Principles of Urban Informatics
Assignment 3

Posted on: 09/22/2014
Due Date: 09/29/2014

Data description

In this assignment, we are going to use tweets as our datasets. The general idea is to review basic concepts of Python, like lists and dictionaries, and make sure that you are all familiarized with those data structures.

The datasets are csv files that store, in each line, a public tweet from a user:

```
@alyssaprager,Thu Sep 18 18:03:40 EDT 2014,-73.98533,40.759337,#NewYork
,#TimesSquare
```

The first column is the screen name of the user in Twitter (alyssaprager in this case). The second column is the date and time of the tweet (Thu Sep 18 18:03:40 EDT 2014). The third and fourth columns are the longitude and latitude of the user when he tweeted. Starting from the fifth column, we have the hashtags used by the user in his tweet; in this example, the user tweeted two hashtags: #NewYork and #TimesSquare. The number of hashtags is arbitrary per line.

All the tweets were captured using Twitter's public streaming API. More informations here: `https://dev.twitter.com/streaming/public`.

All outputs need to be exactly as in the examples. Pay attention to date formats, spaces and punctuation. In this assignment you'll get zero if the output is different from the expected. A tip: you can use a tool to compare your output with the sample output provided. In unix systems you can use the *diff* command to check your output against the expected one. For windows, you can do this using the *FC* command.

For all problems, your solution should receive command line parameters with the input files names, so it **should not depend** on internet connection to run. Download the sample files to your computer and point to them to execute your solutions.

You *cannot* use Pandas in this assignment.

Problem 1

Create a Python file that reads each line of the dataset `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_`

1.txt and computes two things: number of tweets in the csv file and range of dates. Your output should be in the format:

The output of the sample is available at http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_1.txt. Note: attention to the date format, spaces and punctuation.

Problem 2

Compute the number of *unique* users in the dataset and also the range of dates in the dataset. For this problem, use the dataset http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_2.txt. Your output should be in the format:

```
> python problem2.py sample_data_problem_2.txt
49 users tweeted between September 19 2014, 21:00:18 and September 19
2014, 21:00:29
```

The output of the sample is available at http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_2.txt.

Problem 3

Given two datasets, output all hashtags that happen in both datasets. The hashtags must be unique (in other words, no two hashtags outputed can be equal), and they are sorted in lexicographical order. For this problem, use the datasets http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_3_1.txt and http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_3_2.txt. Your output should be a list of all hashtags sorted in lexicographical order, one hashtag per line.

The output of the sample is available at http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_3.txt.

Problem 4

Compute the 10 most popular hashtags in the dataset. The hashtags must be sorted by popularity. For this problem, use the dataset http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_4.txt. Your output should be a list of all hashtags sorted in popularity, with their count value, one hashtag per line. If the count of two hashtags are the same, then they should be sorted in lexicographical order.

The output of the sample is available at http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_4.txt.

Problem 5

Compute the date and time when most of the tweets happened. Consider seconds as the time granularity here. For instance, if 10 tweets happened on Thu Sep 18 18:03:30 EDT 2014, and 11 tweets happened on Thu Sep 18 18:03:31 EDT 2014,

then Thu Sep 18 18:03:31 EDT 2014 is the date and time when most of the tweets happened. For this problem, use the dataset `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_5.txt`. Your output should be in the format:

```
> python problem5.py sample_data_problem_5.txt
September 19 2014, 21:39:19 with 13 tweets
```

The output of the sample is available at `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_5.txt`.

Problem 6

Now, consider that the granularity of the time is hours. For instance, if 10 tweets happened on Thu Sep 18 18:03:30 EDT 2014, 11 tweets happened on Thu Sep 18 18:03:31 EDT 2014, and 15 tweets happened on Thu Sep 18 19:05:30 EDT 2014. Then, 18:00 of Sep 18 is the time and date when most of the tweets happened. For this problem, use the dataset `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_6.txt`. Your output should be in the format:

```
> python problem6.py sample_data_problem_6.txt
September 19 2014, 20h with 15996 tweets
```

Note that we are only outputting the month, day, year and hour, because our time granularity is hours.

The output of the sample is available at `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_6.txt`.

Problem 7

Now, output the user who tweeted the most in the dataset and also the date and time range of the dataset. Note that it doesn't matter if the user tweeted 1 hashtag or 10 hashtags; in this problem, we are ignoring the number of hashtags used by the user. For this problem, use the dataset `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_7.txt`. Your output should be in the format:

```
> python problem7.py sample_data_problem_7.txt
@trendinaliaUS tweeted the most
Dataset range: September 19 2014, 21:00:18 and September 19 2014,
21:00:29
```

If two users have the same number of tweets, you should use lexicographical order to break the tie.

The output of the sample is available at `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_7.txt`.

Problem 8

In our datasets, we also have the latitude and longitude of the tweet. We will calculate now the 5 most popular hashtags in two different cities: New York and San Francisco.

We can consider that New York is inside a box, in wich the bottom-left point has the coordinates (-74.2557, 40.4957) and the top-right point has the coordinates (-73.6895, 40.9176). San Francisco can also be considered inside a box, with bottom-left point as (-122.5155, 37.7038) and top-right point as (-122.3247, 37.8545).

For this problem, use the dataset `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_data_problem_8.txt`. Output the 5 most popular hashtags of New York, with their count, and then do the same for San Francisco. The hashtags must be sorted by their count value. If the count of two hashtags are the same, then they should be sorted in lexicographical order.

```
> python problem8.py sample_data_problem_8.txt
New York:
#nyc, 119
#HighClassHoodlums, 86
#NYC, 72
#Job, 40
#newyork, 30
San Francisco:
#SanFrancisco, 17
#sanfrancisco, 14
#Job, 13
#Jobs, 10
#sf, 9
```

The output of the sample is available at `http://vgc.poly.edu/projects/gx5003-fall2014/week3/lab/data/sample_output_problem_8.txt`.

Questions

Any questions should be sent to the teaching staff (Instructor Role and Teaching Assistant Role) through the NYU Classes system.

How to submit your assignment?

Your assignment should be submitted using the NYU Classes system. Create a zip file with your source code *problem1.py through problem8.py*. Name the zip file as NetID_assignment_3.zip, changing NetID to your NYU Net ID.

Grading

The grading is going to be done by a series of tests and manual inspection when required. Make sure your code runs on the sample datasets as specified to minimize the need for manual inspection of the code, which can be very subjective.