Scalable Data Analytics **Homework 1**
Spring 2020 **Feb.16, 2020**

Deadlines Homework 1 is due on Feb.16th 23:55pm. No late submission is accepted.

How to submit: Please submit a zip file to the *Assignment/Homework_1* folder in the iCollege. The zip file name should be 'Yourname-Pantherid.zip'. The zipped file should contain three separate ipython notebook files '1-generator.ipynb', '2-HOF.ipynb', and '3-generator-HOF.ipynb' for the first, second and third problems respectively.

Data Set: Citibike dataset posted in the iCollege.

---

1. (2 points) Python's Generators and Streaming.

   Compute the median age of the Citibike's subscribed customers. You are required to read data line by line and are not allowed to store the entire data set in memory. Indeed, you should not have any containers (e.g. list, dictionary, DataFrame, etc.) with more than 100 elements in memory. You should use yield when you want to iterate over a sequence, but don't want to store the entire sequence in memory as shown in the Codes/Lab3.

   What to submit:

   Turn in an ipython notebook with the plot of the histogram of customers age and print out a single number showing the median age of the subscribed customers.

2. (4 points) Python's Higher Order Functions

   This is how you can read the file and transform it to a list of lists.

   ```
   import pandas as pd
   df = pd.read_csv("citibike.csv")
   rows = df.values.tolist()
   ```

   (a).

   Determine the number trips that gender 1 made, and that gender 2 made. We can do this by just counting the number of occurrences of "1" and "2" in the gender column (2pt):

   ```
   <Read file>
   <YOUR HOF EXPRESSION>

   # After this, you should get something like
   # (37805, 7848)
   ```

(b).

Count the number of trips per birth_year using higher order functions (2pt):

```
<Read file>
<YOUR HOF EXPRESSION>

# After this, you should get something like
# {"1900.0": 22, "1901.0": 1, "1910.0": 2, "1922.0": 4, ... "1995.0": 256,
"1996.0": 124, "1997.0": 94, "1998.0": 59, "1999.0": 17}
```

Hint: math.isnan() is able to remove all the nan values.

What to submit:

Turn in an ipython notebook print out the results for problems (a) and (b).

3. (4 points) Extract the first ride of the day from a Citibike data stream. The first ride of the day is interpreted as the ride with the earliest starting time of a day. For the sample data, which is a week worth of citibike records, your program should only generate 7 items (one for each day).

   (a) Streaming Computation (2pt): you are asked to complete the task using steaming computation methods. You can only iterate the data set once using yield as shown in Codes/Lab3. You can store a container (e.g. list, dictionary, DataFrame,etc.) with maximum 7 elements in memory. The data set has been sorted by the starting time.

   (b) High Order Functions (2pt): you are asked to complete the same task using higher order functions map(), filter() and/or reduce() instead of streaming computation methods.

   Hints for (b):

```
<ANY FUNCTION TO BE USED IN YOUR HOF>

with open('citibike.csv','r') as fi:
    reader = csv.DictReader(fi)
    first_birth_years = <YOUR HOF EXPRESSION>

# After this, your first_birth_years should be
# [1978, 1992, 1982, 1969, 1971, 1989, 1963]
```

   What to submit:

   Turn in an ipython notebook print out the birth years of the first riders each day for problems (a) and (b).