Project 2: Weather Data

CS3753/CS5163: Data Science Summer 2023

Instructor: Dr. Mohammad Imran Chowdhury

Total: 100 Points

Due: 06/27/2023 11:59 PM

In this project, I invite you to do the following tasks using the python libraries such as **NumPy**, and **Matplotlib** covered in class.

Task 1 (10 points): Load the Weather Dataset provided to you as **``weather.zip"** file into the Jupyter Notebook. Note that this notebook requires Python 3.6 or higher. After extracting and loading the weather dataset zip file into the Jupyter Notebook, **Type ``ls weather"** to list all files under the unzipped weather folder. The output should be as follows: **(10 points)**

```
In [3]: ls weather

 Volume in drive C is Windows
 Volume Serial Number is 80EA-62F4

 Directory of C:\Users\imran\[CS 5163] Class Demo\Project 2\weather

12/09/2022  08:06 PM    <DIR>          .
12/09/2022  08:06 PM    <DIR>          ..
12/09/2022  08:06 PM            44,303 project2.ipynb
12/09/2022  08:06 PM            24,088 readme.txt
12/09/2022  08:06 PM         1,711,530 RSM00030710.dly
12/09/2022  08:06 PM         8,523,546 stations.txt
12/09/2022  08:06 PM         4,165,020 USW00014922.dly
12/09/2022  08:06 PM         3,692,790 USW00022536.dly
12/09/2022  08:06 PM         3,502,710 USW00023188.dly
12/09/2022  08:06 PM               576 weather.README
               8 File(s)     21,664,563 bytes
               2 Dir(s)  63,773,245,440 bytes free
```

Task 2 (30 points): This task has three (03) steps. These are as follows:

(a) First, find and plot the mean TMAX in Minneapolis, a cool place, for the year 1940 through 2014. **(20 points)**

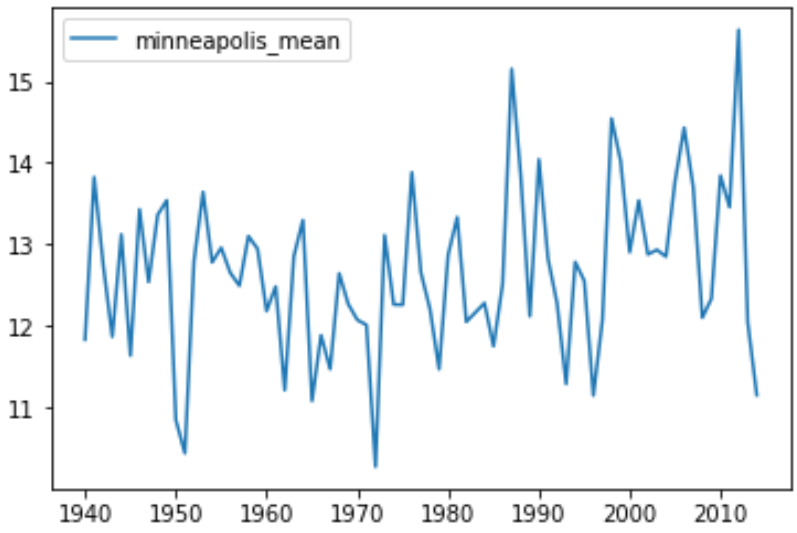
Here, you need to work with the Minneapolis temperature data file 'USW00014922.dly' to find the TMAX series in Minneapolis first. This requires you to call the method **getobs(filename, obs)** as shown in class, where the **filename** should be 'USW00014922.dly' and **obs** should be 'TMAX'. Note that **getobs(filename, obs)** method further requires to call the method **parsefile(filename)** and **unroll()** as shown in class. Next, you need to compute the Mean of the TMAX series for the year 1940 through 2014. This requires you to call the **selectyear(TMAX_series, year)** and **np.mean()** method on the TMAX series as shown in class. Note that **np.mean()** would not work properly unless you call the **fillnans(TMAX_series)** method on the TMAX series. You can also find the python implementation of all these methods in the given **project2.ipynb file** of the project folder.

The Mean of the TMAX series in Minneapolis for the year 1940 through 2014 should be as follows: **(15 points)**

```
minneapolis_mean
```

```
array([11.83917808, 13.82547945, 12.75561644, 11.87452055, 13.12520548,
       11.6430137 , 13.43260274, 12.54328767, 13.36410959, 13.54219178,
       10.8569863 , 10.44876712, 12.80109589, 13.64356164, 12.78493151,
       12.96191781, 12.65589041, 12.49863014, 13.10438356, 12.94986301,
       12.18849315, 12.48767123, 11.22136986, 12.85616438, 13.30027397,
       11.0890411 , 11.89205479, 11.47780822, 12.64821918, 12.27178082,
       12.07945205, 12.01835616, 10.28465753, 13.11643836, 12.26849315,
       12.26465753, 13.88438356, 12.65945205, 12.21123288, 11.47726027,
       12.87780822, 13.33589041, 12.05726027, 12.1709589 , 12.2860274 ,
       11.75835616, 12.51917808, 15.14986301, 13.84164384, 12.12493151,
       14.04356164, 12.83013699, 12.28273973, 11.29589041, 12.78767123,
       12.55753425, 11.15534247, 12.08328767, 14.54383562, 14.02438356,
       12.90739726, 13.54109589, 12.88109589, 12.93671233, 12.8539726 ,
       13.77972603, 14.43068493, 13.70465753, 12.10739726, 12.33972603,
       13.84465753, 13.45863014, 15.6290411 , 12.05479452, 11.15972603])
```

Also, when you plot the above Mean data using the python Matplotlib library, then the output should be as follows: **(5 points)**



(b) Second, find the year with the highest mean TMAX in Minneapolis. **(5 points)**

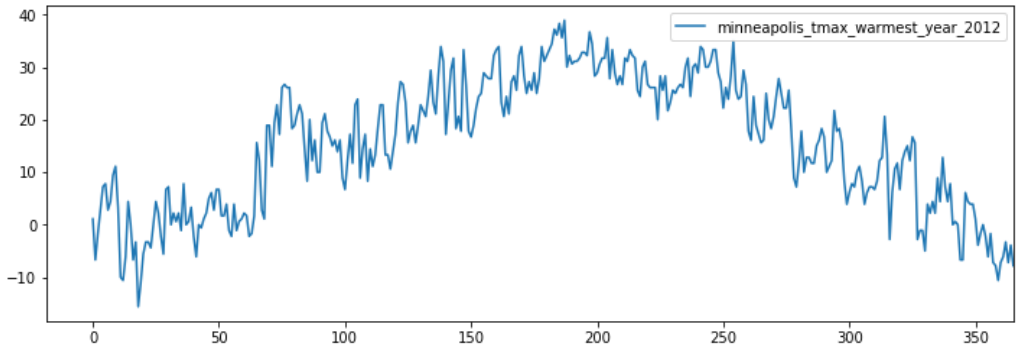
To get the highest mean TMAX year in Minneapolis, you can call the NumPy **np.argmax()** method on the above Minneapolis Mean TMAX series data. The output should be as follows:

```
minneapolis_warmest
```

```
2012
```

(c) Third, plot the temperature data for that highest mean TMAX year. **(5 points)**

This requires you to call the **selectyear(TMAX_series, highest_mean_year)** method as shown in class. Then after call the **plot()** method. The output should be as follows:



Task 3 (30 points): This task also has three (03) steps. These are as follows:

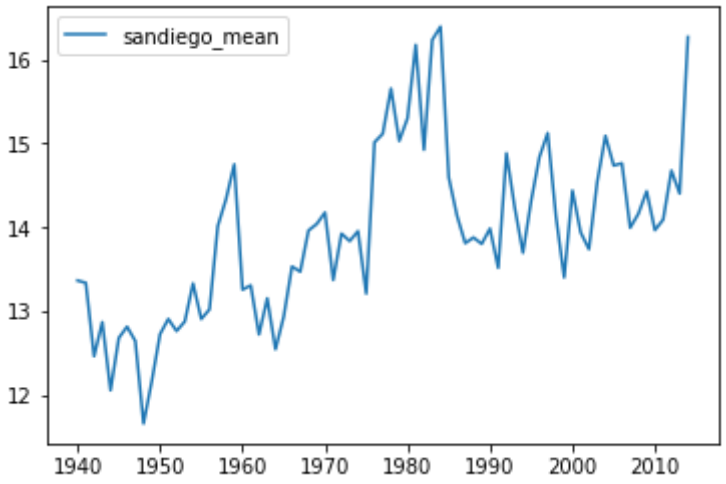
(a) First, find and plot the mean TMIN in San Diego, a warm place, for the year 1940 through 2014. **(20 points)**

Here, you need to work with the San Diego temperature data file 'USW00023188.dly' to find the TMIN series in San Diego first. This requires you to call the method **getobs(filename, obs)** as shown in class, where the **filename** should be 'USW00023188.dly' and **obs** should be 'TMIN'. Note that **getobs(filename, obs)** method further requires to call the method **parsefile(filename)** and **unroll()** as shown in class. Next, you need to compute the Mean of the TMIN series for the year 1940 through 2014. This requires you to call the **selectyear(TMIN _series, year)** and **np.mean()** method on the TMIN series as shown in class. Note that **np.mean()** would not work properly unless you call the **fillnans(TMIN_series)** method on the TMIN series. You can also find the python implementation of all these methods in the given **project2.ipynb file** of the project folder. The Mean of the TMIN series in San Diego for the year 1940 through 2014 should be as follows: **(15 points)**

```
sandiego_mean
```

```
array([13.36575342, 13.33753425, 12.46356164, 12.86931507, 12.05726027,
       12.68109589, 12.81534247, 12.64493151, 11.66082192, 12.16657534,
       12.72684932, 12.90739726, 12.76410959, 12.87671233, 13.33068493,
       12.90712329, 13.02246575, 14.01452055, 14.33506849, 14.75068493,
       13.25671233, 13.30767123, 12.72273973, 13.15342466, 12.54630137,
       12.93726027, 13.53287671, 13.4709589 , 13.96246575, 14.03945205,
       14.17616438, 13.37232877, 13.92493151, 13.83589041, 13.95452055,
       13.20986301, 15.01342466, 15.11479452, 15.6539726 , 15.02931507,
       15.29945205, 16.17452055, 14.92410959, 16.23315068, 16.39178082,
       14.59561644, 14.14      , 13.8109589 , 13.87808219, 13.80273973,
       13.98465753, 13.5169863 , 14.8830137 , 14.23671233, 13.69643836,
       14.31835616, 14.83890411, 15.1230137 , 14.13123288, 13.40273973,
       14.43808219, 13.93890411, 13.73917808, 14.53835616, 15.09013699,
       14.73643836, 14.76273973, 13.99506849, 14.16246575, 14.42958904,
       13.96630137, 14.09150685, 14.67616438, 14.40273973, 16.26794521])
```

Also, when you plot the above Mean data using the python Matplotlib library, then the output should be as follows: **(5 points)**



(b) Second, find the year with the lowest mean TMIN in San Diego. **(5 points)**

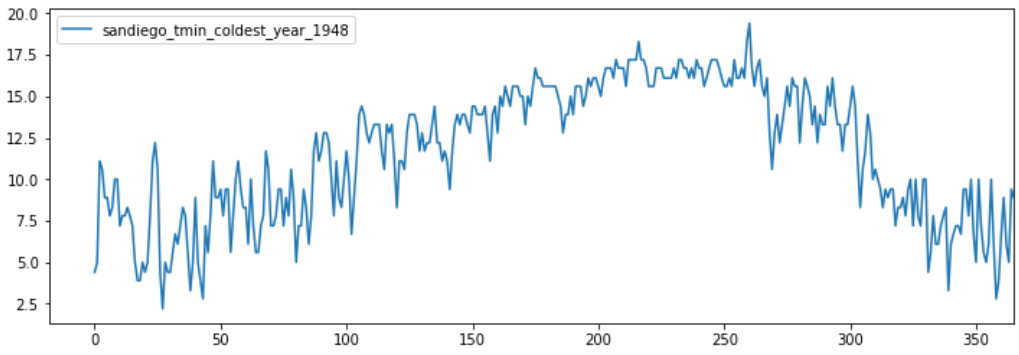
To get the lowest mean TMIN year in San Diego, you can call the NumPy **np.argmin()** method on the above San Diego Mean TMIN series data. The output should be as follows:

```
sandiego_coldest

1948
```

(c) Third, plot the temperature data for that lowest mean TMIN year. **(5 points)**

This requires you to call the **selectyear(TMIN_series, lowest_mean_year)** method as shown in class. Then after call the **plot()** method. The output should be as follows:

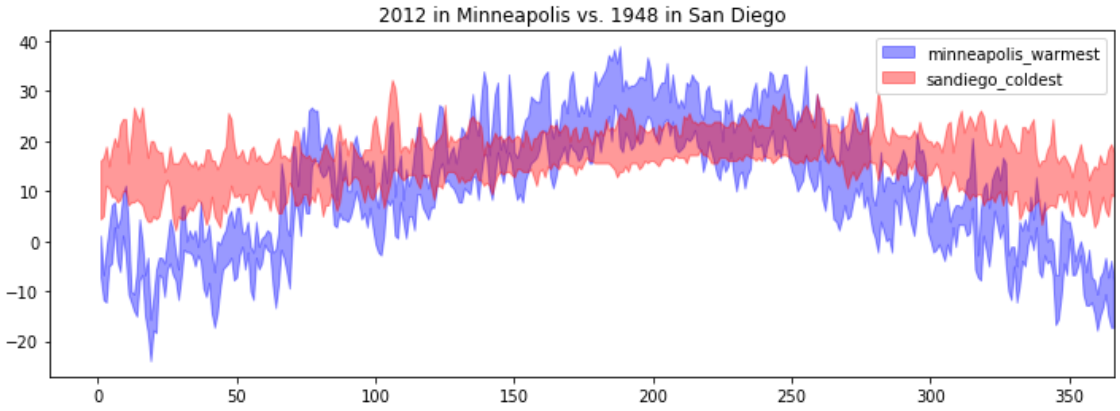


Task 4 (30 points): In the same plot, show TMIN and TMAX series for those years and for those two cities i.e., warmest year for Minneapolis and coldest year for San Diego.

This requires you to call the following methods:

fill_between(); selectyear(); getobs(); fillnans();

Where, **getobs(); fillnans(); selectyear();** methods will be needed to get the minneapolis_tmax, minneapolis_tmin, sandiego_tmin, sandiego_tmax series data and **fill_between();** method from the python Matplotlib library will be needed for plotting the TMAX and TMIN series data. After plotting the TMAX and TMIN series data both for the warmest year for Minneapolis and the coldest year for San Diego, the output should be as follows: **(30 points)**



The submission grading rubric is as follows (points out of 100 total):

Project element	**Points**
Task 1	10
Task 2	30
Task 3	30
Task 4	30

Submission Instructions: Create a compressed file (.zip or .tar.gz files are accepted) with your all source files such as .ipynb files and data files. Generally speaking to complete Task1 through Task4, you just need one .ipynb file. But it's better to submit everything as a compressed file. Submit the compressed file to Blackboard.

Late submission policy: As described in the syllabus, any late submission will the penalized with 10% off after each 24 hours late. For example, an assignment worth 100 points turned in 2 days late will receive a 20 point penalty. Assignments turned in 5 or more days after the due date will receive a grade of 0.