

ECOR 1042
Project Report

CSV Datafile Analysis

Submitted by

T006

MANIT JAWA, 101215842

BALKARAN KARIR, 101229843

NOLAN KISSER, 101222376

ISHTIAQUE KHAN, 101227487

April 12th, 2022

Carleton University,
Faculty of Engineering and Design

1.Content

2	Problem Statement	--
3	Project Goal	--
4	Project Design	--
5	Project Process	--
6	Team Contributions	--
7	References	--

2. Problem Statement

Due to the enormous number of volumes that must be arranged, getting books from libraries might be problematic. Each book needs the storage of a large amount of data in order to identify it, such as the genre or author. Customers may need access to them even if they don't know the specific title, which will be difficult and time-consuming. Employees also should be able to add new books to the shelf and remove old ones that are no longer selling or are damaged.

3. Project Goal

This program loads in a comma separated file on book information and provides options to sort and manipulate the loaded data.

4. Project Design

As mentioned in the previous section the goal of this project has been to sort and manipulate a csv file with data on books. The initial component of this project was to take the raw data from csv file and sort it into a dictionary where the keys were the different book categories and the key values were the list of books with all of their information including title, author, language, rating, publisher and pages. This main sorting function was named the `book_category_dictionary` and took the csv filename as its single argument. This manipulated the data to be presented in a way that was easily readable and it became a basis for the rest of the project (`load_data.py`).

The next part was mainly about extracting and manipulating the data from the `load_data` file. Functions were created to add and remove books from the book category dictionary. All of the book information was entered as an argument for these functions and then the books were either added or removed from the `book_category_dictionary`. Furthermore, there were approximately 6 other functions created to extract information from the dictionary. For

example, the function `get_books_by_publisher` would take a publisher name as an argument and then return the books that were published by that publisher. Much like this there were functions created for author, category and title while there was also a function created to search the data for a specific book which would be the function argument.

The following objective was to sort the `book_category_dictionary` into various ways as the user could possibly need. The bubble sorting algorithm was used to sort the books in the `book_category_dictionary` by alphabetical order, based on the title name, publisher name and author name. The same concept was also used to sort that dictionary by ascending rating of books. Throughout these objectives, there was regular function testing done with a test function module to ensure that the functions were working properly in the variety of situations which could be presented in actual use.

The second last step was to present all of this information to the consumer. All of these functions were set up to manipulate the data, extract information from the data and sort the data but they would be useless without an easy way to use them. For this step we designed a simple text based user interface that the user could use to get any information they wanted from the functions we created. The interface was programmed to have commands for all of the functions we created and there were very quite a few conditional statements to account for any possible input made by the user.

The final step of the design process was to modify and reconfigure any duplications in the code and to write a readme file.

The main technical challenges in this project were to get all of the fine details correct. The main functional coding of each step was not overly complicated but the addition of the smaller details is what became a challenge. For example, for the rate sorting function, books had to be put into ascending order of rating while unrated books had to be put at the front of the list and books with the same rating had to be ordered alphabetically by their titles. Little details such as these posed some minor challenges before we got them functioning perfectly. Another slight problem was compiling all of the code together to make it run without errors and having all automatic tests passing. This was an issue because we all used different variables and had different approaches to our functions. For the following project steps we discussed certain things like variable names prior to coding them so that the compilation of the code would be smoother.

5. Project Process

The program was created in stages, with three modules created by four group members for the most part over a period of several weeks with effort spread evenly.

The project began with the definition of four functions or the completion of four cases, as well as the organisation of a file as a dictionary. Each group member was assigned one

function, however the `book_category_dictionary`, which is case 1 from Milestone Lab 1, is the `load_data` function that was utilised throughout the project.

The second phase, which was completed in Milestone 1 Lab 2, was to create methods to search and edit the dataset using the load data function. Each group member produced three functions, which were then tested using the unit testing approach by the members of the group.

The third phase, also known as Milestone 2 Lab 1, is creating functions that sort a dataset according to specified aspects. In total, the group created six sorting functions. Each of these functions reorders the data supplied by `load_data` according to a set of criteria. The functions were put to the test by the members of the group, who were asked to create tests for each function.

The construction of a user interface was the fourth phase. The user interface would prompt the user to type instructions that would call on the functionalities mentioned earlier. The task was divided evenly and put to the test to see if it worked. As a group, the commands were compiled into a single file.

The original load data method was changed in Milestone 3 to eliminate any duplicate data. The next stage was to create a README file that would explain to users the various functions built in the project.

6. Team Contributions

The report sections 2 and 5 were written by Ishtiaque Khan and the report sections 3, 4 and 6 were written by Balkaran Karir. Nolan Kisser and Mani Jawa read over and edited the report document.

7. References

- [1] C. Ruiz Martin, Milestone Document, Topic “Milestone 1” ECOR1042, Carleton University, Ottawa, ON, Feb., 2022
- [2] C. Ruiz Martin, Milestone Document, Topic “Milestone 2” ECOR1042, Carleton University, Ottawa, ON, Mar., 2022
- [3] C. Ruiz Martin, Milestone Document, Topic “Milestone 3” ECOR1042, Carleton University, Ottawa, ON, Mar., 2022