READ ME

In this project, I have analyzed data in the form of Web Log files. Also, I have visualized the analysis to make implications easier.

The code contains intuitive comments which should facilitate easy understanding of the functionality performed.

**Dataset**: USNA Introductory Physics Spring 2010 <https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=409>

Raw Log File: Spring2010-USNA-McClanahan-1126A-anon   
Anonymized Andes raw logs, instructor McClanahan, Section 1126A

**Tool used to analyze data**: Spyder (Python 3.6): It is an open source, cross platform IDE which supports libraries like NumPy, SciPy, Matplotlib. It also has an embedded IPython console with it

**Steps performed:**

A] Data Munging

* Extract the log data to just the lines containing DDE and DDE-Post as we are only concerned with these commands.
* Extract all commands containing read-student-info and close-problem and put it in a new Data Frame “studentData” to precisely contain information about the student
* As the student ID is part of the read-student-info command, Extract the substring to make a new column “student” which contains the student ID.
* As the time is a string, Convert it into datetime format. The time is in format HH: MM: SS as well as MM: SS. Hence, a condition was required to process it accordingly. For this conversion, datetime library had to be used
* To find out the time a student spent in one session, Take the time difference in between read-student-info and close-problem command.
* To find out the total time spent by a student, add up all the time values grouping by the student.
* The time difference is in datetime format, hence calculate the number of minutes by using “seconds” attribute of datetime and put it in a new data frame named “totalTimeOfStudent” which contains student ID and total time he spent in minutes

After making the raw log data ready for analysis, following actions are performed to gather more data about the student performance:

* Calculate the number of actions performed by taking a difference in index of read-student-info and close-problem command. As it indicates the total actions performed in between reading student info and closing the problem.
* Calculate Actions performed per minute by diving number of actions by total time taken and store it in a new Data Frame “numberOfActions”.

B] Data Visualization

* Install plotly: Please see the instructions to run the code for plotly installation instruction
* Create a plotly graph objects named as “apm” for the scatter of actions per minute, the x axis is

student, and the y axis is actions. Then create a list named” data1” to store the apm data.

* Create a dictionary “layout” to define the graph’s names of title, x axis and y axis.
* Create a dictionary “fig” to define the data and layout refer to
* Output the graph into local file
* Create a plotly graph objects named as” Ttim” for the scatter of total time in minutes, the x axis

is student id, and the y axis is the total time in minutes. Then create a list named as “data2” to

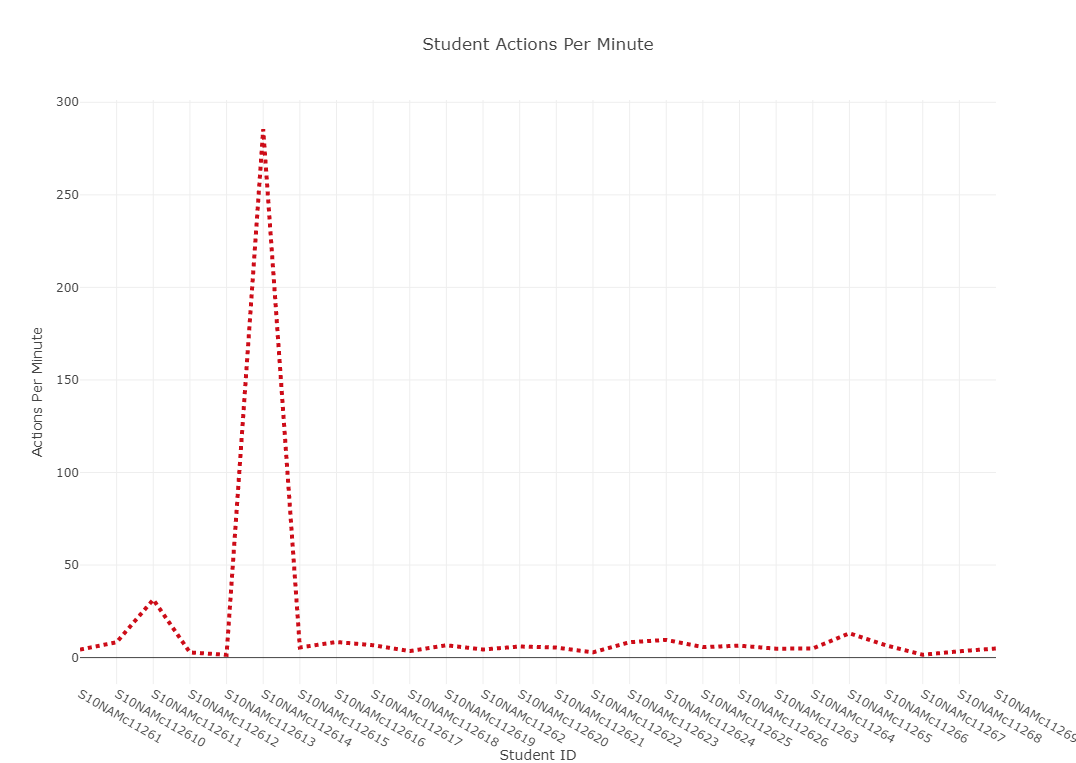
store the ttim data.

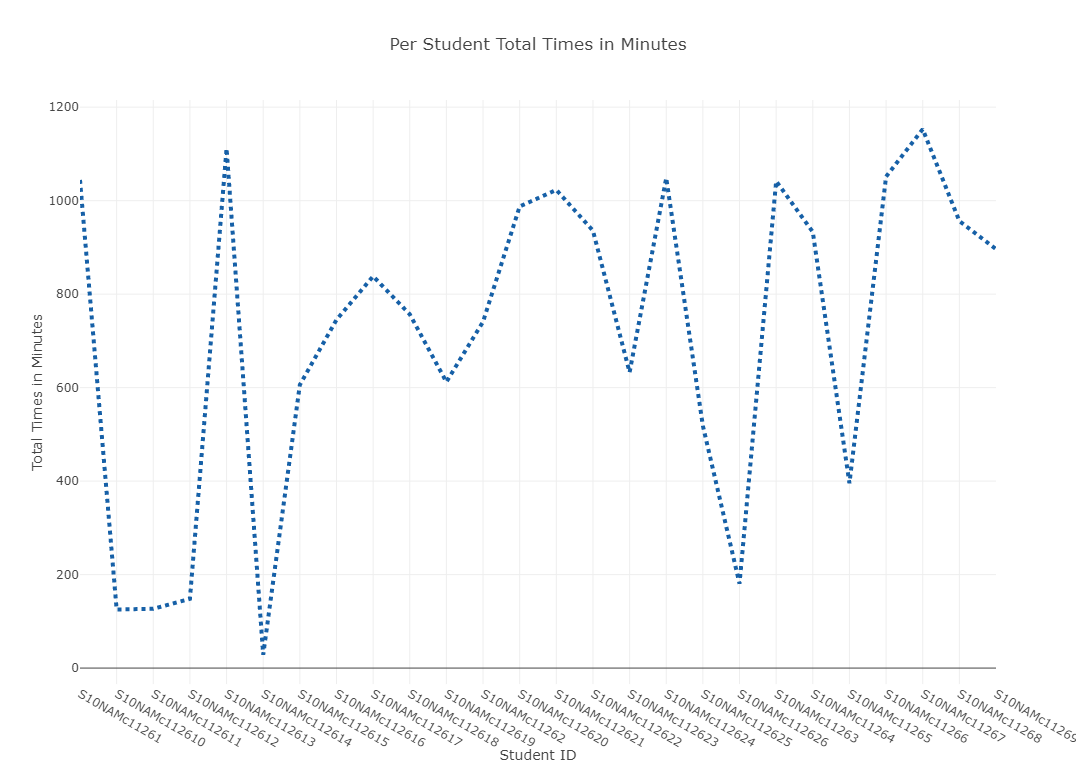
* Repeat the ‘layout’ and ‘fig’ step and output the graph into local file

**Instructions to run the code:**

* If the version of python > 3.4, there is already the pip.exe in the ‘C:\ProgramData\Anaconda3\Scripts’ , so there is no need to install pip.
* If the version of python < 3.4, then download the pip.exe file from https://bootstrap.pypa.io/get-pip.py
* Install plotly by running the command ’pip install plotly’
* Run the code
* The output graph will be downloaded in the downloads and it also gets opened.

**Visualization:**





**Conclusion:**

It is very easy to derive conclusions or findings of an analysis of the data by visualizing it. This fact can be proved from above reults. As you can see, Actions per minute for all students are all in one range but for one particular student(112614), it’s too high. Hence, we can omit that data.

Similarly, Total time in minutes for student 112615 is too less to be true.

Similarly, many such implications can be derived by analyzing a graph easily rather than directly analyzing a data