Report on fairness of sailing scoring

Problem

In this coursework I am investigating how sailors score in a series of races, how performances of sailors fluctuate and how the consistency of their performance affects their score.

Method

To solve this problem, I have used Python 3.7.1 to create a program with a series of functions, one of which reads an external csv file. Once the csv file is read another function (generate_performance) adds the data from the file to a dictionary (performance_list). Another function (calculate_finishing_order) then generates the scores for each of the sailors in the series and puts this data into a list (finishing_order) and then sort's it into the order of how they finished. However, in a series of races the sailor's worst performance is discarded and the sum of the remaining performances calculated, to do this I added two other functions (series_score and sort_series) the first of which generates the score for the sailor and the second of which sorts the series of sailors into order of their performance.

Assumptions

- ☑ That the sailors attend all races.
- ☑ That all series have 6 races.
- ☑ The skill value of the sailors was based on a mean.
- ☑ The code is unbiased.

Results

Although I couldn't get the graphs generated using the matplotlib Python extension I did manage to generate the average scores for each of the sailors in the series.
Below is the average position for each sailor taken over 10 iterations.

```
[['Alice', 1.9], ['Bob', 2.3], ['Clare', 2.5], ['Dennis', 4.3], ['Eva', 4.0]]
```

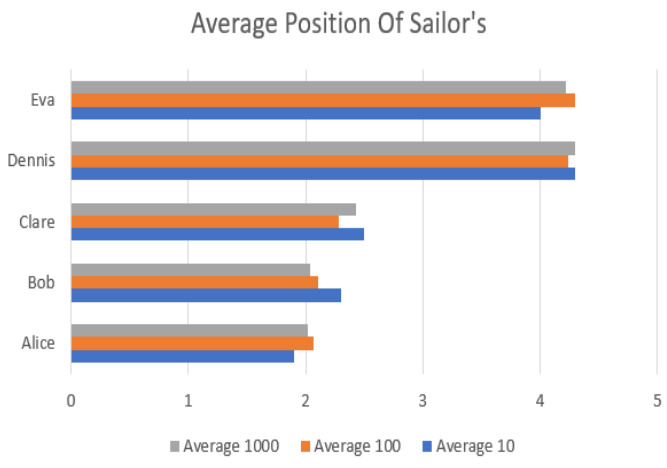
Below is the average position for each sailor taken over 100 iterations.

```
[['Alice', 2.07], ['Bob', 2.11], ['Clare', 2.28], ['Dennis', 4.24], ['Eva', 4.3]]
```

Below is the average position for each sailor taken over 1000 iterations.

```
[['Alice', 2.015], ['Bob', 2.04], ['Clare', 2.43], ['Dennis', 4.294], ['Eva', 4.221]]
```

Then using this data, I produced a Histogram in Excel by inputting the values into a table. The results to the right however show that over 10 races it is better to be a consistently good sailor like Alice however over 100 and 1000 iterations the difference between them goes from a 0.4 lead to Alice to only 0.025 lead to Alice. It is a similar situation on the other end of the graph between Eva and Dennis with Eva ahead by 0.3 and over 1000 iterations Eva's lead narrows to 0.073. The case of Alice and Bob differs from Eva and Dennis as Alice has a lead on Bob by being a consistently good sailor



however Eva is ahead even though she is inconsistent with her position.

Conclusions

The code for this project went through a few iterations going from multiple line and multiple variable functions which I gradually broke down into series of minimal lines of code. The code does however experience limitations with value exceptions, however this may be addressed in further iterations of the code and this can then allow the code to become more effective. The code also at current does not accept any user input this is something that may also be addressed in further iterations of the program. The code is also limited as it can only open one file 'sailor_performance.csv' this could be addressed in further versions of the code with a GUI to select the file which you want the code to read, otherwise at the moment the user would have to either edit the 'sailor_performance.csv' or directly edit the code whenever they wanted it to view a new list of sailors.