**Week 4 Assignment: Data exploration and wrangling with Pandas**

This assignment is meant to help you to understand how to use [Pandas - Python Data Analysis Library](http://pandas.pydata.org/) to do some basic data analysis and data manipulations using **real data**. In this assignment, you are asked to analyze temperature data from Helsinki (in Southern Finland) and Rovaniemi (city in northern Finland) and to explore how their summer temperatures have differed in 2017.

## Data

For this assignment, we will use NOAA weather data (file attached in the Assignments area of LEO): **6153237444115dat.csv**

You can read the full description of the data from a 3505doc.txt (also attached) – the file contains the description for each attribute.

The first five rows of the data look like following:

USAF,WBAN,YR--MODAHRMN,DIR,SPD,GUS,CLG,SKC,L,M,H,VSB,MW,MW,MW,MW,AW,AW,AW,AW,W,TEMP,DEWP,SLP,ALT,STP,MAX,MIN,PCP01,PCP06,PCP24,PCPXX,SD

028450,99999,201705010000,174,10,14,\*\*\*,\*\*\*,\*,\*,\*,2.2,\*\*,\*\*,\*\*,\*\*,67,\*\*,\*\*,\*\*,8,31,31,1009.2,\*\*\*\*\*,984.1,\*\*\*,\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,35

028450,99999,201705010020,180,10,\*\*\*,4,\*\*\*,\*,\*,\*,2.9,\*\*,\*\*,\*\*,\*\*,10,\*\*,\*\*,\*\*,\*,30,30,\*\*\*\*\*\*,29.74,\*\*\*\*\*\*,\*\*\*,\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*

028450,99999,201705010050,190,10,\*\*\*,4,\*\*\*,\*,\*,\*,2.1,\*\*,\*\*,\*\*,\*\*,10,\*\*,\*\*,\*\*,\*,30,30,\*\*\*\*\*\*,29.74,\*\*\*\*\*\*,\*\*\*,\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*

028450,99999,201705010100,188,12,16,\*\*\*,\*\*\*,\*,\*,\*,3.2,\*\*,\*\*,\*\*,\*\*,77,\*\*,\*\*,\*\*,\*,31,30,1009.1,\*\*\*\*\*,984.0,\*\*\*,\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,\*\*\*\*\*,35

**NOTICE**: the data includes \* -characters that represent NoData values.

The most important attributes for this exercise are:

* **USAF** = the station ID number
  + 028450 : Rovaniemi
  + 029980 : Helsinki Kumpula
* **YR--MODAHRMN** = Year-Month-Day-Hour-Minute in Greenwich Mean Time (GMT)
* **TEMP** = Temperature in Fahrenheit
* **MAX** = Maximum temperature in Fahrenheit
* **MIN** = Minimum temperature in Fahrenheit

## Part 1 - Basic statistics of the data

In this problem your task is to download and explore the data from **6153237444115dat.csv** by reading the data into Pandas and conduct following tasks / answer to following questions:

* Read the data into a variable called data.
  + **Important**: When reading the data, it is important that you tell to Pandas that no-data values are specified with varying number of \* characters.
  + You can do this by specifying a following parameter in the read\_csv() -function (see appendix at the end of this file for additional instructions):
    - na\_values=['\*', '\*\*', '\*\*\*', '\*\*\*\*', '\*\*\*\*\*', '\*\*\*\*\*\*']
* How many rows are there in the data?

|  |
| --- |
| 11,694 rows |

* What are the column names?

|  |
| --- |
| ['USAF', 'WBAN', 'YR--MODAHRMN', 'DIR', 'SPD', 'GUS', 'CLG', 'SKC', 'L', 'M', 'H', 'VSB', 'MW', 'MW.1', 'MW.2', 'MW.3', 'AW', 'AW.1', 'AW.2','AW.3', 'W', 'TEMP', 'DEWP', 'SLP', 'ALT', 'STP', 'MAX', 'MIN', 'PCP01','PCP06', 'PCP24', 'PCPXX', 'SD'] |

* What are the datatypes of the columns?

|  |
| --- |
| * USAF int64 * WBAN int64 * YR--MODAHRMN int64 * DIR float64 * SPD float64 * GUS float64 * CLG float64 * SKC object * L float64 * M float64 * H float64 * VSB float64 * MW float64 * MW.1 float64 * MW.2 float64 * MW.3 float64 * AW float64 * AW.1 float64 * AW.2 float64 * AW.3 float64 * W float64 * TEMP float64 * DEWP float64 * SLP float64 * ALT float64 * STP float64 * MAX float64 * MIN float64 * PCP01 float64 * PCP06 float64 * PCP24 float64 * PCPXX float64 * SD float64 |

* What is the mean Fahrenheit temperature in the data? (TEMP column)

|  |
| --- |
| 52.25 |

* What is the standard deviation of the Maximum temperature? (MAX column)

|  |
| --- |
| 10.31 |

* How many unique stations exists in the data? (USAF column)

|  |
| --- |
| 2 |

You should write your code into a data\_exploration.py file and print the answers for the questions above inside the script.

* Save the script to your computer so you can upload to the LEO assignments folder
* Remember to comment your code! (add comments explaining what your code does)

## Part 2 - Data manipulation

The temperatures in our data are represented in Fahrenheit, hence, you need to convert the temperatures into Celsius.

* Select from the data columns USAF, YR--MODAHRMN, TEMP, MAX, MIN and assign them into a new variable called selected
* Remove all rows from selected that has NoData in column TEMP using dropna() -function
* Convert the Fahrenheit temperatures from TEMP into a new column Celsius using the conversion formula:
  + [](https://github.com/Geo-Python-2017/Exercise-5/blob/master/img/Fahrenheit_to_Celsius_formula.PNG)
* Round the values in Celsius to have 1 decimal place (**don't** create a new column --> update the current one)

You can add your codes into a data\_exploration.py file.

* Save the script to your computer so you can upload to the LEO assignments folder
* Remember to comment your code! (add comments explaining what your code does)

## Part 3 - Data selection

In this problem you should divide the data into separate subsets for different stations and save those DataFrames into disk.

* Divide the selection into two separate datasets:
  + Select all rows from selected DataFrame into variable called kumpula where the USAF code is 29980
  + Select all rows from selected DataFrame into variable called rovaniemi where the USAF code is 28450
* **Save kumpula DataFrame into Kumpula\_temps\_May\_Aug\_2017.csv file (CSV format)**
  + separate the columns with ,
  + use only 1 decimal in the floating point numbers
* **Save rovaniemi DataFrame into Rovaniemi\_temps\_May\_Aug\_2017.csv file (CSV format)**
  + separate the columns with ,
  + use only 1 decimal in the floating point numbers
* Save your csv files to your computer and upload to the LEO assignments folder

You can add your codes into a data\_exploration.py file.

* Save the script to your computer so you can upload to the LEO assignments folder
* Remember to comment your code! (add comments explaining what your code does)

## Part 4 - Data analysis

In this part the aim is to understand how different the summer temperatures has been in Helsinki Kumpula and Rovaniemi. Using the data from Part 3 in kumpula and rovaniemi DataFrames answer the following questions (all temperature output in Part 4 should be in Celsius with one decimal place):

**Part 4a**

* What was the median temperature in Celsius (one decimal place) in each of the following locations:
  + Helsinki Kumpula?

|  |
| --- |
| Median temperature in Kumpula 15.0 °C |

* + Rovaniemi?

|  |
| --- |
| Median temperature in Rovaniemi 11.7 °C |

**Part 4b**

Part 4a considers data from quite long period of time (May-Aug), hence the differences might not be so clear. Let's find out what were the mean temperatures in May and June in Kumpula and Rovaniemi.

* Select from rovaniemi and kumpula DataFrames such rows from the DataFrames where YR--MODAHRMN values are from May 2017 (see **Selecting Date Ranges** below for help) and assign them into variables rovaniemi\_may and kumpula\_may
* Do similar procedure for June and assign those values into variables rovaniemi\_june and kumpula\_june
* Using those new subsets print the mean, min and max temperatures for both places in May and June (include labels, not just numbers). You should use Python print commands within your code to print to output and then cut and paste the output below.
  + Kumpula Mean, Min, Max temperatures for May and June

|  |
| --- |
| Kumpula Stats For May: mean: 10.1 min: 2.2 max: 17.8  Kumpula Stats For June: mean: 14.2 min: 6.7 max: 21.1 |

* + Rovaniemi Mean, Min, Max temperatures for May and June

|  |
| --- |
| Rovaniemi Stats For May: mean: 3.5 min: -2.8 max: 12.2  Rovaniemi Stats For June: mean: 11.6 min: 1.1 max: 21.7 |

**Files to upload to LEO:**

* This Assignment Instructions document (with question/textboxes completed)
* Kumpula\_temps\_May\_Aug\_2017.csv
* Rovaniemi\_temps\_May\_Aug\_2017.csv
* Data\_exploration.py