# Task 3: Interactive Analysis with Jupiter

**Author**: Andrei Daniel Bajenaru

## Goal:

Integrate our knowledge in a complete notebook, communicative, that combines the script, explanation and datas of a real context (*books.csv*)

### STEP 1: Load the Data

We will use the data set ***books.csv*** containing the information about books:

*catalog_position, title, author, year_published, genre, section, total_copies, times_borrowed, last_borrowed_date, rating, ratings_count, price, language, page_count, dimensions_width, dimensions_thickness, dimensions_height, catalog_shelf, catalog_row, catalog_row_number*

In [4]:

```
import pandas as pd
df = pd.read_csv("books.csv")
```

### STEP 2: Data Analysis: How Many Times has a book been borrowed?

In this step, we analyze the dataset to understand how each section of the library is performing. First, we group the data by the section each book belongs to. Then, for every section, we calculate:

The number of book titles in that section (**title count**)

The total number of times books from that section have been borrowed (**times_borrowed sum**)

After that, we create a new column called **titles_to_borrow_ratio**, which divides the total borrow count by the number of titles. This ratio helps us see how often books in each section are borrowed on average.

Finally, we sort the sections in descending order based on this ratio to identify which sections are the most actively borrowed and which ones might need more attention or promotion.

In [5]:

```
inventory_gap = df.groupby('section').agg({
    'title': 'count',
    'times_borrowed': 'sum'
})

inventory_gap['titles_to_borrow_ratio'] = inventory_gap['times_borrowed'] / inventory_gap['title']
```

### STEP 3: Readint the Data

We are now able to test and verify the result of our analysis of the previous step.

In [6]:

```
print(inventory_gap.sort_values(by=['titles_to_borrow_ratio'], ascending=False))
```

```
                    title  times_borrowed  titles_to_borrow_ratio
section
Young Adult (YA)       48         10184.0              212.166667
Young Adult            20          3448.0              172.400000
Non-Fiction           215         21595.0              100.441860
Fiction              1148         48570.0               42.308362
Science                28           992.0               35.428571
Children              212          5753.0               27.136792
Children's             23           478.0               20.782609
Reference              52           718.0               13.807692
History                39           522.0               13.384615
Humanities            234           899.0                3.841880
Children's Fiction      1             2.0                2.000000
Rare Books             12             0.0                0.000000
```

## CONCLUSION

To that extend, we have a perfect execution of a **jupyter notebook**, containing a title and author, followed by a step by step execution of the code. The books data regarding the ratio between how many tutkes there are abd how many times they got borrowed has successfully been extracted from the *books.csv* file.