COMP20270

OOP in Python

# Assignment 2

# **MyDataFrame**

Due 7th December.

## Objective

Pandas are one of the most important data structures in Python ([pandas.pydata.org](https://pandas.pydata.org/)). The objective of this assignment is to implement basic versions of some of the objects in Pandas.

**import pandas as pd**

**d = {'Sun Hours': [4.5,4.0,5.1,5],**

**'Max Temp': [19.6,19.1,19.6,20.0],**

**'Min Temp': [12.7,12.5,13.3,12.1],**

**'Rain (mm)': [82,109,65,76],**

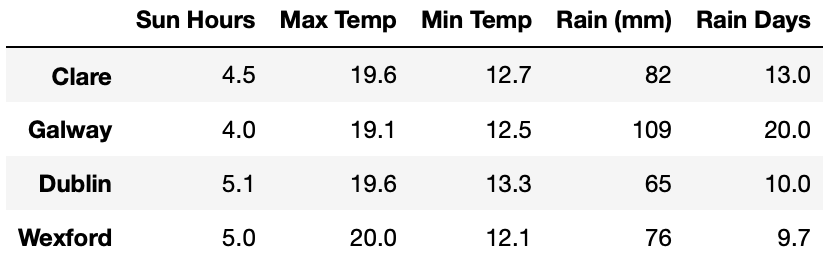
**'Rain Days': [13,20,10,9.7]}**

**Aug\_df = pd.DataFrame(d, index = ['Clare', 'Galway','Dublin',**

**'Wexford'])**

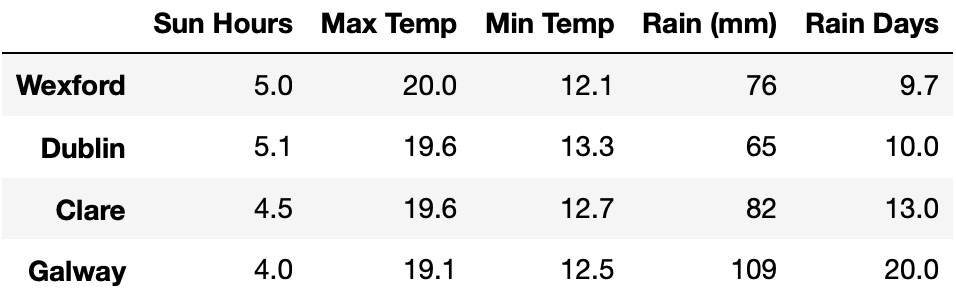
**Aug\_df**

**Out[68]:**



**Aug\_df.sort\_values('Rain Days')**

**Out[71]:**



**Aug\_df.max()**

**Out[75]:**

**Sun Hours 5.1**

**Max Temp 20.0**

**Min Temp 13.3**

**Rain (mm) 109.0**

**Rain Days 20.0**

**dtype: float64**

**Aug\_df.min()**

**Out[76]:**

**Sun Hours 4.0**

**Max Temp 19.1**

**Min Temp 12.1**

**Rain (mm) 65.0**

**Rain Days 9.7**

**dtype: float64**

## Requirements

### MySeries

1. Create a class called MySeries that will implement an indexed series stored as a dict, e.g.

**ms3 = MySeries([1,2,1], index = ['a','b','c'])**

**ms3.s\_dict**

**Out[58]:**

**{'a': 1, 'b': 2, 'c': 1}**

If no index is provided an index is generated automatically as follows:

**ms4 = MySeries([4,5,6])**

**ms4.s\_dict**

**Out[59]:**

**{0: 4, 1: 5, 2: 6}**

An instance of MySeries can also be created directly from a dict.

**d = {'b': 1, 'a': 0, 'c': 2}**

**s2 = MySeries(d)**

**s2.s\_dict**

**Out[97]:**

**{'b': 1, 'a': 0, 'c': 2}**

1. **MySeries** should have methods **min, max** and **mean** to return the relevant values.
2. **MySeries** should also have a print method, e.g.

**ms3 = MySeries([1,2,1], index = ['a','b','c'])**

**ms3.print()**

**Out[88]:**

**a 1**

**b 2**

**c 1**

You may find it useful to implement an **item\_at\_ind()** method:

**ms3.item\_at\_ind('c')**

**Out[89]:**

**1**

### MyDataFrame

### Create a class called MyDataFrame that implements a basic dataframe. The column should be implemented using your MySeries class.

1. The constructor should accept the data as a dictionary and an optional index as a list, e,g:

**d = {'Sun Hours': [4.5,4.0,5.1,5],**

**'Max Temp': [19.6,19.1,19.6,20.0],**

**'Min Temp': [12.7,12.5,13.3,12.1],**

**'Rain (mm)': [82,109,65,76],**

**'Rain Days': [13,20,10,9.7]}**

**df1 = MyDataFrame(d)**

**df2 = MyDataFrame(d, index = ['Clare', 'Galway','Dublin',**

**'Wexford'])**

1. There should be a **print** function with the following behaviour:

### df2.print()

**Out[99]:**

### Sun Hours Max Temp Min Temp Rain (mm) Rain Days

### Clare 4.5 19.6 12.7 82 13

### Galway 4.0 19.1 12.5 109 20

### Dublin 5.1 19.6 13.3 65 10

### Wexford 5 20.0 12.1 76 9.7

1. There should be a **sort\_values** function that will take a column name as input and sort the data-frame based on that column. This should sort the data-frame *in place* rather than return a sorted copy as happens with Pandas.

### df2.sort\_values('Rain (mm)')

### df2.print()

**Out[100]:**

### Sun Hours Max Temp Min Temp Rain (mm) Rain Days

### Dublin 5.1 19.6 13.3 65 10

### Wexford 5 20.0 12.1 76 9.7

### Clare 4.5 19.6 12.7 82 13

### Galway 4.0 19.1 12.5 109 20

1. You should also implement **mean, max** and **min** methods as follows:

**df2.mean()**

**Out[102]:**

**Sun Hours 4.65**

**Max Temp 19.58**

**Min Temp 12.65**

**Rain (mm) 83.00**

**Rain Days 13.18**

**df2.max()**

**Out[102]:**

**Sun Hours 5.10**

**Max Temp 20.00**

**Min Temp 13.30**

**Rain (mm) 109.00**

**Rain Days 20.00**

**df2.min()**

**Out[103]:**

**Sun Hours 4.00**

**Max Temp 19.10**

**Min Temp 12.10**

**Rain (mm) 65.00**

**Rain Days 9.70**

### 

### Other Requirements

1. The constructors for both classes should include some error checking to ensure that the data provided is consistent and can be constructed into a data-frame.
2. Here is another set of sample data showing how the methods should behave.

**films = {'Rank': [112,62,41,172,230,176],**

**'Release Year': [1973,1980,1960,2015,1976,1996],**

**'IMDB Rating': [8.3,8.4,8.5,8.1,8.1,8.1],**

**'Time (minutes)': [129,146,109,118,120,98],**

**'Main Genre': ['Comedy','Horror','Horror','Drama','Drama','Drama']}**

**f\_names = ['Sting','Shining', 'Psycho','Room','Rocky','Fargo']**

**films\_df = MyDataFrame(films, index = f\_names)**

**films\_df.print()**

**Out[104]:**

**Rank Release Year IMDB Rating Time (minutes) Main Genre**

**Sting 112 1973 8.3 129 Comedy**

**Shining 62 1980 8.4 146 Horror**

**Psycho 41 1960 8.5 109 Horror**

**Room 172 2015 8.1 118 Drama**

**Rocky 230 1976 8.1 120 Drama**

**Fargo 176 1996 8.1 98 Drama**

**films\_df.mean()**

**Out[104]:**

**Rank 132.17**

**Release Year 1983.33**

**IMDB Rating 8.25**

**Time (minutes) 120.00**

**films\_df.sort\_values('Release Year')**

**films\_df.print()**

**Out[105]:**

**Rank Release Year IMDB Rating Time (minutes) Main Genre**

**Psycho 41 1960 8.5 109 Horror**

**Sting 112 1973 8.3 129 Comedy**

**Rocky 230 1976 8.1 120 Drama**

**Shining 62 1980 8.4 146 Horror**

**Fargo 176 1996 8.1 98 Drama**

**Room 172 2015 8.1 118 Drama**

**Submission:** This is an individual (not group) project. Submission is through the Brightspace page. Your submission should comprise your notebook only. Clear all outputs in the notebook before saving for submission. You should use markdown cells in the notebook to explain any design decisions you have made.