Analytics in Action

# Python Proficiency Test

C. DANIEL GUETTA

Please complete the following exercise to test your Python proficiency. **You must complete this assignment alone,** but you are welcome to to use google/the internet freely. I will interview a random subset of applicants to ensure you wrote the answer yourself, and if it becomes clear the answer was not yours, disciplinary action will be taken.

Please go to the Citibke data website (https://s3.amazonaws.com/tripdata/index.html) and download the 201306-citibike-tripdata.zip file; this contains all trips carried out on the Citibike system in June 2013.

Unzip the file, and load the data using the following Python command. This command also selects the columns we will require for this exercise, renames them so as not to include spaces, and removes invalid rows.

```
import pandas as pd
import numpy as np
data = (pd.read_csv('201306-citibike-tripdata.csv')
           [['tripduration', 'starttime', 'start station name',
                           'end station name', 'bikeid', 'usertype']]
           .rename(columns = {'start station name':'start_station_name',
                              'end station name':'end_station_name'})
data = data[data.end_station_name.notnull()]
```

You will be asked you submit your answer to the questions below in a single Jupyter notebook, the first cell of which should contain the code above. You will be graded both on correctness and on the clarity of your code.

Please note that some of these questions are difficult – you do not need to answer them all correctly to get into the class.

**Author affiliation**
* *Columbia Business School*

## *Question 1*

Create a dataframe that - for each bike - tells us how many trips it was used for and the average duration of those trips.

## *Question 2*

The `usertype` column describes the user type for that trip. For "Subscribers", trips shorter than 45 minutes are free, and trips over 45 minutes incur a fee. For "Customers", trips shorter than 30 minutes are free, and trips over 30 minutes incur a fee. Create a dataframe that - for each station - lists the proportion of trips *started* at that station that were free.

## *Question 3*

For each station, we define the station's "drop off popularity" as the number of times a bike was dropped off at that station in our data. For each station, we define the station's "proxy popularity" by looking at all the trips *originating* from that station, and averaging the drop off popularity of the stations these trips were ended at. For example, suppose station X had three trips in our data; one trip ending at station A (drop off popularity 1500), one ending at station B (drop off popularity 1200), and one ending at station C (drop off popularity 1600), then the proxy popularity for station X is (1500 + 1200 + 1600)/3 = 1433.33. Create a dataframe that - for each station - finds its proxy popularity.

## *Question 4*

Fit a linear regression model to predict the number of bikes that will be picked up from a station in a given hour based on the hour of day, day of week, and the number of bikes that were picked up from that station in the previous hour. Fit the model on the whole dataset - do not use separate training/test sets.

A colleague of yours suggests that every night, you should use the linear regression model you just created to estimate the number of bikes that will be required at each station the next day, and use it to re-balance bikes across the system. What is the glaring flaw in this suggestion?

## *Question 5*

Consider the following line of code:

```
( data.assign(h = lambda x : pd.to_datetime(x.starttime).dt.hour)
      .groupby('h')
      .tripduration
      .agg([np.mean, np.std, len])
      .reset_index()
      .assign(stde = lambda x : x['std'] / np.sqrt(x['len'])) )
```

Explain what the author of this code was doing, and why they were doing it.