CSUB

Class Project

Group 7

Hunberto Pascual

Kenneth Wood

Nathan Wardinsky

CMPS: 3500

Professor Morales

* Introduction

ClassProjectGroup7.py is a python implementation of basic data analysis, allowing users to load data from a CSV file into a data frame. Users then can explore, describe, and analyze said data. Exploring data allows for printing columns and their name, dropping any column, counting distinct values, searching for any value, and sorting columns. Describe data does mathematical calculations on a specified column, which includes the mean, median, mode, etc. Analysis answers questions such as how many airlines are in the dataset. Our program functions by allowing users to select a task via interactable menus: the user will be prompted for input and the menu functions will call the necessary function to perform the desired task.

* How the group approached the project

Our first step as a group was to connect on discord and set up a GitHub repository so that we could work cooperatively. The first goal was to write code that would read CSV files and load its data into a data frame. After that, we implemented the first couple of tasks so everyone could see how data frames work and have a base to build off. We pushed our code onto a GitHub repository and scheduled meetings via discord to plan and check our progress. Since we were working remotely off a GitHub repository, all coding was done on our personal computers, and the final version of our code was copied onto Odin.

Once the foundation was set, we split up the tasks; one person focused on creating the general structure for our program along with a menu interface, while the others worked on the describing data section. Once the general structure was built along with a working menu interface, the rest was smooth sailing; everyone coded in functions and added them to the appropriate Class. After the completion of Exploring Data and near completion of Describing Data, we shifted our focus on the analysis questions, implementing error handling and run-time measuring. Lastly, in the final week before the due date, we started writing this Report along with discussions about the presentation and power points.

* Explain how data loading, data cleaning, and search capabilities were implemented

Data loading from CSV files was handled using data frames and lists to isolate specific columns when convenient. The loadFile() function prompts the user for the name of a file to load and, if successfully found, loads it into a global data frame accessible to all functions. An error message is displayed if the file is not found, and the user is asked to try again. For data cleaning, since the CSV files were already clean, we didn’t have much cleaning to do. The index column from the files was removed to clean up the output. Search capabilities were implemented using data frames and lists; for example, the searchColumn() function, which finds a value in a column as specified by the user, first gets the column and value from the user, stores them as capitalized strings, locates the column and stores its data as a list of capitalized strings. Doing so allows our function to search for any number or string without worrying about data type or capitalization. Then it’s a simple for loop iterating through the list to find the specified value.

* Explain the structure of your code, how Error Handling was implemented, and how variable naming and comments were managed in your code to improve its readability

One of our main priorities was getting the general structure of our program done as early as possible so that we wouldn’t have to do any major revisions later, which could get messy and complicated. Our program follows an Object-Oriented approach; we have three main classes, ExploreData, DescribeData, and Analysis which house all corresponding functions. Each Class also has its sub-menu, exporeDataMenu(), describeDataMenu(), and analysisMenu(), each of which displays a menu and handles all looping for its respective Class. Some sections have a while loop that allows the user to perform the same task until the user quits by entering q or immediately jumps back to the sub or main menu after successful completion. Outside of the classes, we also have two other general functions, mainMenu() and loadData(); since these two could be implemented as a singular function, we did not create classes for them. MainMenu() is the main control of the program; it allows the user to select data Loading, ExploreData, DescribeData, and Analysis and then calls the corresponding class submenu. The submenu then prints all available operations, including the option of returning to the mainMenu, and calls the necessary function as requested by the user.

Error handling was implemented using try and except statements and general clever coding. Errors that could be foreseen were avoided by implementing if statements; for example, when asking a user to select a menu operation by entering 1,2,3, etc., we store their input as strings rather than integers. If and else loops handle user input so that regardless of what the user enters, no error would occur. For less apparent errors, we purposely set out to cause them and implemented the corresponding except statements, such as FileNotFoundError and IndexError. Some functions have exception handling, which allows for helpful message printing. Additionally, system-wide Error handling was implemented in the mainMenu function.

As a group, we aimed to make our variables and functions easily understandable at a glance by naming everything as descriptively as possible. For example, col\_to\_describe is the name of the column that will be sent to the DescribeData functions We followed the conventional coding style guidelines we use in Class and the initial base of our program also had good naming and comments, so everyone had a good example to follow Comments were encouraged to be added while coding in more complex functions as they help improve readability.

* Answers to each of the questions
* Conclusion

In conclusion, for this class project, we adopted a primarily object-oriented approach to our project, utilizing Classes and functions to structure our code, worked remotely, and communicated via discord. Everyone contributed to the project, communicated effectively, and attended meetings when asked. Progress was steady from the beginning of our project, so there wasn’t a last-minute rush to finish.