In [ ]:
```
import pandas as pd
import numpy as np
```

In [ ]:
```
df = pd.read_csv('goodreads.csv')
df.head()
```

Out[ ]:

4.40	**136455**	**0439023483**	**good_reads:book**	**https://www.goodreads.com/author/show/1**	
0	4.41	16648.0	0439358078	good_reads:book	https://www.goodreads.cor
1	3.56	85746.0	0316015849	good_reads:book	https://www.goodreads.cor
2	4.23	47906.0	0061120081	good_reads:book	https://www.goodreads.cor
3	4.23	34772.0	0679783261	good_reads:book	https://www.goodreads.cor
4	4.25	12363.0	0446675539	good_reads:book	https://www.goodreads.cor

In [ ]:
```
# As the dataset has no column names so better to give the names
columns = ['Rating',
           'Publication Cost',
           'ISBN',
           'Good Reads',
           'Book URL',
           'Year',
           'Genres',
           'Link',
           'Total Revenue',
           'Book Title']
df = pd.read_csv('goodreads.csv', names=columns)
```

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

Out[ ]:

	Rating	Publication Cost	ISBN	Good Reads	
0	4.40	136455.0	0439023483	good_reads:book	https://www.goodreads.com/author/show/1
1	4.41	16648.0	0439358078	good_reads:book	https://www.goodreads.com/author/show/1
2	3.56	85746.0	0316015849	good_reads:book	https://www.goodreads.com/author/show/9
3	4.23	47906.0	0061120081	good_reads:book	https://www.goodreads.com/author/show/
4	4.23	34772.0	0679783261	good_reads:book	https://www.goodreads.com/author/show/1

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 10 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Rating            5998 non-null   float64
 1   Publication Cost  5998 non-null   float64
 2   ISBN              5523 non-null   object 
 3   Good Reads        5998 non-null   object 
 4   Book URL          5998 non-null   object 
 5   Year              5993 non-null   float64
 6   Genres            5938 non-null   object 
 7   Link              6000 non-null   object 
 8   Total Revenue     5998 non-null   float64
 9   Book Title        5998 non-null   object 
dtypes: float64(4), object(6)
memory usage: 468.9+ KB
```

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

Out[ ]:
```
Rating               2
Publication Cost     2
ISBN               477
Good Reads           2
Book URL             2
Year                 7
Genres              62
Link                 0
Total Revenue        2
Book Title           2
dtype: int64
```

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

Out[ ]:
```
558
```

Rating Column

In [ ]:
```
df['Rating'].isna().sum()
```

Out[ ]:
```
2
```

In [ ]:
```
df['Rating'].min(), df['Rating'].max()
```

Out[ ]:
```
(2.0, 5.0)
```

In [ ]:
```
# Replacing or Filling the Null values with the Mean  of that Column
df['Rating'].fillna(df['Rating'].mean(), inplace=True)
```

Publication Cost

In [ ]:
```
df['Publication Cost'].isna().sum()
```

Out[ ]:
```
2
```

In [ ]:
```
# Replacing or Filling the Null values with the Mean  of the Column
df['Publication Cost'].fillna(df['Publication Cost'].mean(), inplace=True)
```

ISBN Column

In [ ]:
```
df['ISBN'].isna().sum()
```

Out[ ]:
```
477
```

In [ ]:
```
# As the Datatype of 'ISBN' is Object so it can't perform mean
# so just replacing ISBN with 'Unknown'
df['ISBN'].fillna('Unknown', inplace=True)
```

Good Reads Column

In [ ]:
```
df['Good Reads'].unique()
```

Out[ ]:
```
array(['good_reads:book', nan], dtype=object)
```

In [ ]:
```
# No need of this column
df.drop(columns=['Good Reads'], inplace=True)
```

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

Out[ ]:

	Rating	Publication Cost	ISBN	Book URL	Year
0	4.40	136455.0	0439023483	https://www.goodreads.com/author/show/153394.S...	2008.0
1	4.41	16648.0	0439358078	https://www.goodreads.com/author/show/1077326....	2003.0

Book URL Column

In [ ]:
```
df['Book URL'].isna().sum()
```

Out[ ]:
```
2
```

In [ ]:
```
# As the Datatype of 'Book URL' is Object so it can't perform mean
# so just replacing Book URL with 'Unknown'
df['Book URL'].fillna('Unknown', inplace=True)
```

Year Column

In [ ]:
```
df['Year'].isna().sum()
```

Out[ ]:
```
7
```

In [ ]:
```
# Replacing or Filling the Null values with the Mean  of the Column
df['Year'].fillna(df['Year'].mean(), inplace=True)
```

In [ ]:
```
# Changing the datatype of Year from float to int
df['Year'] = df['Year'].astype('int')
```

Genres Column

In [ ]:
```
df['Genres'].isna().sum()
```

Out[ ]:
```
62
```

In [ ]:
```
# As Genres is also a object and we will perform some patterns
# so it's better to replace Null values with the any of the row value
# Here I'm taking the First Value
first = df['Genres'].iloc[0]
df['Genres'].fillna(first, inplace=True)
```

In [ ]:
```
# Here the logic comes to extract the genres from each df['Genres] row
def extract_genres(string):
    genre_list = [g.split('/')[-1] for g in string.split('|')]
    return ', '.join(genre_list)
```

In [ ]:
```
# Now just applying the logic to the each data of the column of df['Genres]
df['Genres'] = df['Genres'].apply(extract_genres)
```

In [ ]:
```
df['Genres'].head(4)
```

Out[ ]:
```
0    young-adult, science-fiction, dystopia, fantas...
1    fantasy, young-adult, fiction, fantasy, magic,...
2    young-adult, fantasy, romance, paranormal, vam...
3    classics, fiction, historical-fiction, academi...
Name: Genres, dtype: object
```

Total Revenue Column

In [ ]:
```
df['Total Revenue'].isna().sum()
```

Out[ ]:
```
2
```

In [ ]:
```
# Replacing or Filling the Null values with the Mean  of the Column
df['Total Revenue'].fillna(df['Total Revenue'].mean(), inplace=True)
```

Book Titile Column

In [ ]:
```
df['Book Title'].isna().sum()
```

Out[ ]:
```
2
```

In [ ]:
```
# If we don't know the Book title so no use of the other information
# so it's better to drop the Null values
df.dropna(subset=['Book Title'], inplace=True)
```

In [ ]:
```
df['Book Title'].tail()
```

Out[ ]:
```
5995                                  The River of Doubt
5996                                                Shug
5997                                              Flawed
5998                  Ø£Ø³Ø¹Ø¯ Ø§Ù
Ø±Ø£Ø© Ù·Ù Ø§Ù_Ø¹Ø§Ù_Ù

5999    Legacy of the Drow Collector's Edition (Legacy...
Name: Book Title, dtype: object
```

In [ ]:
```
# As we can't determine the 2nd last value so better to drop the row
df.drop(df.index[-2], inplace=True)
```

Arranging the Columns

In [ ]:
```
arrange_columns = ['Rating',
           'Publication Cost',
           'ISBN',
```

```
            'Book Title',
            'Year',
            'Genres',
            'Link',
            'Total Revenue',
            'Book URL']

df = df[arrange_columns]
```

In [ ]:
```
# Reseting the index
df.reset_index(drop=True, inplace=True)
```

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

Out[ ]:

	Rating	Publication Cost	ISBN	Book Title	Year	Genres	
0	4.40	136455.0	0439023483	The Hunger Games (The Hunger Games, #1)	2008	young-adult, science-fiction, dystopia, fantas...	dir01/276705
1	4.41	16648.0	0439358078	Harry Potter and the Order of the Phoenix (Har...	2003	fantasy, young-adult, fiction, fantasy, magic,...	dir01/2.Harry_Potter_and_
2	3.56	85746.0	0316015849	Twilight (Twilight, #1)	2005	young-adult, fantasy, romance, paranormal, vam...	
3	4.23	47906.0	0061120081	To Kill a Mockingbird	1960	classics, fiction, historical-fiction, academi...	dir01/2657.To
4	4.23	34772.0	0679783261	Pride and Prejudice	1813	classics, fiction, romance, historical-fiction...	dir01/1885.

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

Out[ ]:

	Rating	Publication Cost	ISBN	Book Title	Year	Genres	
5992	4.37	28.0	0393062260	The Book of Psalms	2007	poetry, religion, christian, religion, theolog...	dir60/125
5993	4.17	2226.0	0767913736	The River of Doubt	2005	history, non-fiction, biography, adventure, bo...	dir60/7
5994	3.99	775.0	1416909427	Shug	2006	young-adult, realistic-fiction, romance, conte...	
5995	3.78	540.0	1620612321	Flawed	2012	contemporary, romance, young-adult, sociology,...	
5996	4.35	61.0	0786929081	Legacy of the Drow Collector's Edition (Legacy...	2001	fiction, fantasy, magic, science-fiction-fanta...	dir60/66677.Legacy_

Export to New Cleaned CSV

In [ ]:
```
# Exporting the CSV to New Cleaned CSV Data File without indexing
df.to_csv('cleaned_book_sample.csv', index=False)
```