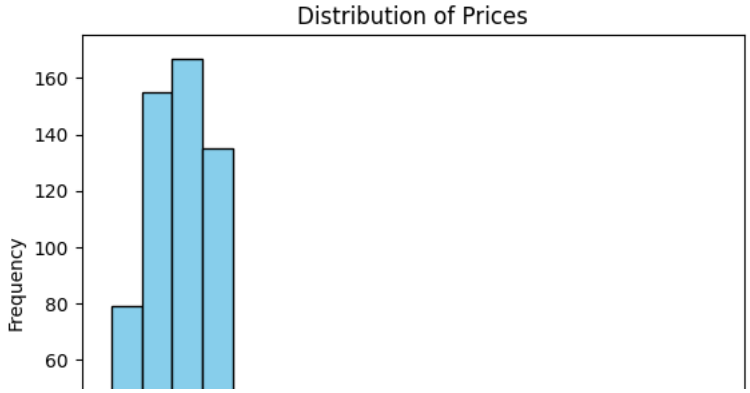
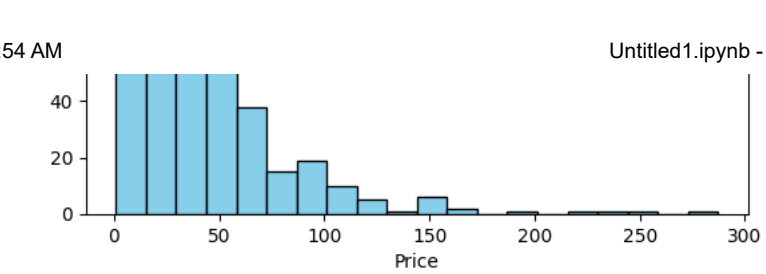
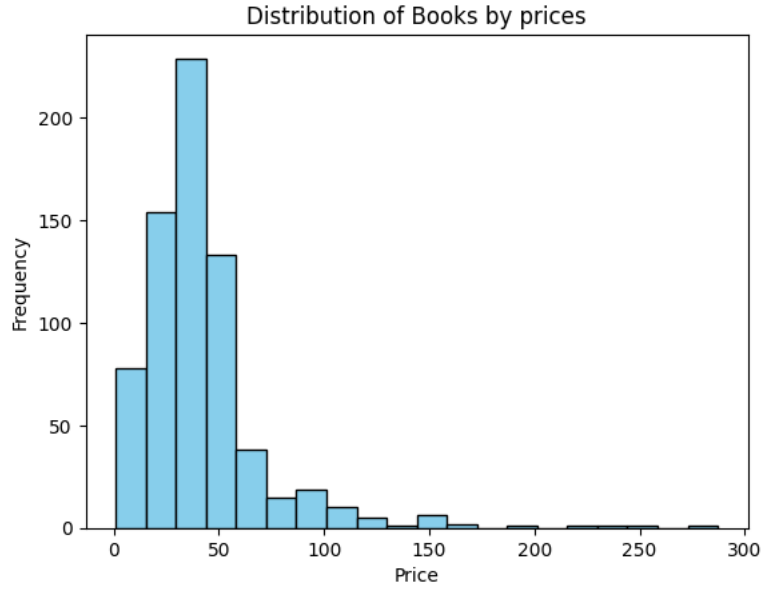
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Sav
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
data = pd.read_csv("/content/final_book_dataset_kaggle2.csv")
Amazon_books = pd.DataFrame(data)
print(Amazon_books.head())
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',100)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
print(Amazon_books.columns)
print(Amazon_books.shape)
print(Amazon_books.isna().sum())
Amazon_books.dropna(subset='n_reviews',inplace=True)
import matplotlib.pyplot as plt
plt.hist(Amazon_books['price'], bins=20, color='skyblue', edgecolor='black')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.title('Distribution of Prices')
plt.show()
median = Amazon_books['price'].median()
Amazon_books['price'].fillna(median, inplace=True)
Amazon_books.isna().sum()
Amazon_books.info()
Amazon_books['n_reviews'] = Amazon_books['n_reviews'].str.replace(',', '').astype(float)
Amazon_books['avg_reviews'] = Amazon_books['avg_reviews'].astype(float)
non_numeric_rows = Amazon_books[Amazon_books['pages'].str.replace('.', '', 1).str.isnumeric() == False]
Amazon_books = Amazon_books.drop(non_numeric_rows.index)
Amazon_books['pages'] = Amazon_books['pages'].astype(float)
import matplotlib.pyplot as plt
plt.hist(Amazon_books['price'], bins=20, color='skyblue', edgecolor='black')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.title('Distribution of Books by prices')
plt.show()
print(Amazon_books['price'].mean())
import matplotlib.pyplot as plt
plt.scatter(Amazon_books['avg_reviews'],Amazon_books['price'])
plt.ylabel('Price')
plt.xlabel('Average Reviews')
plt.title('Scatter Plot of Price vs. Average Reviews')
coef = np.polyfit(Amazon_books['avg_reviews'],Amazon_books['price'],1)
trendline = np.poly1d(coef)
plt.plot(Amazon_books['avg_reviews'],trendline(Amazon_books['avg_reviews']),"r--")
plt.show()
# Scatter plot
plt.scatter(Amazon_books['pages'], Amazon_books['price'])
plt.ylabel('price')
plt.xlabel('Number of pages')
plt.title('Scatter Plot of Number of pages vs. price')
plt.show()
import seaborn as sns
import matplotlib.pyplot as plt  # Import matplotlib for customization

# Calculate the correlation matrix
correlation_matrix = Amazon_books[['pages', 'price']].corr()

# Create a heatmap with labels
plt.figure(figsize=(8, 6))  # Set the figure size
sns.heatmap(
    correlation_matrix,
    annot=True,
    fmt=".2f",
    cbar=True,
    square=True,
)
plt.title('Correlation Heatmap')  # Add a title to the plot
plt.show()
```

```
                                               title  \
0  Data Analysis Using R (Low Priced Edition): A ...
1  Head First Data Analysis: A learner's guide to...
2  Guerrilla Data Analysis Using Microsoft Excel:...
3  Python for Data Analysis: Data Wrangling with ...
4  Excel Data Analysis For Dummies (For Dummies (...

                            author  price price (including used books)  \
0                [ Dr Dhaval Maheta]   6.75                        6.75
1                              NaN  33.72               21.49 - 33.72
2  [ Oz du Soleil,  and , Bill Jelen]  32.07                       32.07
3                [ William McKinney]  53.99                       53.99
4                  [ Paul McFedries]  24.49                       24.49

  pages  avg_reviews n_reviews star5 star4 star3 star2 star1  \
0   500          4.4        23   55%   39%    6%   NaN   NaN
1   484          4.3       124   61%   20%    9%    4%    6%
2   274          4.7        10   87%   13%   NaN   NaN   NaN
3   547          4.6     1,686   75%   16%    5%    2%    2%
4   368          3.9        12   52%   17%   10%   10%   10%

                dimensions       weight language  \
0    8.5 x 1.01 x 11 inches  2.53 pounds  English
1    8 x 0.98 x 9.25 inches  1.96 pounds  English
2  8.25 x 0.6 x 10.75 inches   1.4 pounds  English
3    7 x 1.11 x 9.19 inches  1.47 pounds  English
4  7.38 x 0.83 x 9.25 inches   1.3 pounds  English

                                       publisher         ISBN_13  \
0    Notion Press Media Pvt Ltd (November 22, 2021)  978-1685549596
1     O'Reilly Media; 1st edition (August 18, 2009)  978-0596153939
2  Holy Macro! Books; Third edition (August 1, 2022)  978-1615470747
3   O'Reilly Media; 2nd edition (November 14, 2017)  978-1491957660
4       For Dummies; 5th edition (February 3, 2022)  978-1119844426

                                             link  \
0  /Data-Analysis-Using-Low-Priced/dp/1685549594/...
1  /Head-First-Data-Analysis-statistics/dp/059615...
2  /Guerrilla-Analysis-Using-Microsoft-Excel/dp/1...
3  /Python-Data-Analysis-Wrangling-IPython/dp/149...
4  /Excel-Data-Analysis-Dummies-Computer/dp/11198...

                                   complete_link
0  https://www.amazon.com/Data-Analysis-Using-Low...
1  https://www.amazon.com/Head-First-Data-Analysi...
2  https://www.amazon.com/Guerrilla-Analysis-Usin...
3  https://www.amazon.com/Python-Data-Analysis-Wr...
4  https://www.amazon.com/Excel-Data-Analysis-Dum...
Index(['title', 'author', 'price', 'price (including used books)', 'pages', 'avg_reviews',
       'n_reviews', 'star5', 'star4', 'star3', 'star2', 'star1', 'dimensions', 'weight',
       'language', 'publisher', 'ISBN_13', 'link', 'complete_link'],
      dtype='object')
(830, 19)
title                            0
author                         173
price                          108
price (including used books)   108
pages                           85
avg_reviews                    128
n_reviews                      128
star5                          128
star4                          195
star3                          276
star2                          379
star1                          502
dimensions                     186
weight                         179
language                        71
publisher                      116
ISBN_13                        165
link                             0
complete_link                    0
dtype: int64
```

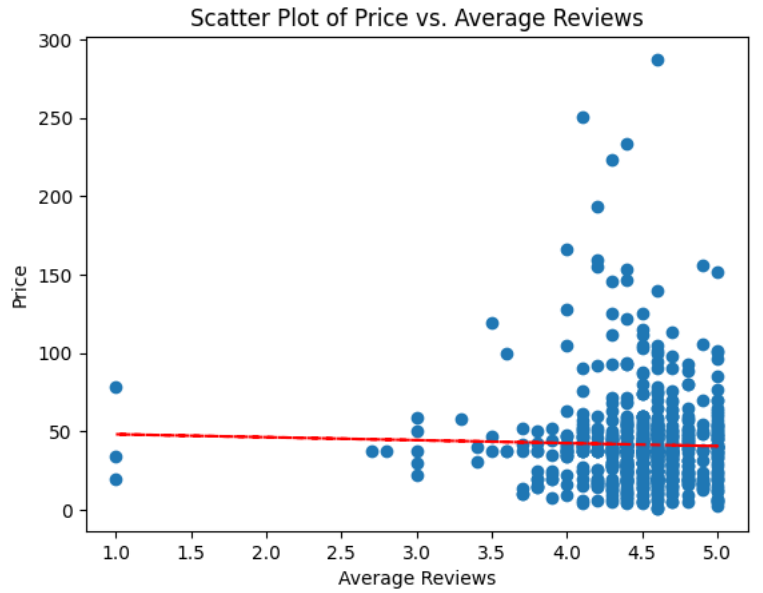




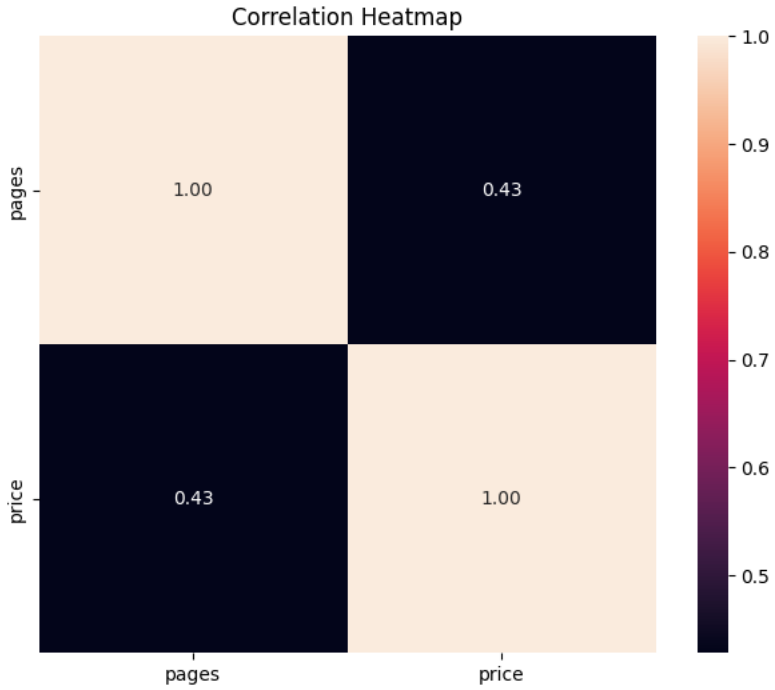

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 702 entries, 0 to 829
Data columns (total 19 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   title                          702 non-null    object
 1   author                         560 non-null    object
 2   price                          702 non-null    float64
 3   price (including used books)   637 non-null    object
 4   pages                          658 non-null    object
 5   avg_reviews                    702 non-null    float64
 6   n_reviews                      702 non-null    object
 7   star5                          702 non-null    object
 8   star4                          635 non-null    object
 9   star3                          554 non-null    object
 10  star2                          451 non-null    object
 11  star1                          328 non-null    object
 12  dimensions                     593 non-null    object
 13  weight                         584 non-null    object
 14  language                       665 non-null    object
 15  publisher                      628 non-null    object
 16  ISBN_13                        597 non-null    object
 17  link                           702 non-null    object
 18  complete_link                  702 non-null    object
dtypes: float64(2), object(17)
memory usage: 109.7+ KB
<ipython-input-1-44f8016adf56>:38: FutureWarning: The default value of regex will change from True to False in a future version.
  non_numeric_rows = Amazon_books[Amazon_books['pages'].str.replace('.', '', 1).str.isnumeric() == False]
```




```
41.6778417266187
```




Scatter Plot of Number of pages vs. price
price
Number of pages
0
50
100
150
200
250
300
0
250
500
750
1000
1250
1500


Correlation Heatmap
pages
price
1.00
0.43
0.43
1.00
pages
price
1.0
0.9
0.8
0.7
0.6
0.5