

Assessment Task 3: Data Analysis - Design

Aim

The goal of this assignment is to plan a solution to a data analysis problem from the very beginning, using the tools and skills you have learned in this subject. This assignment also serves as a foundation for Assessment Four, and, together they should be seen as a single assignment covering the entire software development process, from the initial requirements to the implementation.

Task

You are to read through the task description below and plan a complete solution using the tools shown in this course, and the template provided. To begin with you need to identify sections that should logically be functions, and plan IPO charts for each. In addition to the function planning you are also required to provide detailed pseudocode or flowcharts for each function (a sample of a solution has been provided as a guide).

As you complete the planning for each identified function you may identify new potential functions, and you should separate these out as well.

Detailed instructions

We have been asked to create a program that will allow users to load sets of data from CSV files and then view, manipulate and perform simple statistical analysis on the data.

When the program is loaded the user will see the following menu:

Welcome to The Smart Statistician!

Please choose from the following options:

- 1 – Load data from a file*
- 2 – Display the data to the screen*
- 3 – Rename a set*
- 4 – Sort a set*
- 5 – Analyse a set*
- 6 - Quit*

Option 6 will exit the program, every other option will do some task and then display the menu again until the user chooses 6 at this menu.

If the user enters anything other than a value between 1 and 6 an error message is to be displayed.

Option 1 – load data

When the user chooses option 1 they will be asked for a file name, which is expected to be in the same directory as the program (no need for path information etc). The program will open this file which will have contents similar to this:

Rainfall,35,23,12,65,34,111,54,23,68,97

Age,35,23,14,76

Odometer Reading,35065,67443,23545,12323,72335

Each line is a single 'set' of data, the first element is the name of the set and the rest is the actual data. Each 'set' should be its own list, and you should plan on using a list of lists to store the data. Each set may be any length, so you should plan to allow for any number of entries.

If the file exists then you can assume it is correctly formatted, but if it does not exist you will need to display an error message before returning to the menu.

Option 2 – display data

Choosing this menu option will display each set of data, one after the other, with each set being displayed on a single line as shown here:

Rainfall

35, 23, 12, 65, 34, 111, 54, 23, 68, 97

Age

35, 23, 14, 76

Odometer Reading

35065, 67443, 23545, 12323, 72335

Note that each 'set' has a row of 10 dashes after it.

Option 3 – rename a set

When the user wants to rename a set we first have to find out which set. To do this we will display the sets in order with a number, and we ask the user to enter a number. This might look like this:

Which set do you want to rename?

1 – Rainfall

2 – Age

3 – Odometer Reading

The user will then need to enter the appropriate number (in this case between 1 and 3). The range will change depending on the number of sets in the system.

If the user enters an invalid number then display an error and ask again until they enter a valid number.

Once they have entered the number we then ask for the new name for that set. There are two requirements for the new name:

1. It cannot be blank
2. It cannot already be in the system

You need to check both of these (in programming the empty string is "", and you can always compare the new name to the names already in the system, if names have different capitalisations they are considered different names). If the name is invalid then display an error, and ask for a new name until a valid name is entered.

Once we have a valid name we will replace the first element of the appropriate list with the new name, and then display a message to the user:

Rainfall renamed to Monthly Rainfall

Option 4 – sort a set

The first thing we need to do is find out is which set. We will do this the same way we did in option 3 - display the sets in order with a number, and we ask the user to enter a number. This might look like this:

Which set do you want to sort?

1 – Rainfall

2 – Age

3 – Odometer Reading

The user will then need to enter the appropriate number (in this case between 1 and 3). The range will change depending on the number of sets in the system.

If the user enters an invalid number then display an error and ask again until they enter a valid number.

Once we have the number we need to sort the data. Don't worry about how this will be done here, Python allows us to easily sort lists. In planning we can just say 'sort the list' and that will be fine for now.

Option 5 – stats

The first thing we need to do is find out is which set. We will do this the same way we did in option 3 - display the sets in order with a number, and we ask the user to enter a number. This might look like this:

Which set do you want to analyse?

1 – Rainfall

2 – Age

3 – Odometer Reading

This menu option will produce a statistical report for a set of data, using a format similar to that shown here:

Rainfall

Number of values (n): 10

Min: 12

Max: 111

Median: 44.5

Mode: 23

Since this is only a plan you do not need to describe the formatting details, but you do need to describe how you will calculate each of these values using the Python data structures you have access to (e.g. you should have pseudocode to that shows how to calculate the median of a dataset). You should not reference numpy or pandas as they haven't been covered yet, nor any other library for calculating statistics. You may choose how to deal with a non-unique mode.

Getting started

As a starting point it is suggested that you have your main function handle the welcome and menu, and have every menu option (other than quitting) be a separate function. Remember that your data must be available to each function, so main acts like a hub, and must store the data. Each function will then be passed the lists as input to do their jobs.

Start planning each part independently, but if you find similar parts to each job then consider splitting that into its own function. Consider how much additional code would be needed to make this work, compared to how many lines are in the function. If you save lines of code then do it.

Submission

Your completed solution should be submitted to JCU Online in either Word (.docx) or PDF (.pdf) format.

You are free to use any editor you like, but please ensure you export/save your file in one of those two formats. Please name your file LastnameFirstname.ext (for example John Smith's word document would be named SmithJohn.docx).

If you have chosen to use flowcharts then they must be submitted as image files embedded in your document (see the sample solution). Flowcharts drawn by hand are not acceptable, they must be produced using any flowcharting software you like (see www.lucidchart.com or www.draw.io for free online tools).