Course: Introduction to Data Science (DS2006) - Laboratory 12

Task 1: LogAnalyser class implemented

Task 2: The two lines of code new to me are:

- `import pandas as pd` – the external library `pandas`
- `pd.DataFrame()` – the `DataFrame()` object from the `pandas` library

Task 3:
When importing a library there is an option to give the library an alias. In this line the `pandas` library's alias was chosen to be `pd`. The usage of the library remains, but now instead of calling the DataFrame object as `pandas.DataFrame()`, it can be called like `pd.DataFrame()`.

Task 4:
This line creates an empty instance of the `pandas` library's `DataFrame` object, since no parameters are passed, and stores it as an instance variable named `df`.

Task 5: results.txt implemented

Task 6 & 7: `LogAnalyser.load_file()` implemented

Task 8:
When `print(log_analyser.df.head(2))` is called, the first 2 rows of the `DataFrame` object are displayed, allowing quick preview of the data without printing the entire `DataFrame` object.

Task 9:
`print(log_analyser.df.info())` displays both a summary of the DataFrame structure, including data types, memory usage, how many null values exist, as well as outputting `None`. This is because the `info()` method displays the object summary, while the `print` function outputs the return value of the `info()` method which is `None`.

Task 10:
`print(log_analyser.df.describe())` outputs descriptive statistics of the data such as the value count, mean, as well as the minimum and maximum values.

Task 11:
`print(log_analyser.df.shape)` prints the `DataFrame` object's dimensions as a tuple containing the rows and columns, which outputs `(6,3)` representing the rows and columns from the `results.txt` file.