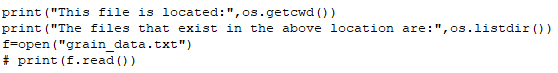
**Assignment 2**

**Question 1**

For this exercise we will be using the data contained in ‘grain\_data.txt’. The text file was saved into the same folder as the python script in order to avoid having to divert to another directory. The file location can be checked via **Fig.1.1** below.

**Fig.1.1:** Assuring that the file ‘grain\_data.txt’ exists in the same location as code.

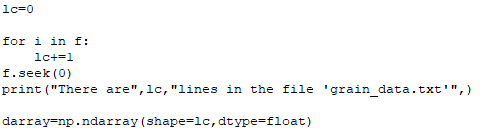


Printing ‘(f.read())’ caused an error when running the rest of the code. It was left out as it wasn’t necessarily needed later. However, it was used to view the file contents earlier on in the build.

After opening the file (**Fig1.1**)an array was then created which was 1000 line long (the same number of values contained in the text file). This was done by **Fig.1.2**.

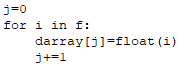
The code would read each line (lc) until there were none left to read, counting the line each time, given by ‘lc=lc+1’. The problem arose when it would try to keep counting past the last line if you ran the programme again. This was solved by the command ‘f.seek(0)’ which told the code to start again at line 0.

**Fig.1.2:** Declare the text file as an array called ‘darray’.



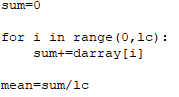
The array is too large to show here so It has not been included.

To create the array the code had to turn each line from a string into an integer (Float) by storing each element of ‘i’ and turning them into a float (See **Fig.1.3**)



**Fig.1.3:** Turing each line in the array into float.

To find the mean average, the sum of all the integers, from the array was taken and divided by ‘lc’ which can be found in **Fig.1.4.**



**Fig.1.4:** Finding the mean average from the text file.

The mean average was thus calculated, printed into the shell and compared using the inbuilt ’numpy’ command, ‘np.mean(darray)’

**Fig.1.41:** What was printed to the shell.



In **Fig 1.5** the code was written to print the mean average into the shell and then create a new text file (If one did not exist already) named ‘question\_a1’ with the value of the ‘mean’ printed into it. It is worth noting that if the file already exists, but you change the quoted text, then the file will be overwritten. This method was repeated for question 1: a, b and c.

**Fig.1.5:** Writing a file.





The mean average of the grain data was calculated to be 0.6µm. Hence it can be corresponded to grain 1. (See **Fig.1.6**)

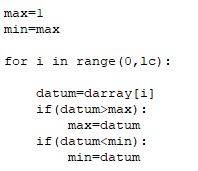
**Fig.1.6:** What was printed to the shell.





In order to calculate the maximum and minimum values, we can use the same array that we created in **Fig.1.2** and **Fig.1.3**

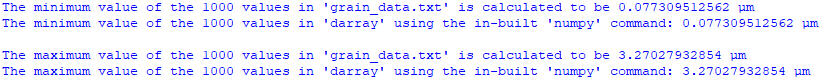
**Fig.1.7:** Calculating the maximum and minimum values.



**Fig.1.7** Tells the programme to read all of the lines in ‘grain\_data.txt’ line by line and that if ‘datum’ is greater than the maximum value then the ‘max’ = ‘datum’. But if ‘datum’ is less than the minimum value then makes ‘min’ = ‘datum’.

This allows us now to sperate the min and max from the 1000 lines of data and print them to the shell. The values were compared using the in-build ‘numpy’ commands’ np.amin(darray)’ and ‘np.amax(darray)’ and printed to the shell (See **Fig.1.71**)

**Fig.1.71:** What was printed to the shell.





(1)

**Fig.1.8** Shows how the loop used to calculate the array of values, ‘darray’ and the loop used too calculate the mean average, ‘mean’ was used to calculate the standard deviation (**Eq.1**)**.**

**Fig.1.8:** Calculating the standard deviation andwhat was printed to the shell.



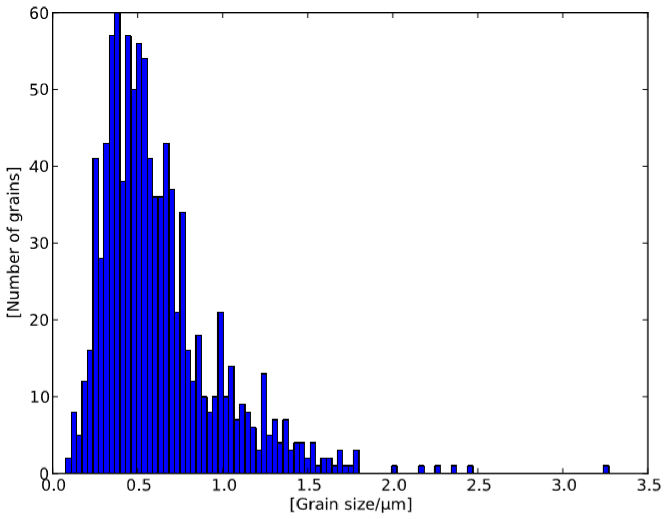
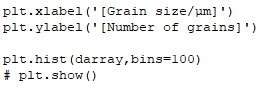


The location of any text files that had be created were also printed into the shell for convenience.

**Question 2**

Firstly, an array of the values form ‘grain\_data.txt’ was formed using the same method from **Fig1.2.** and **Fig.1.3.** The values were then compiled into a histogram showing in **Fig.2.1**

**Fig.2.1:** Histogram of ‘darray’



The mean average was calculated to be 0.626033474479 μm. **Fig.2.1** looks to agree with this where the majority of values being just below this value, with a few significantly larger values, such as the max we calculated (Which can be seen on the far right of the plot) that contribute the to the overall mean average.

The code was written to save the histogram to a pdf file called ‘question\_2\_plot.pdf’ (**Fig.2.2**)

**Fig.2.2:** Writing the histogram to a pdf



The ‘bbox\_inches=’tight’’ command is used to ensure that it trims any unnecessary space from around the image.

The location of the pdf was also also printed into the shell for convenience.