Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm

Python Project

Introduction.

This project involves developing Python based code to solve problems of interest to engineers. The purpose of this project is to assess the student's ability to utilize programming in the solution to engineering related problems. This is a group assignment and, as such, each student is responsible for contributing to the project. Please review the course syllabus for a full description of what differentiates an individual assignment from a team assignment.

Project Overview

Your team is tasked with developing several programs/functions that will allow the instructional team to evaluate your python program skills. Your code should follow the concepts developed in class and follow the commenting and variable naming requirements emphasized throughout the semester. Your grade will be based on the following criteria: (1) Producing the correct result 50%, (2) Commenting and variable naming (20%), and (3) Overall participation to the group effort as measured by the team (30%).

The project will require functions to be defined to carry out the following:

1. Handle Numeric Data
    a. Load numeric data
    b. Plot the data
    c. Interpolate
2. Evaluate Functions
    a. Load and function from file
    b. Plot the function
    c. Integrate the function
3. Process Text Files
    a. Word counting
    b. String search
    c. String search, convert, and plot

1. Task Description – Handle Numeric Data

For this task, the user will create functions that will load numeric data and perform simple operations that we have gone over in class. The deliverable will include the function file and a set of test problems/values and their corresponding results.

Function 1.a – Load Numeric Data

You are tasked with creating a function to load numeric data. The data will be in column form where each column is a variable and each row is an observation. Example files are provided but it is the users responsibility to generate new files for the deliverable. The data will utilize different delimiters for separating the variables and contain two header rows where the first row is the variable name and the second row is the units. Data will either be several variables with the first column corresponding to the independent variable and the remaining columns corresponding to the dependent variables or will be one column. The function will have the following properties:

- Name – the function name will be engr102_load_data

Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm

- Arguments – the function will take multiple arguments
    - File name as a string
    - Delimiter as a string
- Outputs – the function should return 3 things in a single return statement
    - A string indicating whether there is one or several variables returned (either 'one' or 'multiple')
    - A single, one dimensional list of strings containing the variable name and the units i.e. 'Mass (kg)' returned
    - A list of appropriate dimension containing all the numeric data (list of lists where each sublist contains the numeri data for a variable)
- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.
    - Test files – should include files of different number of variables and different delimiters

Function 1.b – Plot Numeric Data

You are tasked with creating a function to plot numeric data. The numeric data files of interest will either be comma separated variables or tab separated. One can assume that comma separated variable data uses the .csv suffix and tab separated data uses the .dat suffix. The data will consists of columns corresponding to different variables and rows corresponding to observations. Each data file will have two header rows that consist of a variable name (row1) and the appropriate units (row2). Sample data are included as part of the project package but testing will utilize data files of similar format but varying in size (rows and columns). Use the first column variable name and units as the x-axis label for either plot.

For a single column array, the function should produce, display, and save a figure that contains a histogram plot. The variable name and units should be used as the x-axis label and the histogram should use frequency as the y-axis. The mean and standard deviation should be written on the figure as a text box in the upper right corner of the plot area. The plot should be titled 'ENGR102 Project 1.b Histogram'. Save the file as a .png file with the same name as title.

For an array with several variables, the first column corresponds to the independent variable and the remaining columns corresponding to the dependent variables. The plot should use markers (not lines) of different colors corresponding to each dependent variable. A legend should be included that lists the variable names. The y-axis should be labeled 'dependent variables'. The plot should utilize gridlines and be titled 'ENGR102 Project 1.b Plot'. Save the file as a .png file with the same name as title.

The function will have the following properties:

- Name – the function name will be engr102_plot_data
- Arguments – the function will take multiple arguments
    - A list corresponding to variable names and units as strings
    - A two-dimensional list of numeric floats
- Outputs
    - A histogram or scatter plot displayed to screen and saved as a .png file

Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm

- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.
    - Test files – sample plots based on test files used in 1.a

Function 1.c – Interpolate Numeric Data

You are tasked with creating a function to linearly interpolate numeric data. The data will be in the form of a two-dimensional list where each column is a variable and each row is an observation. You can assume that the first column is the independent variable and the user will have to select based on a menu prompt on which column to use as the dependent variable. The user should then be prompted to enter a new query for the independent variable and the function should return the resulting value corresponding to the dependent variable. The function will have to identify known points closest to the query value and use those as the known interpolation points by fitting a line between closest points. If the data has only one column, a warning should be displayed to the screen and nothing returned from the function (does not throw an error).

The function will have the following properties:

- Name – the function name will be engr102_interpolate_data
- Arguments – the function will take multiple arguments
    - A list corresponding to variable names and units as strings
    - A two-dimensional list of numeric floats
- Inputs – The user will be prompted for these – do not use them as input arguments
    - A selection from a menu displayed of possible independent variables
    - A query value for the independent variable
- Outputs
    - Returns The resulting dependent variable value corresponding to the input query returned as a float
- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.
    - Test file and a test query with the resulting value returned

2. Task Description – Evaluate Functions

For this task, the user will create functions that will load functions as strings from a text file and perform simple operations of interest to engineering. The deliverable will include the function file and a set of test problems/values and their corresponding results.

Function 2.a – Load Equation

You are tasked with creating a function to load an equation in python syntax similar to the one below from a file:

```
3*(x**2) - 2*x + 1/2
```

Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm
A file of equations will be provided as part of the project files where each line corresponds to a different equation. The functions may use the math or random module.

The function will have the following properties:

- Name – the function name will be engr102_load_function
- Arguments – the function will take a single argument
    - File name as a string
- Outputs
    - Returns A list of strings corresponding to each equation returned
    - Each equation printed to the screen
- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.

Function 2.b – Plot the Equation

You are tasked with creating a function that will plot each of the equations over a range x = [0,10]. Each function should include a label shown in the legend corresponding to the line number i.e. Eq. 1, Eq. 2, etc. Each equation should be represented by a different color and/or different line style and be a line (no markers). The plot should include the following items; gridlines, x-axis label (x), y-axis label (y), title (ENGR102 Project 2b), and the range of the x-axis should be from 0 to 10.

The function will have the following properties:

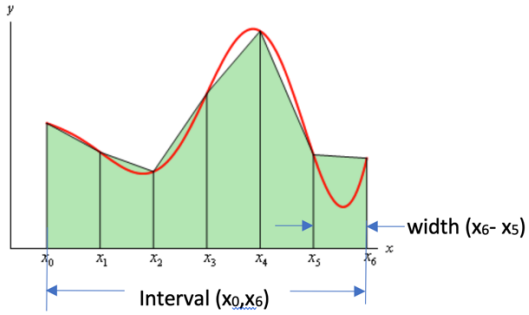
- Name – the function name will be engr102_plot_function
- Arguments – the function will take a single argument
    - A list of strings corresponding to the equations in python syntax
- Outputs
    - A plot displayed to screen and saved as a .png file (use title as prefix)
- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.

Use of the eval() function will be needed. https://docs.python.org/3/library/functions.html#eval

Function 2.c – Integrate Equation

You tasked with developing a function that will calculate the area under a curve defined by a equation, f(x). The function will take in four arguments: (1) a list of strings where each element of the list is a separate equation; (2) a tuple that corresponds to the limits of integration; (3) an integer that will be used to determine the number of trapezoids; and, (4) an integer that will select the equation of the list to integrate. The function should utilize the trapezoid rule to numerically integrate the selected equation over the limits of integration. See https://en.wikipedia.org/wiki/Trapezoidal_rule for more information. The Figure below illustrates the approach. The function is shown in red and the green trapezoids are summed to estimate the integral. As the subinterval gets smaller (number of trapezoids gets larger), the summation of the trapezoids will more accurately estimate the integral of the function.

Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm
The output of the function must include the calculated area (the estimated result of the integral) returned to the calling program.



Use of the eval() function will be needed. https://docs.python.org/3/library/functions.html#eval

The function will have the following properties:

- Name – the function name will be engr102_integrate_function
- Arguments – the function will take in four arguments
  - A list of equations as strings
  - A tuple corresponding to the limits of integration (start with the lowest value)
  - An integer corresponding to the number of trapezoids
  - An integer corresponding to the equation (index of list above) to be integrated
- Outputs
  - The resulting summation of the trapezoid areas returned as a float
- Deliverable
  - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.

3. Task Description – Handle Text Files

For this task, the user will create functions that will load and perform simple operations that we have gone over in class. The deliverable will include the function file and a set of test problems/values and their corresponding results.

Function 3.a – Word Counting

You are tasked with creating a function to count the number of words in a document as well as the number of words that match a user supplied list. Example files are provided to test your function but one should generate additional files for testing. A word is considered a single distinct meaningful element of speech or writing, used with others (or sometimes alone) to form a sentence and typically shown with a space on either side when written or printed (not considering punctuation). The function should NOT be case sensitive.

The function will have the following properties:

- Name – the function name will be engr102_word_count
- Arguments – the function will take multiple arguments
  - A file container object of the file we are interested in counting

    - A list of strings corresponding to words to be count in the file
- Outputs – the function will return 2 things in a single return statement
    - Total number of words returned as an integer
    - A list of integers corresponding to the number of times (count) each element of the input list is found in the file of interest returned
- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.

Function 3.b – String Search

You are tasked with creating a function to locate a given string of length greater than or equal to one in a larger document. The function should return the number of occurrences of the given string as well as the line number the string is found. The function should NOT be case sensitive. Example files are provided to test your function but one should generate additional files for testing.

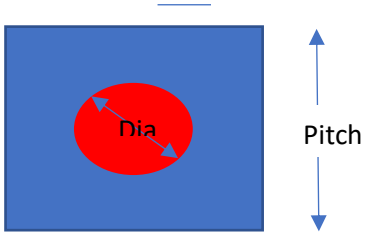
The function will have the following properties:

- Name – the function name will be engr102_string_find
- Arguments – the function will take multiple arguments
    - A file container object of the file we are interested in searching
    - A string corresponding to search string
- Outputs – the function will return 2 things in a single return statement
    - Total number of occurrences of the string returned as an integer
    - A list containing the line numbers of each occurrence of the string returned
- Deliverable
    - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.

Function 3.c – String search, convert, and plot

You are tasked with evaluating the output of a nuclear core design production code. Essentially, a core designer needs to evaluate the multiplication factor as a function of ratio of the volume of moderator to the volume of fuel. By changing the geometry of the fuel pin size and spacing, the moderator to fuel ratio can be changed. The production code then calculates the multiplication factor which is then plotted versus the moderator to fuel ratio for analysis.

The geometry for a nuclear fuel assembly can be simplified to the figure below:



Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm

The figure shows a square cell with a side length equal to the pitch surrounding a circular fuel rod with a diameter shown as Dia. The moderator to volume ratio is simply the ratio of the areas.

$$\frac{\text{mod}}{\text{fuel}} = \frac{\text{Area of Moderator}}{\text{Area of Fuel}} = \frac{4P^2 - \pi D^2}{\pi D^2}$$

Your function will take in a string that corresponds to a text file that contains all of the output file names produced from the production code (The output files will be provided). The function will need to open each of these files, find the value for the pitch, fuel diameter, and multiplication factor, calculate the moderator to fuel ratio, and return the moderator to fuel ratio and multiplication factor as a two-dimensional array. The location of the pitch, fuel diameter, and multiplication will be highlighted in a file that you can use for testing.

The function will need to produce a figure that contains a plot of the multiplication factor as a function of the moderator to fuel ratio. The plot should include axis labels, gridlines, and a title. The line should be a red dot-dashed line with no markers. The scale for the x-axis should run from 0 to 20 in increments of 2 and the y-axis scale should range from 1 to 1.5 in increments of 0.05. The plot should include minor tick marks. The plot should be displayed to the screen as well as saved as a .png file with the name 'mod-to-fuel_vs_k.png'.

- Name – the function name will be engr102_nuclear_results
- Arguments – the function will take one argument
  - A string corresponding to a file name of a file that contains a list of file names as new lines that contain the production code results
- Outputs
  - Returns A two-dimensional array of moderator to fuel ratio (column 0) and the multiplication factor (column 1) returned
  - A plot of the multiplication factor as a function of the moderator to fuel ratio outputted to the screen and saved as a .png file.
- Deliverable
  - The function will be part of a larger file called engr102_project_[Group #]_[sec #].py – my submission would be engr102_project_99_525.py. Your file should contain a header as described in class.

General Comments

Limited or no commenting of the code will result in a significant grade deduction. The function file will require a header that will include all team members that participated in the project and a mutually agreed upon weighting of their contribution. The weighting should be a percent of the total project so the weighting should sum to 100%. The absence of a team member will require further documentation that can be discussed with the instructor.

Grading

The teaching team will use a set of test problems to evaluate each of the functions. PLEASE NOTE: THE ORDER OF ARGUMENTS AND OUTPUTS MATTER. Please order from left to right corresponding to the bullet order (first bullet, first argument or output). The grading will be as follows: 50% - produces the correct output; 20% - Code is commented and uses proper variable naming; and, 30% – Contribution to

Programming Project v6 – ENGR102 All Kurwitz Sections
Assigned: 4/05/19
Due Sunday 4/28/19 – 11:59 pm
team effort (Example: a team member on a four person team that was given a weight of 20% would receive (20/(100/4))*30% = 24%).

Supporting Files

1. Files for Part 1

- steam.csv
- sine.csv
- poly.csv
- temp.csv
- para.dat
- trig.dat
- volts.dat
- height.dat

2. Files for Part 2

- equations.txt

3. Files for Part 3

- un.txt
- doi.txt
- example-data.rtf – A rich text file where the dia, pitch, and multiplication factor are highlighted
- archive.zip – a zip archive of all production code output to use for 3.c