

## **Assessment Task 4: Data Analysis - Implement**

### **Aim**

The goal of this assignment is to review and implement your plan from Assessment Three using the programming skills you have learned in this subject. This assignment builds on Assessment Three, 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 review your plan from Assessment Three and read through the feedback provided by your tutor. You need to adjust your plan accordingly and then implement the solution in Python 3 (any submission in Python 2 or any other language is unacceptable).

### **Detailed instructions**

**These are the same instructions as for Assessment Three, with minor clarifications on implementation.**

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 from a file

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 the data to the screen

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 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 a valid number we need to sort the data. This should be easy with Python, if not then revisit the material on Lists.

#### **Option 5 – analyse 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 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*

You should use string formatting to display the report as shown here.

## **Getting started**

As a starting point it is suggested that you create your main function and display the welcome and menu, and have every menu option (other than quitting) just display a message.

Then have that message display from inside a separate function.

Next create your lists (with dummy data, no files yet) and pass them around to each function, just to test the data flows.

Finally, start implementing each part following your amended plan.

## **Submission**

Your completed solution should be submitted to JCU Online as a single Python 3 (\*.py) file. You are free to use any IDE you like, including Jupyter Notebook, IDLE or any other tool you like, but ensure that you only submit your source code as a \*.py file. Please name your file LastnameFirstname.py (for example John Smith's code would be named SmithJohn.py).