Reflective Report on Task 1:

The aim of the task is to test my knowledge of programming concepts as well as my understanding of the use of built-in functions in python. My choice of approach in all the task was based on simplicity and ease of understanding.

The first function involved finding the arithmetic mean of a list of values. I used the built-in sum and length function in python to get the average of the list of data.

For the second function, I simply used the formula for standard deviation to write the function. I found this straight forward as this just involved manipulating a formula to create a function.

The third task involved creating a function that gets the minimum, maximum as well as the total number of items of a list. For this task, I decided to sort the values in the list using the sort function in python. I could have used some of the built-in functions already in python, however, I wanted to show the inner workings of those functions.

The percentile function was a bit tricky as there are different ways this could be done. I tried to implement something close to the numpy percentile function, but I later found that a bit difficult to achieve, as I kept getting some margin of error. I decided to stick with the simpler approach which involves picking the rank greater than or equal to the percentile position.

The fifth and sixth task involved reading csv files. The fifth task was easy as I could use a for loop to read each line in the csv file. Another function I could have used is the readlines function in python, however, I realised that was not necessary as I could loop through the lines in the file without needing to use the function. I used the same approach for the sixth task, but here, I used the next function to get the column names, as this only captures the first line in the csv file.

I attempted the seventh task by reading in the values of dictionary output from the sixth function. While making the necessary calculations as well as adding the calculations to a dictionary. I realised that there was a None value in one of the columns, for this, I used a try and except statement to convert the strings to floats while skipping any value that cannot be converted to floats. One thing I could have done to make my code in this exercise shorter is to make the calculations on each column and add the values to a dictionary as I go. Instead, I added the values to empty lists first.

I had some challenges with the eighth task, but in the end, I opted to set up a table with a constant shape. Where the outputs can still be defined.

I found the exercises engaging, and I experimented with a lot of things while working on it.