

# EE2703 Assignment 2 Report

**Adithya A EE24B081**

09.09.2025

## AIM OF THE EXPERIMENT

Using the dataset given in the file (GlobalLandTemperaturesByMajorCity.csv) and the helper functions given in the initial python file 'city\_temperature\_function.py', five functions need to be implemented. These functions have to pass all the test cases, including those which do not possess proper data.

### Function 1- Temperature Extremes

Read the csv file using the DictReader() function. Keep two variables called max\_temp and max\_month. When iterating through each of the rows of the csv file, if the temperature column is not empty, compare with max\_temp. If the temperature is greater than max\_temp, update max\_temp with the current temperature and max\_month with the current year\_month.

Similarly, for min\_temp, keep two variables as min\_temp and min\_month, iterate and update if the current temperature is lesser than min\_temp. Return the following variables in the form of a dictionary, as given in 'city\_temperature\_function.py'.

### Function 2 - Average Seasonal Temperatures

Use the get\_city\_temperatures function to get all the valid year-months and their corresponding temperatures as temp\_data. Make four sets named spring, summer, autumn and winter with the number of the corresponding months, for example, spring = ('03', '04', '05'). Also, make two variables called temp\_sum and count. Iterate through temp\_data. If the month lies in the given season, add the temperature to temp\_sum and increment the count. The average seasonal temperature will be temp\_sum/count. Return the average in the form of a dictionary, as given in 'city\_temperature\_function.py'.

### Function 3 - Decade Comparison

If the decade1 and decade2 do not divide by 10, raise ValueError as they are not valid decades. Use the get\_city\_temperatures function to get all the valid year-months and their corresponding temperatures as temp\_data. Make four variables tempd1, countd1, tempd2, countd2. Iterate through each key and value. If the year is in the range of the decade given in decade1, add the temperature to tempd1 and increment countd1. Similarly, do the same for decade2. Find the difference between the two average temperatures, to find if it is warming, cooling or stable. Return the variables in the form of a dictionary, as given in 'city\_temperature\_function.py'.

### Function 4- Similar Cities

Use the get\_available\_cities to get the set of available cities for comparison. Use the get\_average\_temp function, which I wrote in the .py file to get temp\_data, the dictionary of all average temperatures for the cities. Iterate through the cities set and find the average temperature and the country of the city. If the absolute value of the difference of average temperatures is lesser than the threshold value, add the city, country, average temperature, and the absolute difference to a list. Return the list in the form of a dictionary, as given in 'city\_temperature\_function.py'

### Function 5- Temperature Trends

Use the get\_city\_temperatures function to get temp\_data for the test city. Now iterate through the rows to get the temperatures for each year. The average temperatures equal the sum of the temperatures of each year divided by the number of temperatures recorded. Find and add them to a dictionary. Now find the moving averages, where you take groups of years the size of a variable 'window' and find the average temperature for it. Use the annual\_means dictionary to iterate through the average temperatures. If the slope increases, it is part of the warming period, and if the slope is negative, it is part of the cooling period. The overall\_slope is the difference of the last average temperature and the first average temperature divided by the year difference in between. Return the results in the form of a dictionary, as given in 'city\_temperature\_function.py'

## Data Handling

The `get_city_temperatures` function and the other functions take temperatures which are valid and can be represented in the form of a float. The other values are discarded. This may cause issues in the earlier years, but won't cause much problems as the data is close to complete in the more recent years.

## Plots for The Assignment

