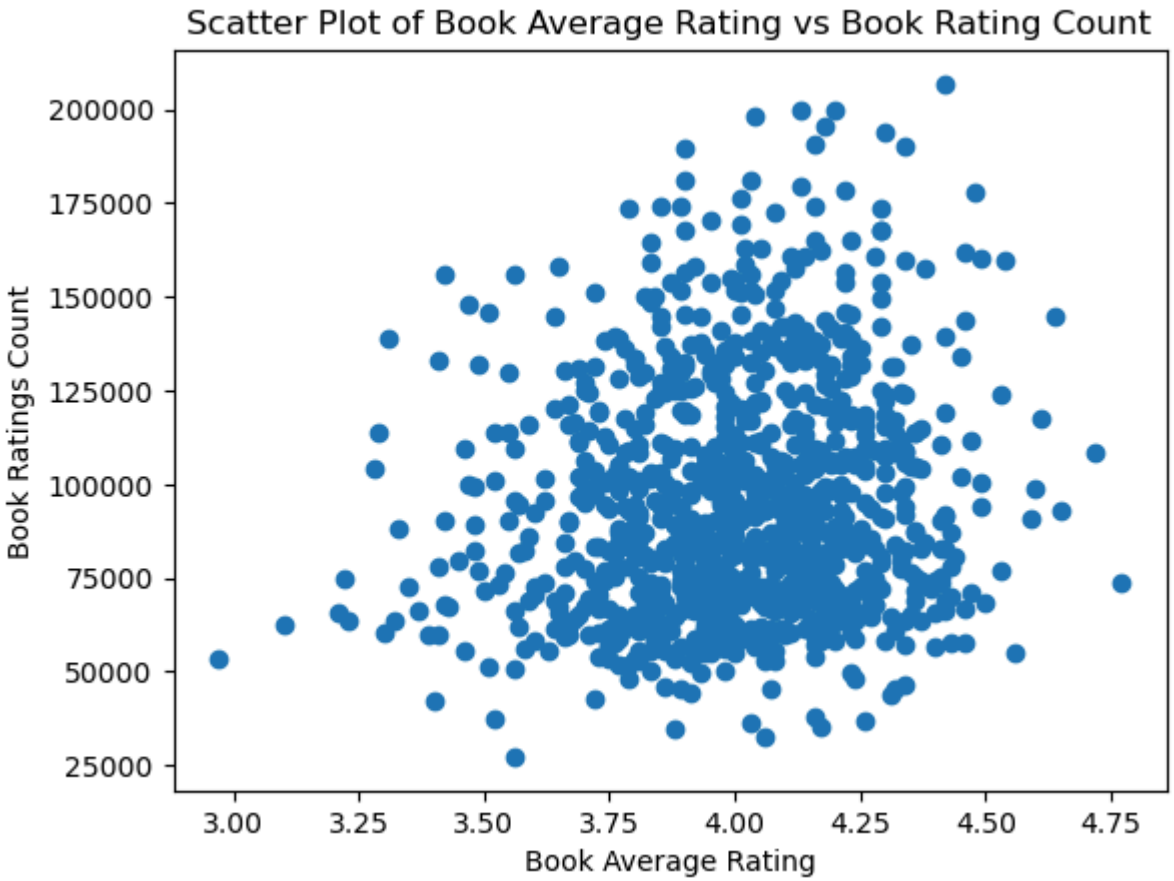
In [48]:
```
df.groupby('language_code').size().sort_values(ascending=False)
```

Out[48]:
```
language_code
eng      670
en-US    226
en-GB     29
en-CA      7
fre        4
ara        2
spa        2
nl         1
dtype: int64
```

In [49]:
```
df.groupby('Author_Rating')['Book_ratings_count'].var()
```

Out[49]:
```
Author_Rating
Excellent       4.419857e+08
Famous          1.227555e+09
Intermediate    1.170331e+09
Novice          9.523157e+08
Name: Book_ratings_count, dtype: float64
```

In [50]:
```
# Averege book rating vs Book Rating Count

plt.scatter(df['Book_average_rating'], df['Book_ratings_count'])
plt.xlabel('Book Average Rating')
plt.ylabel('Book Ratings Count')
plt.title('Scatter Plot of Book Average Rating vs Book Rating Count')
plt.show()
```



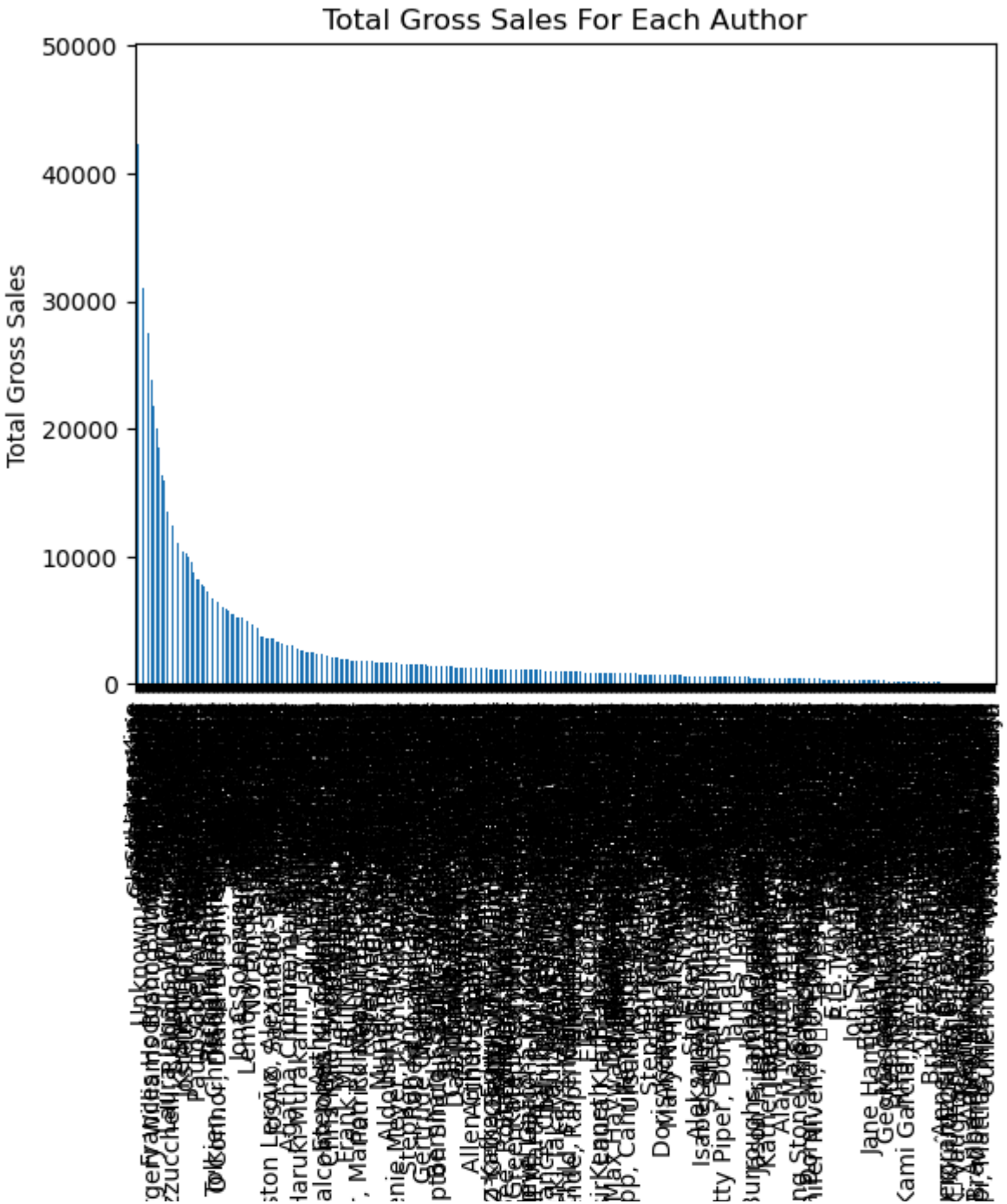
In [51]:
```
# Get the total sales
total_gross_sales_by_author = df.groupby('Author')['gross sales'].sum().sort_values(ascending=False)
```

In [52]:
```
total_gross_sales_by_author
```

Out[52]:
```
Author
Harper Lee              47795.00
Stephen King            43322.65
David Sedaris           42323.41
Charlaine Harris        39453.08
Laini Taylor            38278.41
                          ...
Frank Warren              107.91
Ayaan Hirsi Ali           107.91
Walter M. Miller Jr.      106.92
Michael Shaara            105.93
Blake Crouch              105.93
Name: gross sales, Length: 669, dtype: float64
```

In [54]:
```
total_gross_sales_by_author.plot(kind='bar')
plt.xlabel('Author')
plt.ylabel('Total Gross Sales')
plt.title('Total Gross Sales For Each Author')
plt.show()
```

```
C:\Users\sanad\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 141 (\x8d) missing from font(s) D
ejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\sanad\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 144 (\x90) missing from font(s) D
ejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\sanad\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 129 (\x81) missing from font(s) D
ejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```


Total Gross Sales For Each Author
Total Gross Sales
50000
40000
30000
20000
10000
0

Mar
Frank Miller, David Maz

Mikhail Bulgakov, Katherine Tiernan

Gas
H
Neil Gaiman, Sam Kieth, Mike Dringenberg, M
Irma S. Rombauer
Stephe
U

obson, Tom Johnson, Rob Kutner, Chris Regan, Jason Reich, Jason Ross, Stephen Colbert, Samantha B
Bill Willingham, Lan Medina, Sa
Astrid Lindgren, Lauren E
verman, Michael Brumm, Eric Drydale, Rob Dubbin, Glenn Eichler, Peter Grosz, Peter Gwinn, Jay Kats
Barbara Kingsolver, Steven L. Ho

Wa
William S. B

Neil Gaiman, Kelley Jones, Charles Vess, Colle

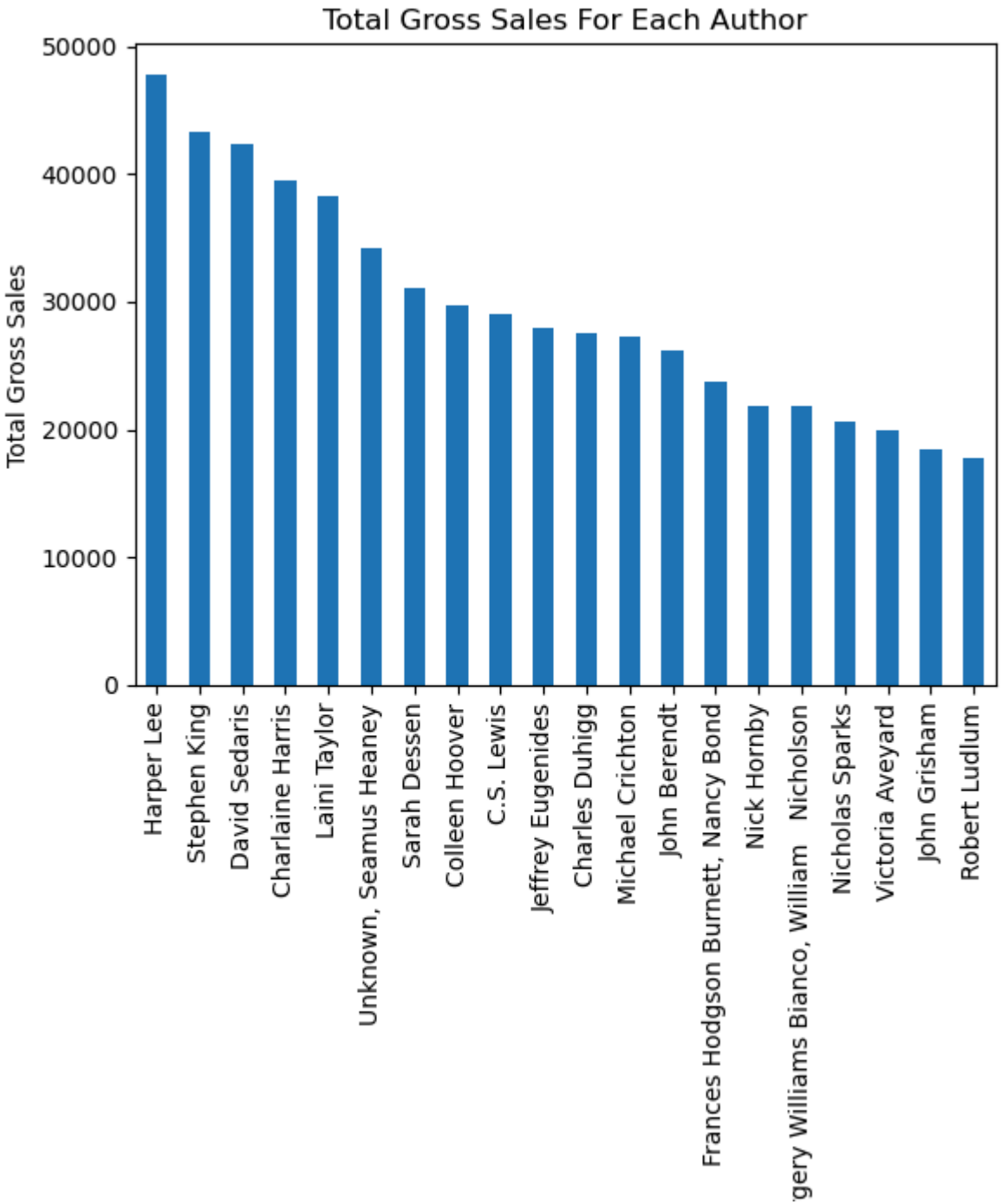
I

Elif Shafak, Ø¥Ù„ÙŠÙ Ø´Ø§Ù Ø§Ù,, Ø§Ø±Ø

n, David Javerbaum, Rich Bloomquist, Steve Bodow, Tim Carvell, Eric Drysdale, J.R. Havlan, Scott Jac

Stephen Colbert, Richard Dahm, Paul Dinello, Allison Sil

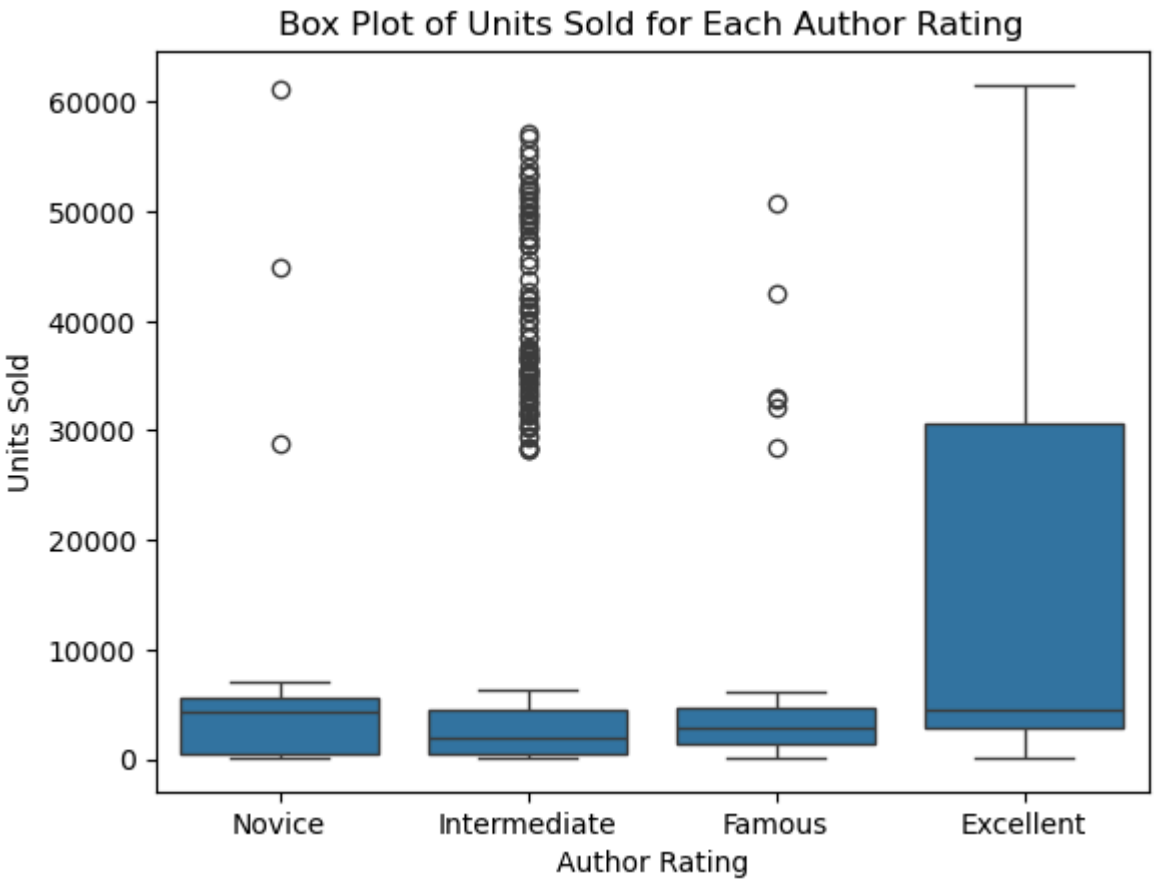
In [55]:
```
# Get the total sales for each author

total_gross_sales_by_author.head(20).plot(kind='bar')
plt.xlabel('Author')
plt.ylabel('Total Gross Sales')
plt.title('Total Gross Sales For Each Author')
plt.show()
```


Total Gross Sales For Each Author
Total Gross Sales
50000
40000
30000
20000
10000
0
Harper Lee
Stephen King
David Sedaris
Charlaine Harris
Laini Taylor
Unknown, Seamus Heaney
Sarah Dessen
Colleen Hoover
C.S. Lewis
Jeffrey Eugenides
Charles Duhigg
Michael Crichton
John Berendt
Frances Hodgson Burnett, Nancy Bond
Nick Hornby
gery Williams Bianco, William Nicholson
Nicholas Sparks
Victoria Aveyard
John Grisham
Robert Ludlum

Mar

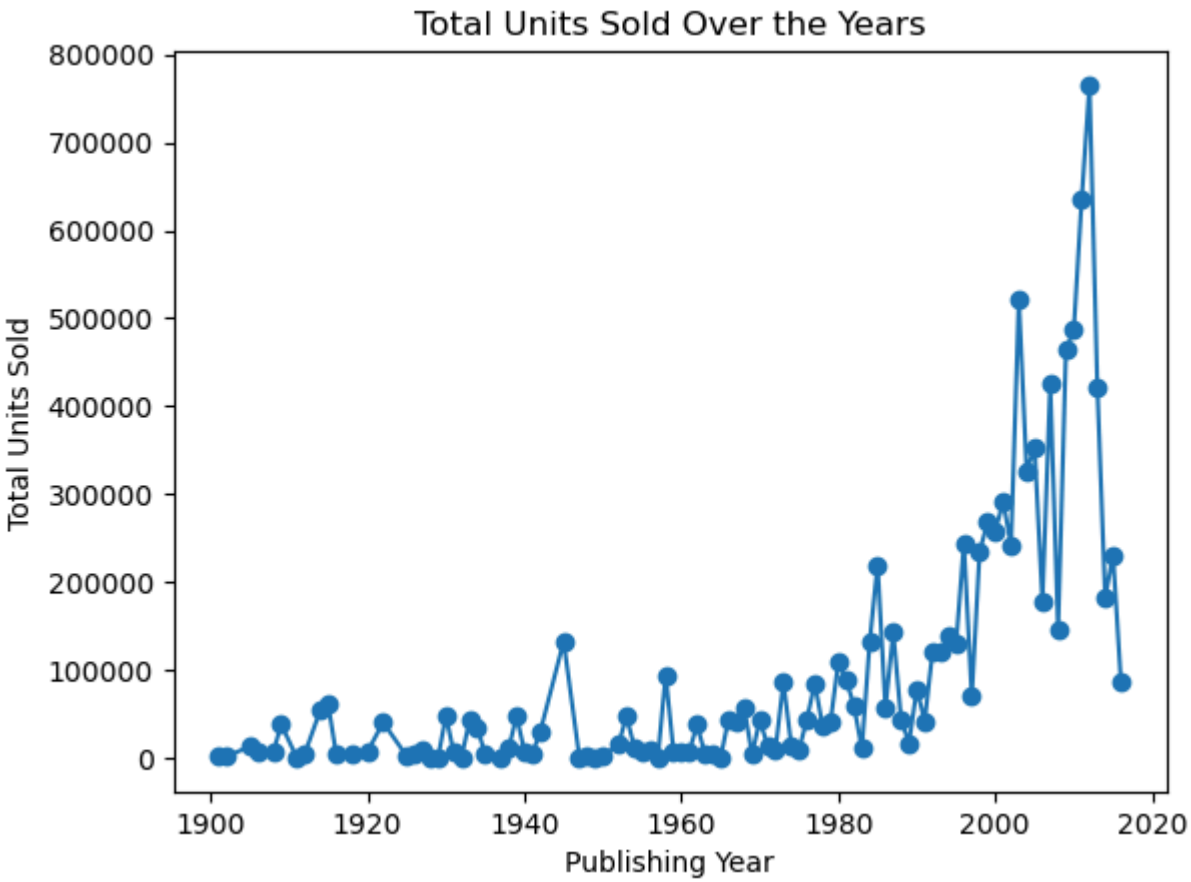
Author

In [56]:
```
# Get the unit sold for each author rating
sns.boxplot(x='Author_Rating', y= 'units sold', data=df)
plt.xlabel('Author Rating')
plt.ylabel('Units Sold')
plt.title('Box Plot of Units Sold for Each Author Rating')
plt.show()
```



In [57]:
```
# Get the total unit sold over the years
df.groupby('Publishing Year')['units sold'].sum().plot(kind='line', marker='o')
```

```
plt.xlabel('Publishing Year')
plt.ylabel('Total Units Sold')
plt.title('Total Units Sold Over the Years')
plt.show()
```



In [ ]: