Project Big Data – Assignment 3

The deadline for this assignment is Sunday, June 23, at 23:59.

Part 1

The goal of this part of the assignment is to provide you with practice and experience in some basic data exploration and hypothesis testing with Python. You will work with data from the "HUE bedtime procrastination study". A cleaned version of the data is available on Canvas (hue_week_3.csv), as well as another file that contains data from the post-study questionnaire that participants filled out at the end of the study (hue_questionnaire.csv). This file contains the following information:

gender	1 = male, 2 = female
age	Numeric age value
chronotype	Single item (7-point scale), do you consider yourself more of a morning (1) or an evening person? (7)
bp_scale	Dutch version of the Bedtime Procrastination Scale (see: http://selfregulationlab.nl/wp-content/uploads/2012/04/J-Health-Psychol-2016-Kroese-853-62.pdf).
motivation	Questions pertaining to personality traits related to procrastination. Single item (7-point scale), how motivated were you to go to bed on time each night? (1 = not motivated, 7 = very motivated)
daytime_sleepiness	Dutch translation of the Epworth Sleepiness Scale (4-point scale from 0-3; 8 questions, values summed)
self_reported_effectiveness	Single item (7-point scale), do you feel more rested since the intervention

For this assignment (part 1), you will use Python to examine this post-questionnaire data in relation to the HUE data file, visualize trends and relationships, look for correlations between factors, test for significant differences between groups and build a regression model to predict bedtime delay. You are required to hand in your Python code to show that all transformations, visualizations and analyses have been done in Python. Your Python code will not be graded, however. You will submit your findings as a report, with numbering consistent with that described below. In order to perform the analyses, a number of transformations on the data still need to be done. To help you along, a Python template (template_part_1.py) will be made available with a recommended structure for your Python code.

The following steps must be implemented (in Python):
(20 points)

- Read the hue data file and the questionnaire data file into two separate pandas DataFrames.
- Create a new DataFrame that contains the following Series:

ID	Participant ID
group	Participant group (1 for experimental, 0 for control)
delay_nights	The number of nights a participant delayed their bedtime (range: 0-12)

delay_time	The mean time in seconds a participant delayed their bedtime (total delay in seconds, divided by the number of observations measured for each individual, rounded to nearest second).
sleep_time	The mean bedtime in seconds of a participant.

- Set the index of this new DataFrame to 'ID'. Note that there should only be a single row per participant ID.
- Fill this new DataFrame by transforming data from the DataFrame about participants' bedtimes (from the hue data file).
- Merge this new DataFrame with the post-questionnaire data and store the resulting DataFrame in a new variable. Perform this joining operation of the two DataFrames in such a way that the resulting DataFrame only contains IDs that were present in both datasets.
- Remove the rows that have NaN values in this merged DataFrame.

Perform the steps below and write in your report the following:

1. (5 points) Use the scipy.stats package to calculate correlations between the following sets of determinants:
   a. Bedtime procrastination scale (bp_scale, a personality trait) and mean time spent delaying bedtime. Use the "Pearson correlation tests" to calculate the correlation.
   b. Age and mean time spent delaying bedtime. Use the "Kendall rank correlation test" to calculate the correlation.
   c. Mean time spent delaying bedtime and daytime sleepiness. Use the "Pearson correlation test" to calculate the correlation.
2. (15 points) Use the scipy.stats package to determine whether there are significant differences between the experimental group and the control group in terms of:
   a. The number of nights participants delayed their bedtime
   b. The time participants spent in bed each night
   c. The mean time participants spent delaying their bedtime

   Use knowledge gained in the course 'Statistics' to determine which statistical test is appropriate: the t-test or the Wilcoxon rank-sum test. Explain your choice of test and discuss your findings.
3. (5 points) Formulate and concisely argue for a hypothesis about which factor or factors (max. three) you believe would best predict delay time. Write your hypothesis down. Note that you should theorize about why you think these factors might be good predictors before performing any analyses. Note that there is no single correct answer here. Your argument is most important.
4. (15 points) Use statsmodels.api to build a regression model that uses your three hypothesized determinants to predict delay_time (see page 1 of this document). Make sure that delay_time is not included in the list of predictors. Interpret and discuss your findings.
5. (15 points) Create least three distinct, meaningful, well-crafted visualizations that either provide insight into the data, or help support your conclusions. This means creating three different kinds of plots (not three boxplots, or three scatterplots for example). Interpret and discuss your findings.

Part 2

The goal of this part of the assignment is to provide you with practice in implementing MapReduce in Python. Using the map_reduce_hue.csv dataset, you will implement two simple MapReduce algorithms. Use the provided template (template_part_2.py). It is important to stick to this template.

1. (10 points) Write a MapReduce algorithm that counts and outputs the total number of times the fitness value is strictly higher than 50. The expected output is a single integer.
2. (15 points) Write a MapReduce algorithm that calculates the mean fitness per participant. Do not use any statistical packages to calculate the mean. The expected output is one line per participant, containing the participants ID and the mean of his or her fitness, separated by a tab ("\t"). The outputted lines do not have to be sorted.

Assume that there are multiple threads handling the mapping phase, but that there is only one thread in charge of reducing. This means that if you need to use global variables, you should do so only in the reducer. In addition, assume that the end of the input will be signaled by an empty row (delimiters only, no values). Finally, make sure that each function writes its output to the standard output via the print() function. See template_map_reduce.py for a template to get started. Submit your code for grading.