Introduction

In this assignment, I have written a code which takes a csv file path as an input and convert it to a set of sets containing the components of the input file. Additionally , there is another function which calculates the mean, max, min, and standard deviation of the numeric data of the parsed data, after converting the data to NumPy array for easier handling of data.

Assessment

1- Output min, max, mean, and standard deviation of original CSV files

In this section, statistical data are calculated using the stats() function for each csv file in the whether-data file. First, I converted the parsed data from my previous function to a NumPy array, and then used np.vectorize() to convert all the numeric data (by column i.e. axis=0) to vectors in order to be able to perform mathematical procedures. I have skipped the date and time in my calculations because mathematically it is not feasible in my opinion to perform statistics on them as they are discrete data and not continuous.

I have used np.mean to calculate the mean, np.min to calculate the minimum, np.max to calculate the maximum, and np.std for standard deviation calculation as shown in the figure below:

```
def stats(parsed_data):
    arr = np.array(parsed_data)
    vector = np.vectorize(np.float)

    arr_mean = np.mean(vector(arr[0:, 1:][1:]), axis=0)
    arr_min = np.min(vector(arr[0:, 1:][1:]), axis=0)
    arr_max = np.max(vector(arr[0:, 1:][1:]), axis=0)
    arr_std = np.std(vector(arr[0:, 1:][1:]), axis=0)

    print(f"Header:                    {arr[0, 1:]}\n"
          f"Mean:                      {arr_mean}\n"
          f"Minimum:                   {arr_min}\n"
          f"Maximum:                   {arr_max}\n"
          f"Standard Deviation:        {arr_std}\n")

    return arr_mean, arr_std, arr_max, arr_min
```

The results for each file are shown in the figures below:

a- barometer-1617.csv

```
Header:                         ['Baro']
Mean:                           [1009.99887324]
Minimum:                        [979.6]
Maximum:                        [1035.6]
Standard Deviation:             [9.8557511]
```

b- indoor-temperature-1617.csv

```
Header:                         ['Humidity' 'Temperature' 'Temperature_range (low)'
 'Temperature_range (high)']
Mean:                           [48.51977401 21.82788489 20.5559322  23.53361582]
Minimum:                        [37.   18.04 14.9  19.7 ]
Maximum:                        [59.   29.21 28.2  31.1 ]
Standard Deviation:             [5.1815518  2.05539796 2.40172522 1.69906087]
```

c- outside-temperature-1617.csv

```
Header:                         ['Temperature' 'Temperature_range (low)' 'Temperature_range (high)']
Mean:                           [11.13887676  7.8656338  15.52422535]
Minimum:                        [-1.81 -4.1   1.5 ]
Maximum:                        [26.38 18.7  38.5 ]
Standard Deviation:             [5.34749391 4.87205331 7.02453079]
```

d- rainfall-1617.csv

```
Header:                         ['mm']
Mean:                           [1.54872521]
Minimum:                        [0.]
Maximum:                        [23.2]
Standard Deviation:             [3.31988682]
```

2- Modifying a CSV file and outputting the summary statistics and compare with the original one

For the next parst of the assessment, I have modified the rainfall-1617.csv file and input two values which are up to 66x the value of the mean so we can refer to them as clear outliers as seen below:

DateTime	mm
10/9/2016 0:00	0
10/10/2016 0:00	50
10/11/2016 0:00	100
10/12/2016 0:00	0
10/13/2016 0:00	0
10/14/2016 0:00	1.1
10/15/2016 0:00	2.1
10/16/2016 0:00	8.4
10/17/2016 0:00	1.1

Below we can see the comparison of the original statistics (left) and modified data statistics (right):

Header:	['mm']	Header:	['mm']
Mean:	[1.54872521]	Mean:	[1.97393768]
Minimum:	[0.]	Minimum:	[0.]
Maximum:	[23.2]	Maximum:	[100.]
Standard Deviation:	[3.31988682]	Standard Deviation:	[6.70332466]

We can see that maximum value has changed along with the mean and standard deviation. The standard deviation is now almost double of the original unmodified values. To identify the outliers, we can see that the mean is still close to the original value.

Hence, we can iterate through all the values and compare them to the mean value and have a certain error percentage and if that percentage is high enough according to the user's specifications, it can be replaced with the mean value.