Project – Mozilla Bug Report Analysis

This project asks you to analyse bug reports that were collected from the Mozilla open source project. The frequency of reported bug reports over time give us an indication of how the reliability of the software is changing. Two sets of bug reports are provided to you from the Mozilla project. One set corresponds to the bugs reported for version 1.4 and the other for version 1.7. Download the zip files, put them in two separate folders and unzip them. Each bug report in the zip files are provided as text files. Then you will parse the text files, clean the data, and do some analysis on the collected data.

- ★ Although the filenames seem to be numbered in an order, some are missing. Therefore, use the **os.listdir()** method to get the list of filenames to open.

- ★ You need to first do some data cleaning and preparation. Open and evaluate one of the bug report text files. You will see that each text file has a title line which is made of the bug id, and the summary of the bug. Then, the attribute names and their corresponding values are provided in each line. You should parse the text files to get the attribute - value pairs and create a DataFrame to store the parsed data, where data from each bug report corresponds to a row in the DataFrame and the attributes corresponds to each column.

1. The “opened” attribute provides the date the bug report was created. Convert the date and time values in the opened column to datetime datatype to be able to process it using the datetime methods.

2. You will see that some reports are for enhancements or trivial bugs. Also some bugs are Duplicates or invalid. Write a function to create a new data frame that consists of only reports more severe bugs and valid unique bugs. To do this, you need to eliminate the rows where severity is equal to enhancement, trivial or minor. You also need to eliminate bugs where the resolution is Duplicate or invalid. Use Boolean masking to perform these operations efficiently.

3. To see how frequently the reports are arriving calculate the inter arrival times for bugs. Inter-arrival time is the time between two bug arrivals. You can calculate the inter arrival times in minutes. Don’t forget to sort the data by opened column first.

4. Plot the interarrival times against the opened date.

5. Calculate the number of bugs per day and per month. Then create plots for them.

6. Write a function that will display average bug inter-arrival times for the data grouped by the passed column name. For instance, it should display the average inter-arrival times for each bug severity type.

7. Ran t-tests to see if the average inter-arrival times for each of the data sets (version 1.4 and version 1.7) are different.

In your project please pick and follow a coding style convention. Name your variables and functions carefully. Also don't forget to document your code. Use docstrings below each function definition to describe the purpose of your function. Use comments when necessary. Avoid global variables.

You may find s summary of the coding conventions used in the Introduction to Scripting in Python series as a reading in week 3 of the Python Programming Essentials course under the Programming Tips and Practice section.

Submission:

To submit the homework, you need to send the files in **.ipynb** format. The deadline for the homework is **16th of September, 10.00am**. You can send the homework files to **veribilimi2018@ozyegin.edu.tr**.