Final Project

ENDG 233 Fall 2021

Department of Electrical and Software Engineering
Schulich School of Engineering
Due: December 9, 2021 at 11:59 PM

The purpose of this final project is to demonstrate your understanding of programming basics, Python syntax, data manipulation, and data visualization. This assignment is worth 25% of your final grade.

Final Project – Learning Outcomes

- Design and document a terminal-based Python application
- Select, import, and manipulate a set of data
- Create and operate on data using numpy arrays
- Process data according to user input
- Display data using matplotlib

Project Logistics

It is recommended that you work with a partner for this project, however you may also choose to work individually. You may not switch partners or move to individual work after the project has begun. Project expectations are the same for pairs or individuals. Both members in a pair will receive the same grade. Your partner must be in the same block as you.

Once you have chosen a partner or chosen to work individually, you must self-enroll in an empty Group on D2L. Go to Communication -> Groups. Find your block number and choose from either the “Individual” or “Pairs” category. Select any open number and self-enroll. This will be your group number. If working with a partner, both students must be enrolled in the same group. **All individual and group enrollments must be complete by November 22nd at 8:00 am**.

There is a pinned thread under the Final Project discussion topic that you can use to find a partner. Post a reply in the thread if you are looking for a partner. Remember to include your block and how prospective partners can get in contact with you. Once you are no longer looking for a partner, please go back and update your post.

Pair Submissions: When submitting, ensure that both names are listed in the description. Only one group member needs to submit, but it is recommended that you complete the submission process together to double-check that nothing is missed! **If the submitting group member uploads the wrong files, it will impact the grade of both members.**

Program Specifications

For this final project, you have flexibility to design and develop your own terminal-based data analysis and visualization program in Python.

Your application must meet the following design specifications:

Stage 1: Dataset Selection and Importing

- A suggested dataset is included on D2L. You may use the provided data or select data of your own choosing.
- You must use at least three separate csv files that can be related in some way.
- Each csv files must have at least three columns and at least 50 rows of data.
- You may edit the datasets before you begin coding, but your program should not modify the csv files directly.
- You may not hard-code any data values within the program- all information must be read in. This means that a TA could change a value or header name in your dataset and still get the desired results.
- You must create at least three separate numpy arrays using your chosen dataset.
- You may not use global variables. You must import the data within your main function.

Stage 2: User Interface and Analysis

- Your application must return useful information. Design an interface that allows users to search based on some sort of criteria or keywords.
- The user must provide at least two pieces of selection information (e.g. "school name" and "average grade") that is used to calculate/sort/filter, etc. and return the results.
    - For example, you could prompt the user to provide a country name, and then choose which statistics they would like to output (average number of threatened species, delta population change over time, etc.). Or search by region and then identify which country in the region has the min/max number of threatened plants, etc. This is your opportunity to get creative!
- There is a minimum expectation that your user input will be used to manipulate the data, not just select and print original data.
- Give the user clear input instructions. If an invalid entry is given, notify the user and allow them to re-enter the information. Your program should not terminate.
- All output information must be clearly defined and formatted using print statements. Consider formatting your data into a nice-looking table!

Stage 3: Analysis Syntax

- You may choose what data trends to calculated and present from your data. However, you must use the built-in numpy methods to calculate a min, max, or mean at least twice.
- Your code must include and use at least two user-defined functions that have parameters passed in.
- Your code must include and use at least one user-defined class that contains a constructor and at least one instance method.

Stage 4: Data Plotting

- Use your data to create at least two separate figures using matplotlib. These plots should be output at the end of your program.
- All plots/subplots must be clearly readable include the following:
    - Title
    - Axis labels
    - Legend

Additional Specifications

- Your code must follow the conventions discussed so far in the course (names_with_underscores, ClassNames, four spaces for indentations, spaces between variables/operators, comments throughout, etc.)
- All classes, methods, and functions must contain docstring documentation.
    - For each class, include a functionality summary and describe any class and/or instance variables (do not include a separate docstring for constructors)
    - For each method/function, include a functionality summary and describe parameters and return values (or specify if there are none)
    - Your main functions does not need a docstring but should be well-commented
- You may go beyond the minimum requirements specified here!
- You may only use built-in Python functions that support compound data structures, user entry, or casting (such as len(), input() or int()). The only modules you may import for analysis are numpy, matplotlib, and math. You may use other modules such as pandas or csv for importing the data only. These requirements should be clearly indicated in the header comments.
- Your code will be run by the TAs as your end user using VS Code and a minimum of Python 3.9.
- FAQs about the project will be answered on the D2L discussion boards. Please check the boards for any clarifications before submitting.
- The grading rubric will be posted separately to D2L.
- Screenshots of an example interface are provided at the end of this handout. They are not meant as a template- they are just an example of how you could prompt the user for input.

Assignment Tasks and Deliverables

- You must submit the following items via the D2L dropox. Upload all the files in a single submission. Do not zip your files.
    - Your python code file
    - Your input dataset (your TAs will need the data csv files to run your code)
    - A brief PDF summary (ideally less than two pages) that gives an overview of your project and how it meets the requirements. You should also include a citation of your dataset source. Your summary should also include:
        - Evidence of your design process and planning (flowchart, pseudocode, mind-map, etc.)
        - Evidence of task management and timeline of milestones
        - Screenshots of successful execution as PNG files
        - Both plots saved as PNG files

  - You do not need to create an instruction file- your user input instructions should be clear from your interface, but you may put any introductory information directly in the comments under your code header.
  - You will also give a 2-minute beta demonstration during the short lab from Dec. 6th to 9th. All team members must participate. Be prepared to answer questions about your program. You must have at least some analysis functionality to demonstrate.
- Make sure to watch all video lessons and review the corresponding active learning content.
- Open VSCode and start a new terminal. Make sure that your virtual environment is activated. If your module imports cannot be resolved, check that your virtual environment is selected as the Python interpreter.
- Create your `.py` file and remember to place your information in the header.
- Remember to test your program execution via the terminal.
- Your screenshot of successful program execution should also show that your virtual environment is active when running the program
- When submitting, make sure you list your name and block number in the description. If you are working with a partner, ensure that both names are listed. **Only one group member needs to submit, but it is recommended that you complete the submission process together to double-check that nothing is missed!** If the submitting group member uploads the wrong files, it will impact the grade of both members.