ENTS749C (Spring 2015): Vehicular Networks
Mini Project #1 (100 points)

All ENTS 749C mini-projects must be completed by each student individually and must be your own work. *Please note that if plagiarism is detected in your solutions, all the students involved in it will be subject to the appropriate university regulations.*

Carefully read the Project Description as well as the Appendix for useful Python modules you are required to learn and use in this project. Your solutions for this project are <u>due no later than 5pm, March 13 (Friday)</u>. Late submission policy is strictly enforced and will be as follows: up to 1-day late: 25% reduction (hence, max. score you can get is 75 points); between 1- to 2-days late: 50% reduction (max. is 50 points); more than 2 days late: 100% reduction (no points).

PROJECT DESCRIPTION:

Problem: Alice drives her Controller Area Network (CAN) enabled car. She has volunteered to make her vehicle's CAN bus data available to an insurance company XYZ for the purpose of earning the "safe driver" price reduction. Her vehicle data can be used to create position traces and monitor some of vehicular system parameters when moving on these traces.

XYZ obtains the data file called `alicedata.json`. Now XYZ has to mine this data file to assess the scenario that Alice's vehicle went through and learn her driving behavior. You work for XYZ and are a Python expert. You must use Python to process this data file and derive useful information that helps make decisions on the driver behavior.

What Python Code Must You Develop?

Create a single Python module named `yourlastname_proj1.py` to contain your Python functions `func1` to `func8` corresponding to questions (1) to (8) below (note: `yourlastname` is your last name).

(1) Write a Python function `func1` that reads a data file is in JavaScript Object Notation (JSON) as a Python list of Python dictionary items. The data file must be an input parameter, and the returned object must be a Python list. Hint: consider using the Python built-in function called `with` and the built-in Python module called `json`, along with any other needed Python functions, to read the JSON file line-by-line.

(2) Write a Python function `func2` to legibly print the first 10 signal entries of the data file; each entry contains the vehicle signal name, timestamp, and signal value. Note that the `print(data)` does <u>not</u> provide a legible "pretty" print; you need to find another function.

(3) Write a Python function `func3` to automatically identify all of the different signal names in the data file, print these signal names, then prompt the user to enter a signal name, and finally return the number of occurrences and value range of the user-entered signal.

(4) Write a Python function `func4` to find the vehicle trip time period and vehicle trip distance over which the data file was recorded?

(5) Write a Python function `func5` to plot each signal vs. trip time; such as vehicle speed, engine RPM, and other signal types versus time as the x-axis.

(6) Write a Python function `func6` to compute maximum and average speeds of Alice's vehicle?

(7) Write a Python function `func7` to identify where in the world Alice was driving and plot her vehicle's position (latitude and longitude) trace on a Google Map? Hint: consider using `pygmap` module available at: https://code.google.com/p/pygmaps/

(8) (Open-ended question): Given the data in the data file, what other function `func8`, different from the above 7, would you implement and what is its purpose/output?

How to Test Your Python Code?

Under the Python interactive mode/command line you must be able to type a sequence of commands to test if your functions work. An example is given below. But note that may you choose to pass more input parameters for any of the 8 functions (including any default valued parameters); you may even pass the data file to the function if appropriate. It's your code and you have the flexibility to design it as you see fit.

```
>>> import yourlastname_proj1 as pp
>>> a=pp.func1('alicedata.json')
>>> pp.func2(a)
….here is the legibly printed output….
>>> pp.func3('alicedata.json')
:
>>> pp.func8('alicedata.json')
```

Also, note that it may be wise to create a small `testdata.json` file (with 5-10 entries only) for initial testing of your functions.

What Report Must You Document?

This will contain four sections: Assumptions, Python-based Data Analysis, Data Plots, and Observations. Each section is described below.

- Assumptions: In this section, include all the assumptions you make about the problem, data, and its analysis within this section as bullet points.
- Python-based Data Analysis: In this section, simply include the command sequence you entered in Python command line/interactive mode (you can use the Python IDE window) to analyze Alice's data. Note this will include the functions you developed for Questions (1)-(8). Also, include the interpreter output (if any) for each of the commands in the sequence. You can include the snapshot of the Python IDE window or document the text of the commands and outputs.
- Data Plots: In this section, include all plots/graphical outputs of `func5` and `func7` (and `func8` if needed).
- Observations: In this section, use the information learnt from processing the data file from Alice's vehicle to answer the following questions:
    a. What can you tell about the vehicle's trip? Such as how long did it stay in idle, how fast did it go in the trip, and where in the world did it make its trip.
    b. What do you know about Alice's driving behavior? Base your judgment mainly on the temporal plots obtained from question (5) above.

What Material Must You Submit?

- Project report as a PDF file
- Python code: a file titled `yourlastname_proj1.py` containing the functions `func1-func8`

How is the Project Graded?

- Solutions to Questions (1)-(8): 80%
- Project Report: 10%
- Python Coding: 10% (In your scripts, make sure you follow Python conventions/style such as variable/function naming, script commenting, and coding style, etc.).

APPENDIX:
Plotting functions in Python
Apart from the modules referenced in the project description above, the project requires you to use Python modules including `matplotlib` and `numpy`. You must learn these modules and use their functions as appropriate for the given project. Note that the project includes a percentage of points for Python coding style; use of modules will result in a lesser number of lines of code in your functions which improves code style/efficiency. Instructions to install `matplotlib` and `numpy` are as follows:

- Install `matplotlib` module
    - This module contains useful functions for plotting in Python (similar to Matlab)
    - Go to: http://matplotlib.org/ to see documentation and downloadable exe.
    - Use matplotlib-1.4.3.win32-py3.4.exe; or another version suited for your PC/Python.
        - To check version of the Python in your PC, use Python commands:
          ```
          >>> import platform
          >>> platform.architecture()
          ```
- Install `numpy` module
    - This module (abbreviation for Numerical Python) contains useful functions for array and math manipulations; `matplotlib` depends on the `numpy` module.
    - Go to http://sourceforge.net/projects/numpy/files/NumPy/1.9.2rc1/
    - Download and install: numpy-1.9.2rc1-win32-superpack-python3.4.exe
- Successful installation of the above modules can be tested using the following Python commands and their corresponding outcomes:
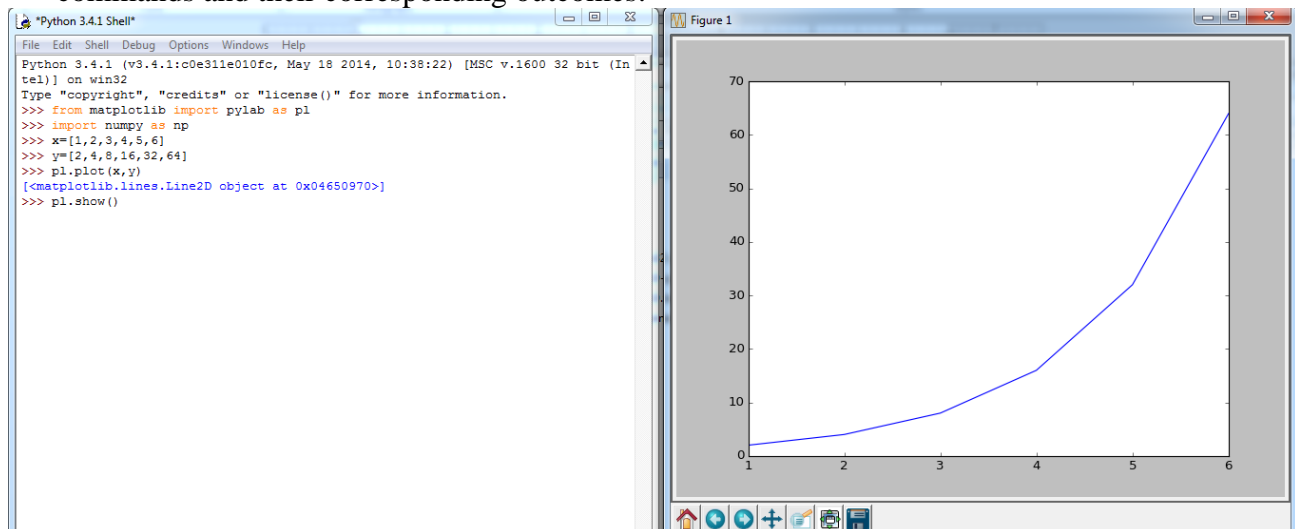
- If Python interpreter outputs any error, you will need to debug the installation errors on your own using the guidance provided at http://matplotlib.org and http://numpy.org.
    - Note: Based on your computer and its OS, you may need to make additional configuration changes to make use of matplotlib. This may include installing additional open-source software, such as datelib.py, six.py, pyparsing.py, Microsoft Visual C++ Express 2010, which may not be available in your Python installation. Most of these platform specific changes are documented at http://matplotlib.org/ . Feel free to discuss these configuration issues with other students.

- In addition, you may find it useful to see the <u>tutorial on plotting with Python</u> (http://www.ast.uct.ac.za/~sarblyth/pythonGuide/PythonPlottingBeginnersGuide.pdf)