Week 5 - Assessed Exercises

Data Programming with Python

This week we learnt about Series and DataFrames. In particular, using indices, indexing and slicing, boolean indexing, and using simple functions on series and data frames.

Each question asks you to write a function with a specific set of input arguments. The *.py* template defines the function name and inputs for each question, **do not** change these. Be sure you test your functions before you submit your code to make sure that they are outputting the correct answer.

You may find it useful to test your functions on the `Diamonds` dataset from Week 1. Locate it on your computer and copy it into your current working directory. The *.py* file contains some suggested tests for each function. You don't need to include the output of your tests in your PDF.

1. Write a function that takes a `DataFrame` *df* and returns a subset of this `DataFrame`. The function inputs should be the `DataFrame` *df*, and two numerical arrays *rowinds* and *colinds*, which specify the rows and columns you wish to be includes in your new `DataFrame`.
2. This question is similar to Q1, but instead of using numerical indices we're going to specify a boolean condition for selecting the data for our subset. Your inputs should include a `DataFrame` *df*, a column of that `DataFrame` *col*, the label of another column *label* and two values *val1* and *val2*. The function should output the entries of the column labelled *label* for which the entries of the column *col* are greater than the number *val1* and less than *val2*.
3. We define a distance measure for the distance between observations i and j as

   $$\text{dist} = \left(\frac{\text{carat}_i - \text{carat}_j}{0.8}\right)^2 + \left(\frac{\text{table}_i - \text{table}_j}{57}\right)^2 .$$

   Write a function that takes a `DataFrame` *df* as its input and computes the distance between each of the observations in *df*. The output should be a $n \times n$ matrix, where n is the number of rows in *df*. The entry in the ith row and jth column of this matrix should be the distance between the ith and jth measurements (i.e. ith and jth row of *df*). You can assume that *df* has columns *carat* and *table* and `df.carat` and `df.table` will work inside your function.
4. The dissimilarity score is the sum of all the distances for a particular measurement, i.e. the sum of each row of the distance matrix. Write a function which takes a `DataFrame` *df* as an input and computes the dissimilarity score for each measurement and add this as an extra column called *Dissimilarity* to *df*. This extended `DataFrame` should be returned by the function. Note: You can call your function from Q3 inside the `exercise4` function.

All of your code should be written into the *.py* template. Save your filled *.py* file with the following name structure *SurnameFirstname_Week5.py* (where *Surname* and *Firstname* should be replaced with your name) and upload it to Brightspace. Additionally, you must upload a PDF of your code. Create a PDF from Canopy by selecting *File* → *Print*, and print to PDF.