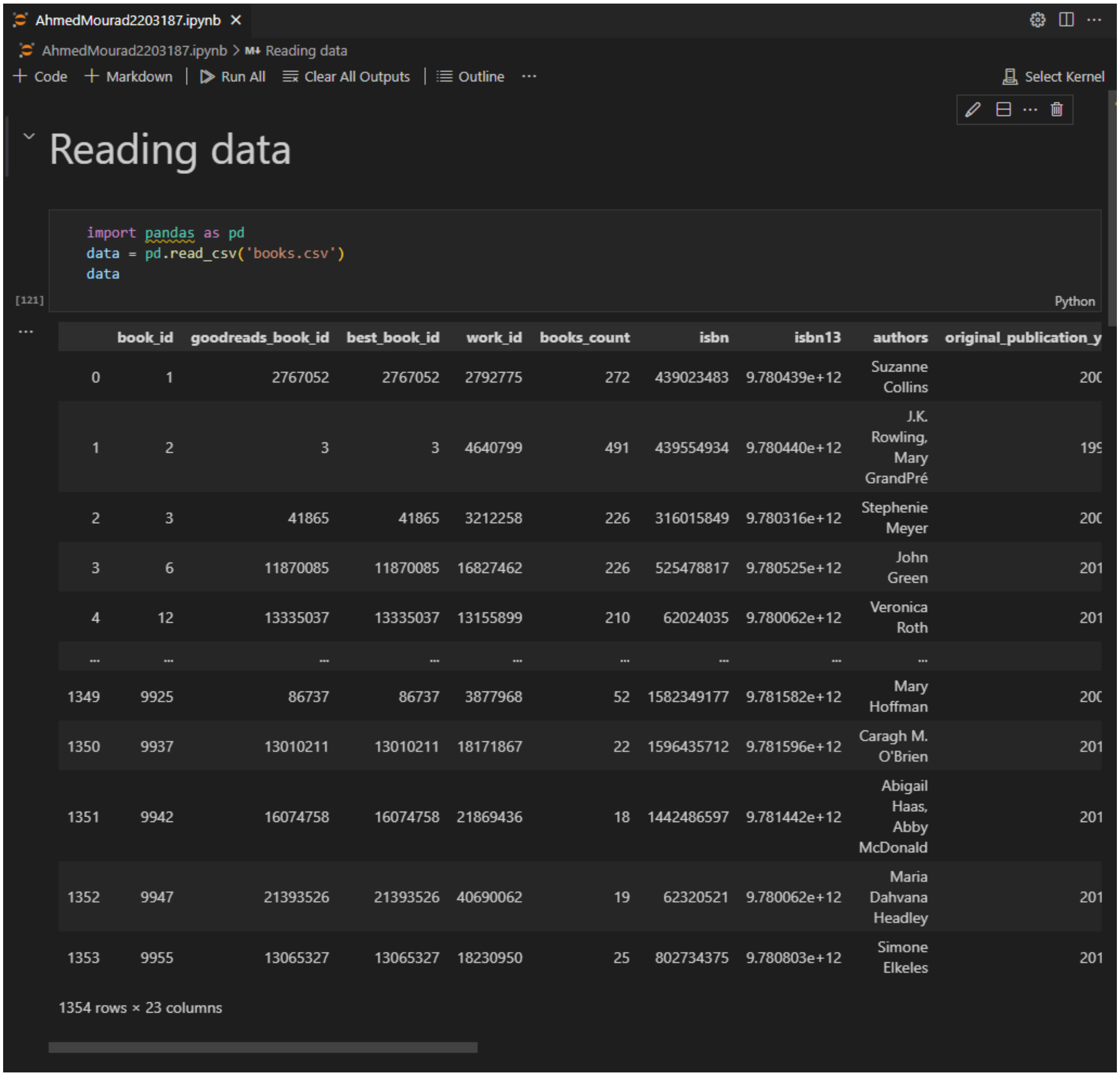
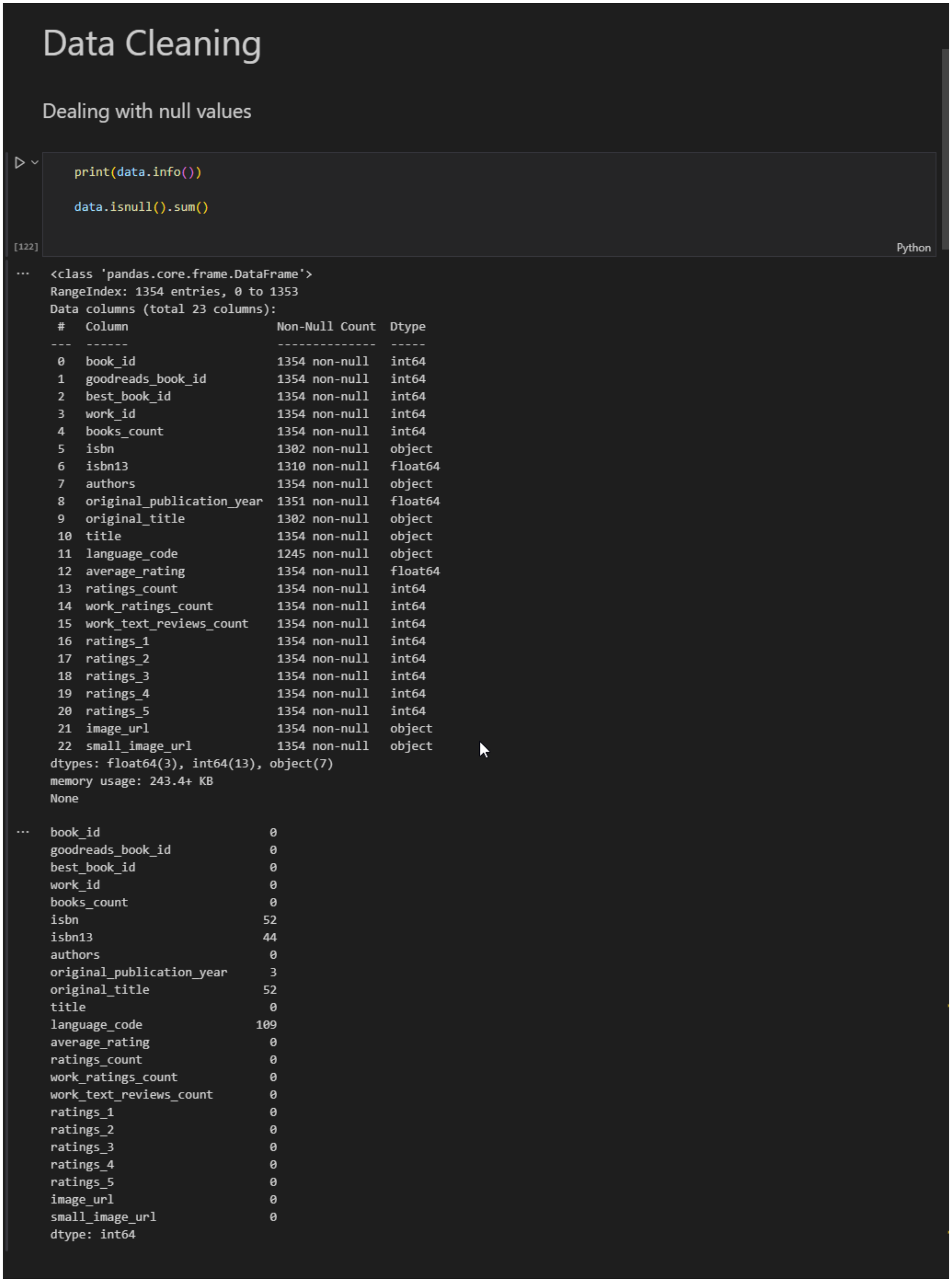


Reading data

```python
import pandas as pd
data = pd.read_csv('books.csv')
data
```

	book_id	goodreads_book_id	best_book_id	work_id	books_count	isbn	isbn13	authors	original_publication_y
0	1	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins	200
1	2	3	3	4640799	491	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré	199
2	3	41865	41865	3212258	226	316015849	9.780316e+12	Stephenie Meyer	200
3	6	11870085	11870085	16827462	226	525478817	9.780525e+12	John Green	201
4	12	13335037	13335037	13155899	210	62024035	9.780062e+12	Veronica Roth	201
...	...	...	...	...	...	...	...	...	
1349	9925	86737	86737	3877968	52	1582349177	9.781582e+12	Mary Hoffman	200
1350	9937	13010211	13010211	18171867	22	1596435712	9.781596e+12	Caragh M. O'Brien	201
1351	9942	16074758	16074758	21869436	18	1442486597	9.781442e+12	Abigail Haas, Abby McDonald	201
1352	9947	21393526	21393526	40690062	19	62320521	9.780062e+12	Maria Dahvana Headley	201
1353	9955	13065327	13065327	18230950	25	802734375	9.780803e+12	Simone Elkeles	201

1354 rows × 23 columns



Data Cleaning

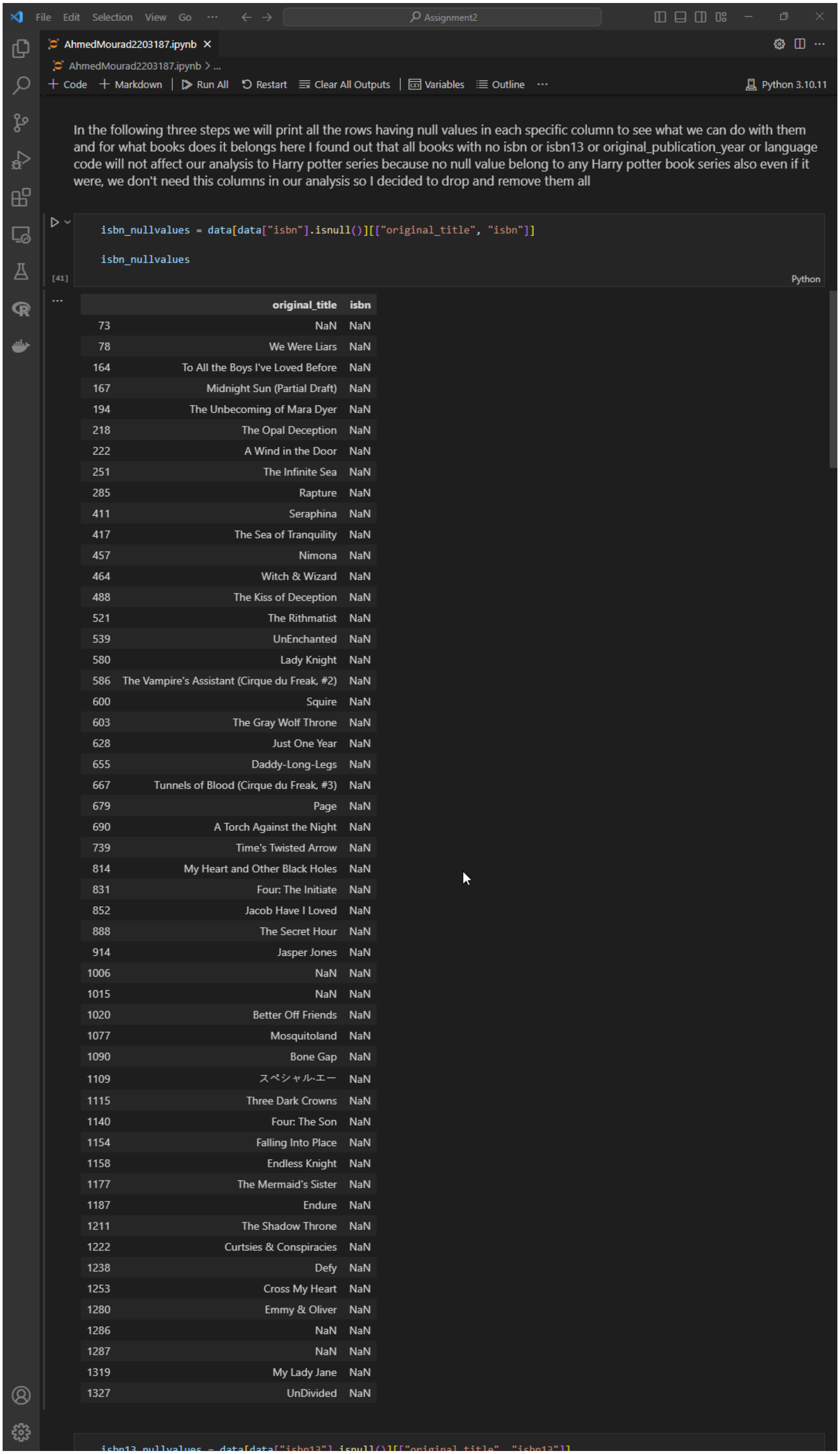
Dealing with null values

```python
print(data.info())

data.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1354 entries, 0 to 1353
Data columns (total 23 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   book_id                    1354 non-null   int64  
 1   goodreads_book_id          1354 non-null   int64  
 2   best_book_id               1354 non-null   int64  
 3   work_id                    1354 non-null   int64  
 4   books_count                1354 non-null   int64  
 5   isbn                       1302 non-null   object 
 6   isbn13                     1310 non-null   float64
 7   authors                    1354 non-null   object 
 8   original_publication_year  1351 non-null   float64
 9   original_title             1302 non-null   object 
 10  title                      1354 non-null   object 
 11  language_code              1245 non-null   object 
 12  average_rating             1354 non-null   float64
 13  ratings_count              1354 non-null   int64  
 14  work_ratings_count         1354 non-null   int64  
 15  work_text_reviews_count    1354 non-null   int64  
 16  ratings_1                  1354 non-null   int64  
 17  ratings_2                  1354 non-null   int64  
 18  ratings_3                  1354 non-null   int64  
 19  ratings_4                  1354 non-null   int64  
 20  ratings_5                  1354 non-null   int64  
 21  image_url                  1354 non-null   object 
 22  small_image_url            1354 non-null   object 
dtypes: float64(3), int64(13), object(7)
memory usage: 243.4+ KB
None
```

```
book_id                        0
goodreads_book_id              0
best_book_id                   0
work_id                        0
books_count                    0
isbn                          52
isbn13                        44
authors                        0
original_publication_year      3
original_title                52
title                          0
language_code                109
average_rating                 0
ratings_count                  0
work_ratings_count             0
work_text_reviews_count        0
ratings_1                      0
ratings_2                      0
ratings_3                      0
ratings_4                      0
ratings_5                      0
image_url                      0
small_image_url                0
dtype: int64
```



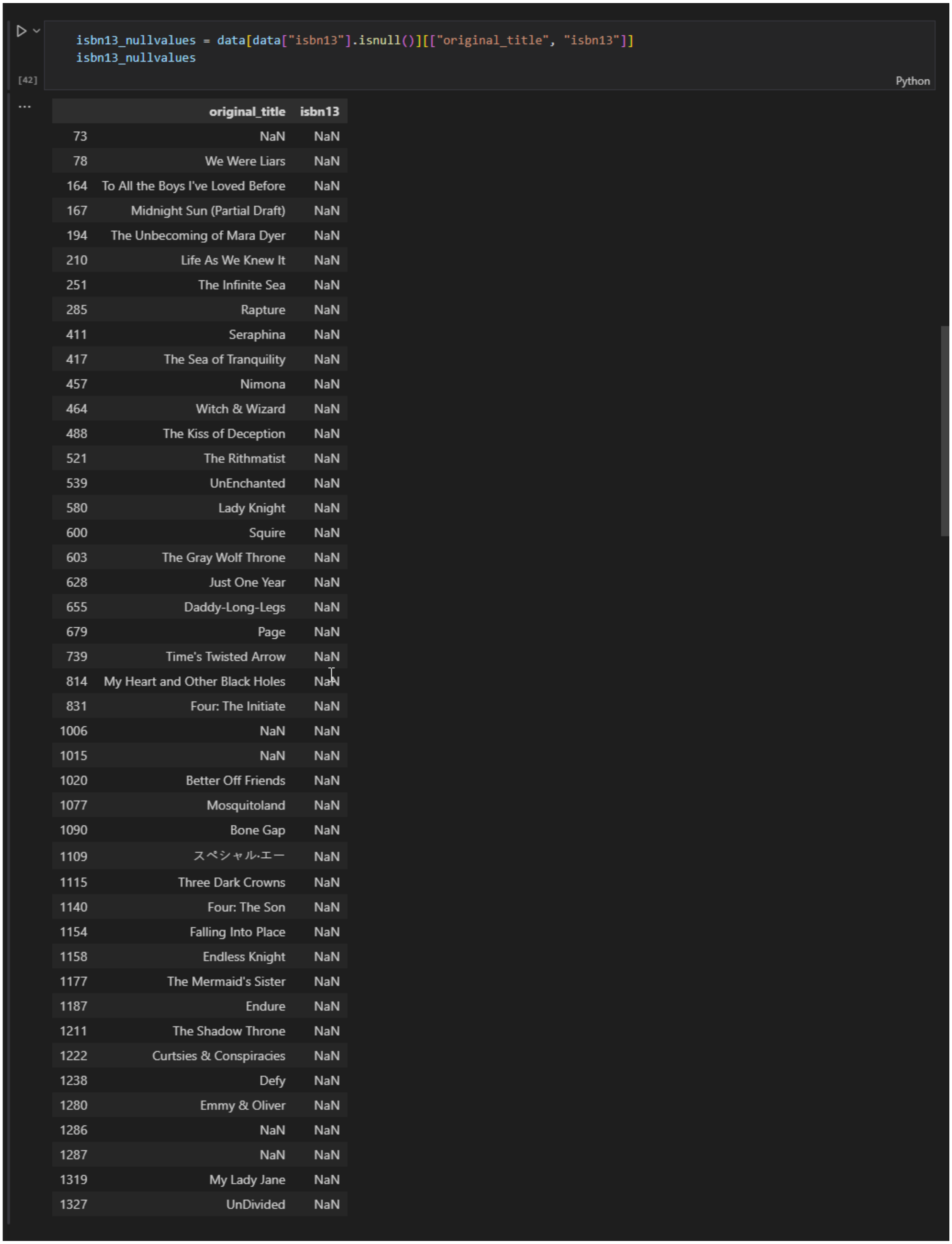
In the following three steps we will print all the rows having null values in each specific column to see what we can do with them and for what books does it belongs here I found out that all books with no isbn or isbn13 or original_publication_year or language code will not affect our analysis to Harry potter series because no null value belong to any Harry potter book series also even if it were, we don't need this columns in our analysis so I decided to drop and remove them all

```python
isbn_nullvalues = data[data["isbn"].isnull()][["original_title", "isbn"]]

isbn_nullvalues
```

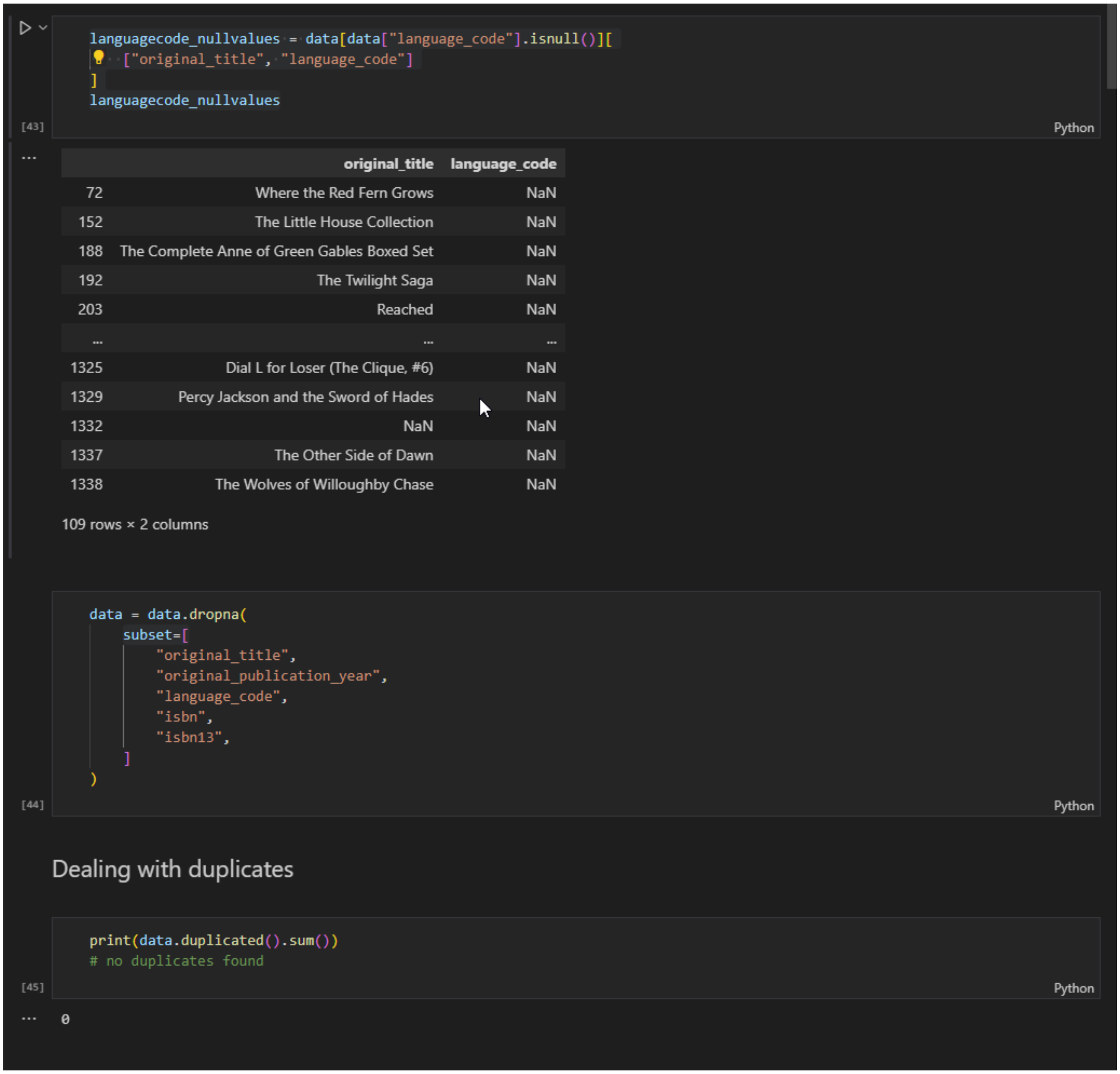
	original_title	isbn
73	NaN	NaN
78	We Were Liars	NaN
164	To All the Boys I've Loved Before	NaN
167	Midnight Sun (Partial Draft)	NaN
194	The Unbecoming of Mara Dyer	NaN
218	The Opal Deception	NaN
222	A Wind in the Door	NaN
251	The Infinite Sea	NaN
285	Rapture	NaN
411	Seraphina	NaN
417	The Sea of Tranquility	NaN
457	Nimona	NaN
464	Witch & Wizard	NaN
488	The Kiss of Deception	NaN
521	The Rithmatist	NaN
539	UnEnchanted	NaN
580	Lady Knight	NaN
586	The Vampire's Assistant (Cirque du Freak, #2)	NaN
600	Squire	NaN
603	The Gray Wolf Throne	NaN
628	Just One Year	NaN
655	Daddy-Long-Legs	NaN
667	Tunnels of Blood (Cirque du Freak, #3)	NaN
679	Page	NaN
690	A Torch Against the Night	NaN
739	Time's Twisted Arrow	NaN
814	My Heart and Other Black Holes	NaN
831	Four: The Initiate	NaN
852	Jacob Have I Loved	NaN
888	The Secret Hour	NaN
914	Jasper Jones	NaN
1006	NaN	NaN
1015	NaN	NaN
1020	Better Off Friends	NaN
1077	Mosquitoland	NaN
1090	Bone Gap	NaN
1109	スペシャル·エー	NaN
1115	Three Dark Crowns	NaN
1140	Four: The Son	NaN
1154	Falling Into Place	NaN
1158	Endless Knight	NaN
1177	The Mermaid's Sister	NaN
1187	Endure	NaN
1211	The Shadow Throne	NaN
1222	Curtsies & Conspiracies	NaN
1238	Defy	NaN
1253	Cross My Heart	NaN
1280	Emmy & Oliver	NaN
1286	NaN	NaN
1287	NaN	NaN
1319	My Lady Jane	NaN
1327	UnDivided	NaN

```python
isbn13_nullvalues = data[data["isbn13"].isnull()][["original_title", "isbn13"]]
```



```python
isbn13_nullvalues = data[data["isbn13"].isnull()][["original_title", "isbn13"]]
isbn13_nullvalues
```

	original_title	isbn13
73	NaN	NaN
78	We Were Liars	NaN
164	To All the Boys I've Loved Before	NaN
167	Midnight Sun (Partial Draft)	NaN
194	The Unbecoming of Mara Dyer	NaN
210	Life As We Knew It	NaN
251	The Infinite Sea	NaN
285	Rapture	NaN
411	Seraphina	NaN
417	The Sea of Tranquility	NaN
457	Nimona	NaN
464	Witch & Wizard	NaN
488	The Kiss of Deception	NaN
521	The Rithmatist	NaN
539	UnEnchanted	NaN
580	Lady Knight	NaN
600	Squire	NaN
603	The Gray Wolf Throne	NaN
628	Just One Year	NaN
655	Daddy-Long-Legs	NaN
679	Page	NaN
739	Time's Twisted Arrow	NaN
814	My Heart and Other Black Holes	NaN
831	Four: The Initiate	NaN
1006	NaN	NaN
1015	NaN	NaN
1020	Better Off Friends	NaN
1077	Mosquitoland	NaN
1090	Bone Gap	NaN
1109	スペシャル·エー	NaN
1115	Three Dark Crowns	NaN
1140	Four: The Son	NaN
1154	Falling Into Place	NaN
1158	Endless Knight	NaN
1177	The Mermaid's Sister	NaN
1187	Endure	NaN
1211	The Shadow Throne	NaN
1222	Curtsies & Conspiracies	NaN
1238	Defy	NaN
1280	Emmy & Oliver	NaN
1286	NaN	NaN
1287	NaN	NaN
1319	My Lady Jane	NaN
1327	UnDivided	NaN



```python
languagecode_nullvalues = data[data["language_code"].isnull()][
    ["original_title", "language_code"]
]
languagecode_nullvalues
```

	original_title	language_code
72	Where the Red Fern Grows	NaN
152	The Little House Collection	NaN
188	The Complete Anne of Green Gables Boxed Set	NaN
192	The Twilight Saga	NaN
203	Reached	NaN
...	...	...
1325	Dial L for Loser (The Clique, #6)	NaN
1329	Percy Jackson and the Sword of Hades	NaN
1332	NaN	NaN
1337	The Other Side of Dawn	NaN
1338	The Wolves of Willoughby Chase	NaN

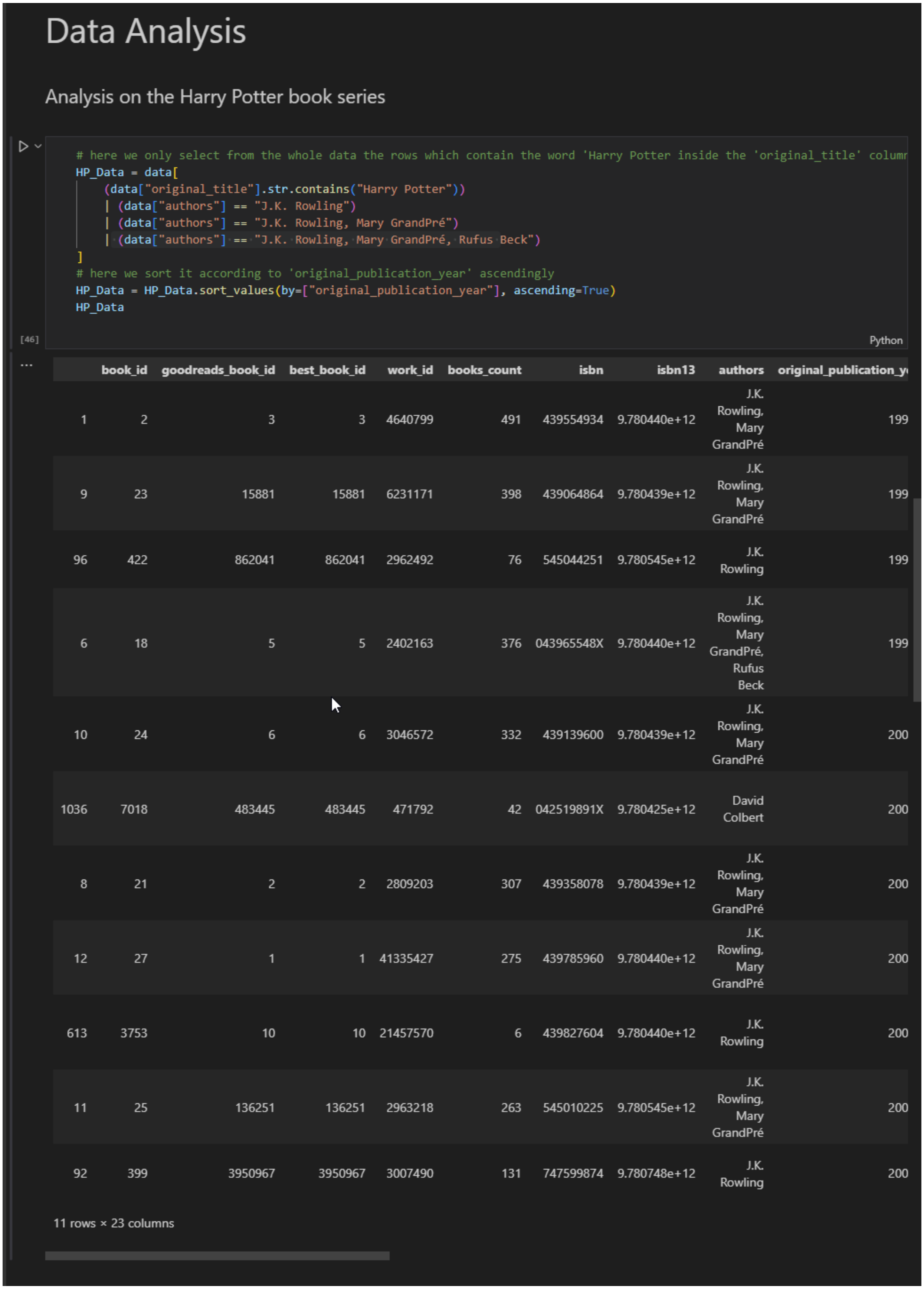
109 rows × 2 columns

```python
data = data.dropna(
    subset=[
        "original_title",
        "original_publication_year",
        "language_code",
        "isbn",
        "isbn13",
    ]
)
```

Dealing with duplicates

```python
print(data.duplicated().sum())
# no duplicates found
```

```
0
```



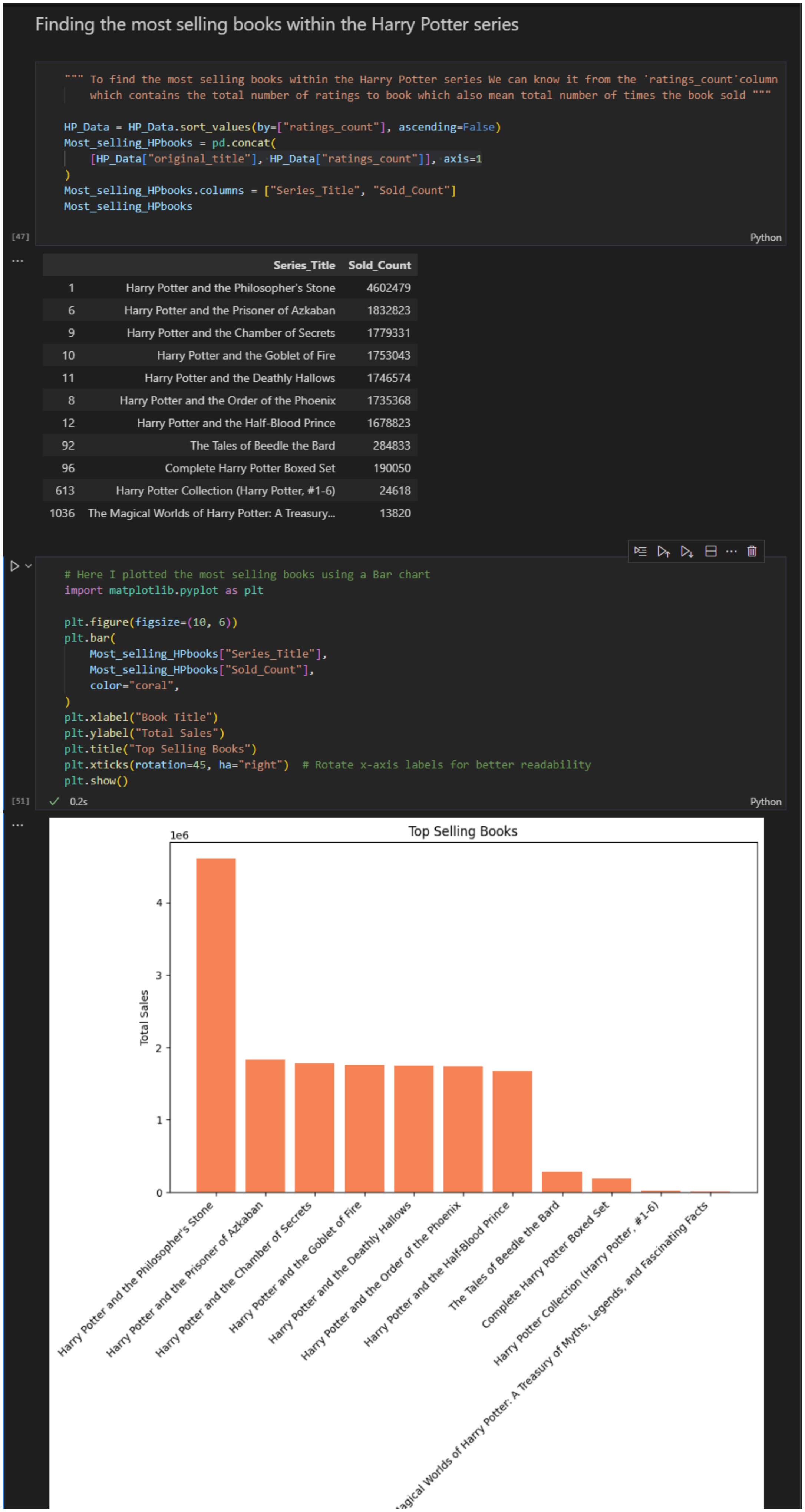
Data Analysis

Analysis on the Harry Potter book series

```
# here we only select from the whole data the rows which contain the word 'Harry Potter inside the 'original_title' colum
HP_Data = data[
    (data["original_title"].str.contains("Harry Potter"))
    | (data["authors"] == "J.K. Rowling")
    | (data["authors"] == "J.K. Rowling, Mary GrandPré")
    | (data["authors"] == "J.K. Rowling, Mary GrandPré, Rufus Beck")
]
# here we sort it according to 'original_publication_year' ascendingly
HP_Data = HP_Data.sort_values(by=["original_publication_year"], ascending=True)
HP_Data
```

	book_id	goodreads_book_id	best_book_id	work_id	books_count	isbn	isbn13	authors	original_publication_y
1	2	3	3	4640799	491	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré	199
9	23	15881	15881	6231171	398	439064864	9.780439e+12	J.K. Rowling, Mary GrandPré	199
96	422	862041	862041	2962492	76	545044251	9.780545e+12	J.K. Rowling	199
6	18	5	5	2402163	376	043965548X	9.780440e+12	J.K. Rowling, Mary GrandPré, Rufus Beck	199
10	24	6	6	3046572	332	439139600	9.780439e+12	J.K. Rowling, Mary GrandPré	200
1036	7018	483445	483445	471792	42	042519891X	9.780425e+12	David Colbert	200
8	21	2	2	2809203	307	439358078	9.780439e+12	J.K. Rowling, Mary GrandPré	200
12	27	1	1	41335427	275	439785960	9.780440e+12	J.K. Rowling, Mary GrandPré	200
613	3753	10	10	21457570	6	439827604	9.780440e+12	J.K. Rowling	200
11	25	136251	136251	2963218	263	545010225	9.780545e+12	J.K. Rowling, Mary GrandPré	200
92	399	3950967	3950967	3007490	131	747599874	9.780748e+12	J.K. Rowling	200

11 rows × 23 columns



Finding the most selling books within the Harry Potter series

```python
""" To find the most selling books within the Harry Potter series We can know it from the 'ratings_count'column
    which contains the total number of ratings to book which also mean total number of times the book sold """

HP_Data = HP_Data.sort_values(by=["ratings_count"], ascending=False)
Most_selling_HPbooks = pd.concat(
    [HP_Data["original_title"], HP_Data["ratings_count"]], axis=1
)
Most_selling_HPbooks.columns = ["Series_Title", "Sold_Count"]
Most_selling_HPbooks
```

[47] Python

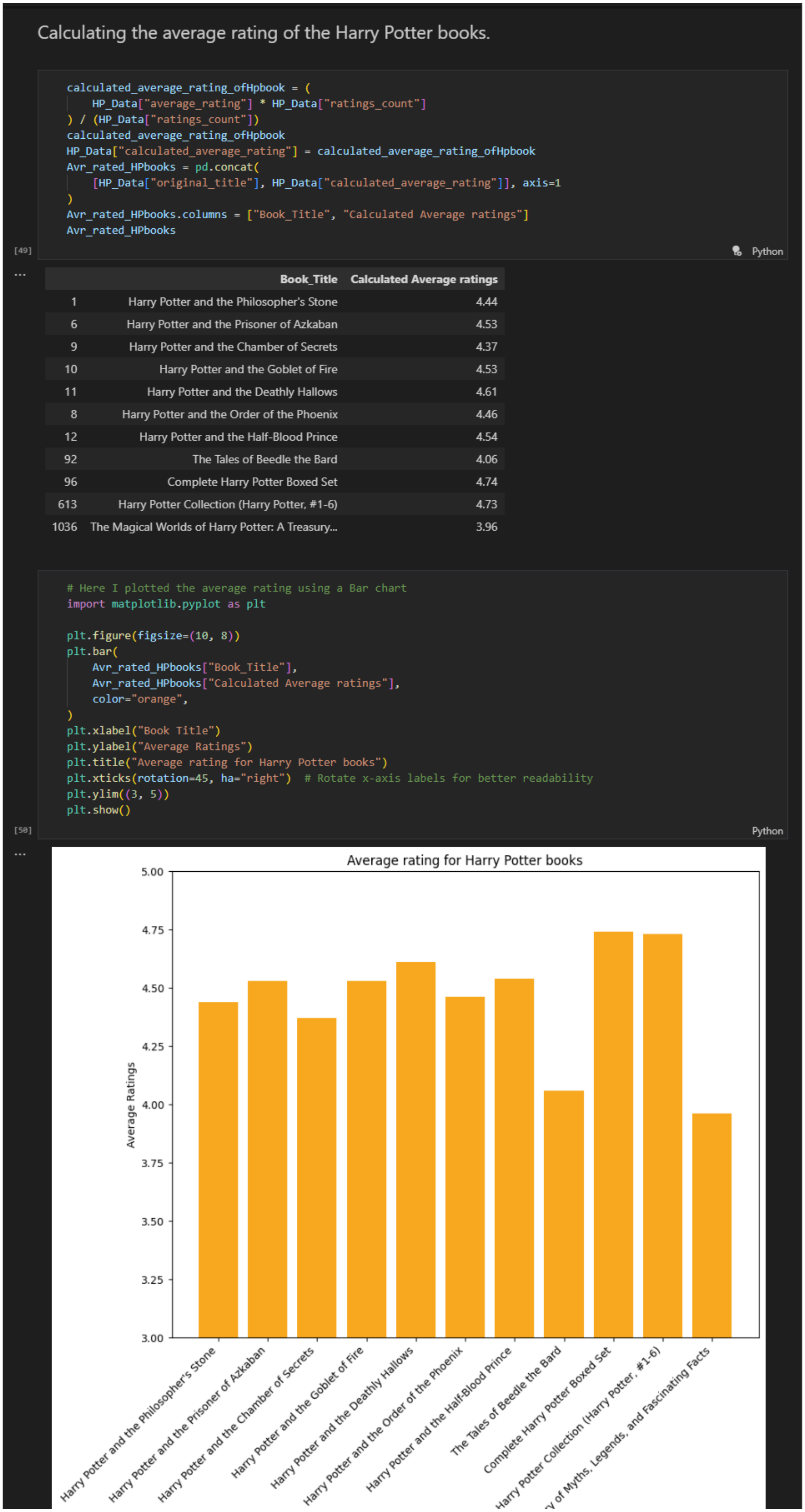
```
                                    Series_Title  Sold_Count
1         Harry Potter and the Philosopher's Stone     4602479
6        Harry Potter and the Prisoner of Azkaban     1832823
9         Harry Potter and the Chamber of Secrets     1779331
10            Harry Potter and the Goblet of Fire     1753043
11           Harry Potter and the Deathly Hallows     1746574
8       Harry Potter and the Order of the Phoenix     1735368
12         Harry Potter and the Half-Blood Prince     1678823
92                   The Tales of Beedle the Bard      284833
96                 Complete Harry Potter Boxed Set      190050
613     Harry Potter Collection (Harry Potter, #1-6)      24618
1036 The Magical Worlds of Harry Potter: A Treasury...    13820
```

```python
# Here I plotted the most selling books using a Bar chart
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.bar(
    Most_selling_HPbooks["Series_Title"],
    Most_selling_HPbooks["Sold_Count"],
    color="coral",
)
plt.xlabel("Book Title")
plt.ylabel("Total Sales")
plt.title("Top Selling Books")
plt.xticks(rotation=45, ha="right")  # Rotate x-axis labels for better readability
plt.show()
```

[51] ✓ 0.2s Python

Top Selling Books
1e6
Total Sales
0
1
2
3
4
Harry Potter and the Philosopher's Stone
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Chamber of Secrets
Harry Potter and the Goblet of Fire
Harry Potter and the Deathly Hallows
Harry Potter and the Order of the Phoenix
Harry Potter and the Half-Blood Prince
The Tales of Beedle the Bard
Complete Harry Potter Boxed Set
Harry Potter Collection (Harry Potter, #1-6)
agical Worlds of Harry Potter: A Treasury of Myths, Legends, and Fascinating Facts



Calculating the average rating of the Harry Potter books.

```
calculated_average_rating_ofHpbook = (
    HP_Data["average_rating"] * HP_Data["ratings_count"]
) / (HP_Data["ratings_count"])
calculated_average_rating_ofHpbook
HP_Data["calculated_average_rating"] = calculated_average_rating_ofHpbook
Avr_rated_HPbooks = pd.concat(
    [HP_Data["original_title"], HP_Data["calculated_average_rating"]], axis=1
)
Avr_rated_HPbooks.columns = ["Book_Title", "Calculated Average ratings"]
Avr_rated_HPbooks
```

	Book_Title	Calculated Average ratings
1	Harry Potter and the Philosopher's Stone	4.44
6	Harry Potter and the Prisoner of Azkaban	4.53
9	Harry Potter and the Chamber of Secrets	4.37
10	Harry Potter and the Goblet of Fire	4.53
11	Harry Potter and the Deathly Hallows	4.61
8	Harry Potter and the Order of the Phoenix	4.46
12	Harry Potter and the Half-Blood Prince	4.54
92	The Tales of Beedle the Bard	4.06
96	Complete Harry Potter Boxed Set	4.74
613	Harry Potter Collection (Harry Potter, #1-6)	4.73
1036	The Magical Worlds of Harry Potter: A Treasury...	3.96

```
# Here I plotted the average rating using a Bar chart
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 8))
plt.bar(
    Avr_rated_HPbooks["Book_Title"],
    Avr_rated_HPbooks["Calculated Average ratings"],
    color="orange",
)
plt.xlabel("Book Title")
plt.ylabel("Average Ratings")
plt.title("Average rating for Harry Potter books")
plt.xticks(rotation=45, ha="right")  # Rotate x-axis labels for better readability
plt.ylim((3, 5))
plt.show()
```

[50]
Python
...
Average rating for Harry Potter books
Average Ratings
5.00
4.75
4.50
4.25
4.00
3.75
3.50
3.25
3.00
Harry Potter and the Philosopher's Stone
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Chamber of Secrets
Harry Potter and the Goblet of Fire
Harry Potter and the Deathly Hallows
Harry Potter and the Order of the Phoenix
Harry Potter and the Half-Blood Prince
The Tales of Beedle the Bard
Complete Harry Potter Boxed Set
Harry Potter Collection (Harry Potter, #1-6)
y of Myths, Legends, and Fascinating Facts