Assignment 2

Due date: Friday, June 24, 2022
Submission via Git only

Programming environment

For this assignment you must ensure your work executes correctly on the virtual machines (i.e., Senjhalla or its analogous environment for students with M1 Mackbooks) you installed/configured as part of Assignment 0 as this is our "reference platform". This same environment will be used by the teaching team when grading the work submitted by the SENG 265 students.

Since we are now more familiar with the Git framework, **all test files and sample code for this assignment are available in the 'a2' folder of your git repository** and you must use the git pull command to download a copy of the files.

Any programming done outside of Senjhalla might result in lost marks or even 0 marks for the assignment. Make sure that your submitted file is named **"f1_statistics.py"** and is inside your **a2 folder**.

Individual work

This assignment is to be completed by each individual student (i.e., no group work). Naturally you will want to discuss aspects of the problem with fellow students, and such discussion is encouraged. However, sharing of code fragments is strictly forbidden. Code-similarity analysis tools are used to check submitted work for plagiarism.

Learning objectives

I. Learn or review basic features of the Python 3 programming language.
II. Use the Python 3 programming language to write an introductory data science project.
III. Use Git to manage changes in your source code and annotate the evolution of your solution with messages provided during commits. Remember, the only acceptable way of submitting your work is using Git.
IV. Test your code against the provided test cases.

f1_statistics.py: Descriptive analytics about F1 data

a) f1_statistics is a small program that uses relevant Python structures and libraries to process historical Formula One (F1) data to produce descriptive analytics.
b) Based on provided arguments and data files (i.e., datasets), f1_statistics.py will help us answer the following questions:

Q1: Who are the top 20 drivers with most wins in F1 history?

Q2: What are the top 10 countries with most race-winners in F1 history?

Q3: What are the top 10 constructors with most wins in F1?

Q4: What are the top 20 countries with most hosted F1 races?

Q5: Who are the top 5 drivers with most wins in F1 history who started a race not being on pole position?

c) The format of the arguments used for the program are like the ones used for Assignment 1.

d) As opposed to the first assignment, the output of your program won't be directed to stdout. After each execution, your program must generate two files:

  a. **output.csv:** A file compliant with the CSV File format that contains data in a table-based format to answer the question passed as an argument to the program (e.g., **Q1**, **Q2**)

  b. **output_graph_qX.pdf \ output_graph_qX.jpeg:** This is a file that contains an image with a graph\figure (e.g., bar plot or pie plot) that allows to visualize the data in **output.csv** and answer the question passed as an argument to the program (e.g., **Q1**, **Q2**). Please note that the 'X' character in the name of the file corresponds to the question passed as an argument. For example, **output_graph_q1.pdf, output_graph_q2.pdf, output_graph_q3.pdf, output_graph_q4.pdf**, and **output_graph_q5.pdf.**

e) Although the **Unix diff** command might still be used, the most reliable way to test your program is to use the provided **tester.py** file which will validate the output produced by your program in **output.csv**, given a particular test (i.e., a set of arguments).

Restrictions

- Your program must be decomposed into easy-to-understand components. Good program decomposition is required. Methods or functions must be short and effectively parameterized.
- Unwieldy functions are not accepted. The main function should especially be easy to understand and represent a decomposition of the problem at hand.
- Typing (i.e., type hints) must be used in variables and functions defined in your program.
- Do not use global variables.
- Although any Python module and Python libraries may be used (e.g., the numpy, pandas, and matplotlib modules could be useful), **ONLY IF** your implementation requires the installation of a specific library that was not automatically installed in Senjhalla as part of A0 (e.g., numpy and matplotlib were automatically installed), you **MUST** include the installation command to be used in Senjhalla in a file called **README.txt.** Failing to provide this information will result in significant lost marks or even 0 marks for the assignment.

Testing your solution

- Refer to the example commands in the file "TESTS.md" for appropriate command line input. Your solution must accommodate all specified command line inputs.
- Make sure to use the test files provided as part of this assignment.
- Keep all your code in one file (**f1_statistics.py**) for this assignment. In Assignment 4 we will use the multi-module features of Python.
- Use the test files and listed test cases to guide your implementation effort. Develop your program incrementally. Save stages of your incremental development in your Git repository.
- For this assignment you can assume all test inputs are well-formed (i.e., exception handling is not required).
- Do not rely on visual inspection. You can use the provided tester file (i.e., "tester.py") so you can verify the validity of your outputs in a simpler manner.

What to submit

A single Python source file named "f1_statistics.py" submitted (i.e., Git push to your remote repository) to the a2 folder in your repository.

Grading

Assignment 2 grading scheme is as follows. In general, straying from the assignment requirements might result in zero marks due to automated grading.

Scheme

- **(30%) # Tests Passed**
- **(30%) # Appropriate Graphs\Figures Generated**
- **(40%) Code Qualitative Assessment**
    - **(20%) Functional decomposition, program-scope or file-scope variables:** quality coding requires the good use of functions. Code that relies on few large functions to accomplish its goals is considered poor-quality code. Typically, a good program has a main function that does some basic tasks and calls other functions, that do most of the work.
    - **(20%) Code structure:** The code style must be consistent, uniform, and facilitate readability throughout the file.
    - **(15%) Proper naming conventions:** You must use proper names for functions and variables. Using random or single character variables is considered improper coding and significantly reduces code readability. Single character variables as loop variables is fine.
    - **(15%) Typing:** to facilitate readability and maintainability in your solution, typing (i.e., type hints) is required for this assignment when declaring variables and functions.
    - **(10%) Debugging/Comment artifacts:** You must submit a clean file with no residual commented lines of code or unintended text.
    - **(10%) Documentation and commenting:** the purpose of documentation and commenting is to write information so that anyone other than yourself (with knowledge of coding) can review your program and quickly understand how it works. In terms of marking, documentation is not a large mark, but it will be part of the quality assessment.
    - **(10%) Quality of solution:** marker will access the submission for logical and functional quality of the solution. Some examples that would result in a reduction of marks: solutions that read the input files several times, solutions that represent the data in inappropriate data structures, solutions which scale unreasonably with the size of the input.

Scale

A grade		B grade		C grade		D grade		F grade		
grade	marks	grade	marks	grade	marks	grade	marks	grade	marks	description
A+	90-100	B+	77-79							Either no submission given, or submission represents little work or none of the tests pass. No submission, 0 marks. Submissions that do not run, 0 marks. Submissions that fail all tests and show a poor to no effort (as assessed by the marker) are given 0 marks. Submissions that fail all tests, but represent a sincere effort (as assessed by the marker) may be given a few marks
A	85-89	B	73-76	C+	65-69					
A-	80-84	B-	70-72	C	60-64	D	50-59	F	0-49	

Input specification

- All input test files are in CSV format and come from the “Formula 1 World Championship (1950 - 2022)” dataset in Kaggle.com.
- The metadata (i.e., description of each column) for the input datasets can be found in the file **METADATA.md**.

Output specification

- After execution, your program must produce\create the following files:
  - **A. output.csv**: This file will contain two-dimensional data (i.e., a table with two columns) that represents the answer to the question passed as an argument to the program (e.g., **Q1**, **Q2**, **Q3**, **Q4**, and **Q5**).
    - The first column in the table (i.e., X axis) must be named 'subject'.
    - The second column in the table (i.e., Y axis) must be named 'statistic'.
    - Examples of the expected output.csv files for each test are provided (i..e, test01.csv, test02.csv, test03.csv, test04.csv, test05.csv).
  - B. **output_graph_qX.pdf \ output_graph_qX.jpeg:** This is a file that contains an image with a graph\figure (e.g., bar plot or pie plot) that allows to visualize the data in output.csv and answer the question passed as an argument to the program (e.g., **Q1**, **Q2**, etc.). The 'X' character in the name of the file corresponds to the question passed as an argument (e.g., output_graph_q1.pdf, output_graph_q2.pdf, etc.).
    - Note that the results obtained in this assignment might differ from official F1 publications as the current championship is under development and future race results won't be reflected in the datasets used (the Kaggle repository is continuously updated though).
    - Any image format is accepted.
    - Only one image is necessary for each test.
    - The effectiveness of the image generated to answer the question asked to the program will be evaluated. Thus, choose and design your plots carefully. Include plot title, labels, axis names, and other relevant information in the plots to make sure that the question can be answered using your visualization.
    - Examples of plots generated using the matplotlib module are presented below.

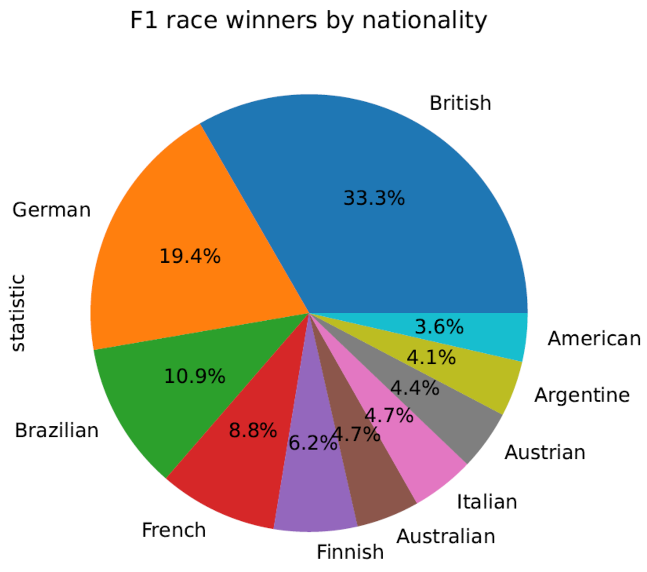

Figure: Example plot for **Q2**

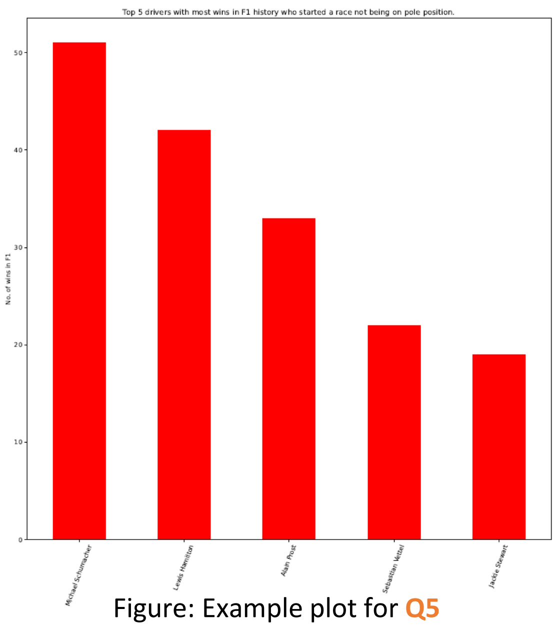

Figure: Example plot for **Q5**