MNXB11 Group E report

Ossian Malmborg

November 2024

Introduction

The goal of this project was to analyze Swedish weather data to find several different trends in the weather. This was done using BASH, C++ and ROOT for the purpose of gaining experience in coding as a group using GitHub. The data that was used came from the SMHI data provided by this course and also from the SMHI website itself.

Code

Firstly the code regarding the data on Gothenburg will be discussed, secondly the code that regards Kiruna and Jonkoping.

The data for Gotheburg was sourced from the website of SMHI and not the course provided data. First, the data was cleaned using the code in data.cxx which uses the goteburg.csv document as input and outputs the cleaned data called filtered_gote_temp.csv. One thing of note is that the date has 3 columns, which is because the source data also has 3 columns for date. From here the data is cleaned further to focus on the month of July in particular, this is done using data_july.cxx which uses filtered_gote_temp.csv as input and outputs filtered_july_temp.csv. There are 3 different analysis and figures that use this data, these are julygraph.cxx, scattorplot.cxx, and tempgraph.cxx. julygraph.cxx produces the graph july_temperature.png, scattorplot.cxx produces temp_scatterplot.png, and tempgraph.cxx produces temperature_histogram.png.

The data for Kiruna and Jonkoping was sourced from this course and its version of the SMHI data. The raw data KirunaAirport.csv and Jonkoping-Axamos_Airport.csv are turned into .root files with the code in convertCSVtoROOT.C. It should be noted that both .csv files are already cleaned before coming into the repository, more on this in the ”Problems” section. plotAverageTemperatureByYearWithFit.C takes the data for Kiruna and Jonkoping and outputs the graph averageTemperatureWithSingleFitByYear.png. Since the date was just one large number string, one could obtain the year by simply dividing by 10,000. It should be noted that the method for obtaining the year of the date is fairly bad, it just so happened to work in this case. plotSimpleTemperatureDifference.C was used to produce the graph for simpleTemperatureDifference.png. There happens to be a dead line at the end of the graph, which seems to be due to a reading error.

Analysis

For the analysis there were a total of 5 questions to be answered, these can be split into groups of 3 and 2 questions that are connected. The questions relating to the Gothenburg data were "how did the temperature in Gothenburg change between 2010 and 2020?", "what are the most common temperatures of that decade in Gothenburg?", and "how did the temperature change in that decade in the month of July for Gothenburg?". The first question is represented by the graph of julygraph.cxx (1). In this case the span of 10 years is not enough to come to any conclusion about a trend or a pattern. The second question is represented by the scatterplot of scattorplot.cxx (2). Yet again the span of the years makes it hard to come to any conclusions, but the temperature seems to have a trend towards rising. The high peaks remain the same but the lower peaks seem to have gone up a bit. The third question is represented by the histogram temperature_histogram.png (3). Here we can see that there are actually two peaks, meaning there are two common temperature ranges. These are however very close to each other, meaning the range of common temperatures is still relatively close to being Gaussian.

The questions for the data of Kiruna and Jonkoping were "what is the average yearly temperature in Kiruna and Jonkoping?" and "what is the difference of these temperatures?". The first question is answered by the graph averageTemperatureWithSingleFitByYear.png (4). This graph clearly shows that Jonkoping is consistently warmer than Kiruna. It would also seem that both of the cities are getting warmer. For the second question the graph simpleTemperatureDifference.png (5) was created. This one shows the difference between the two as a single line. The red line shows that the two cities seem to be getting gradually closer to each other in yearly average temperature.

Results

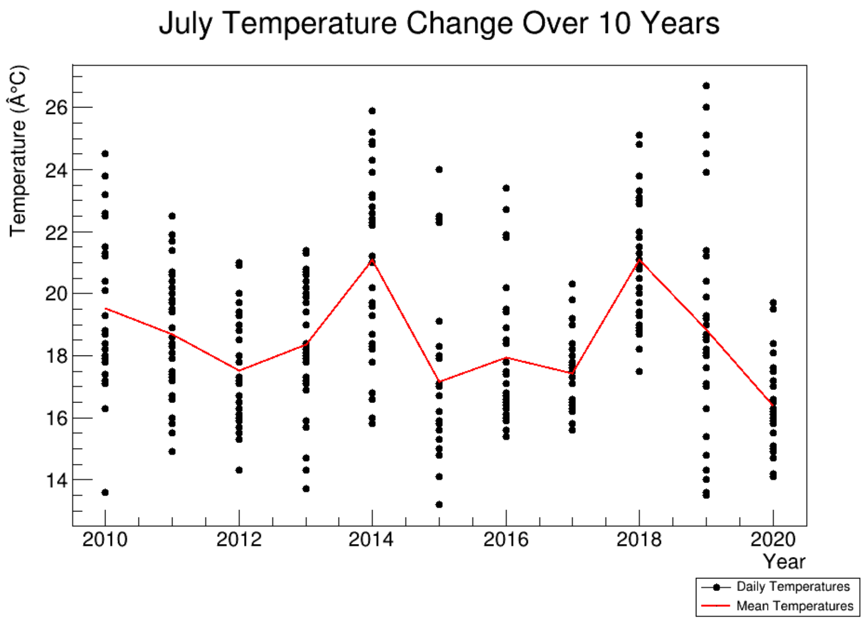

Figure 1: july_temperature.png

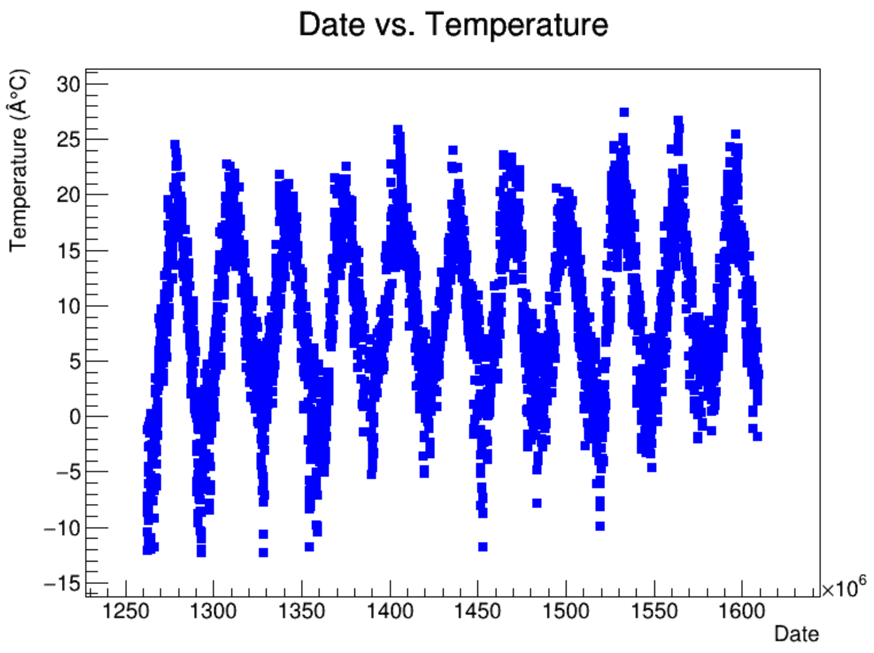

Figure 2: temp_scatterplot.png

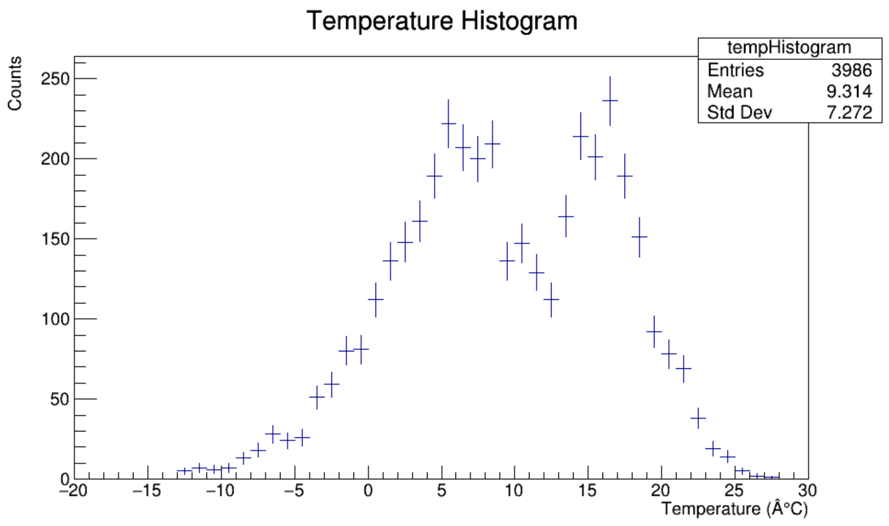

Figure 3: temperature_histogram.png

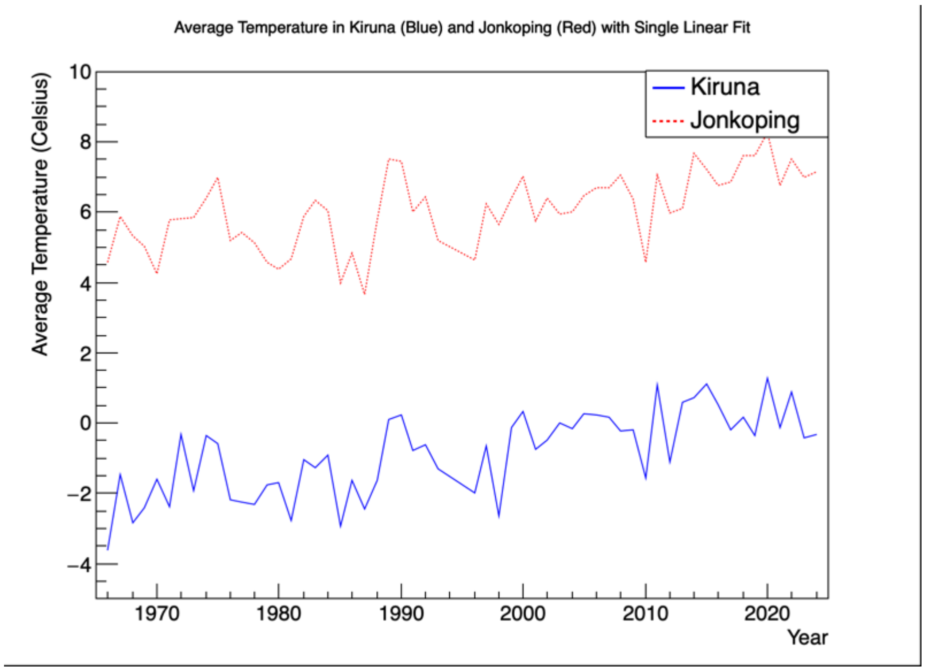

Figure 4: averageTemperatureWithSingleFitByYear.png

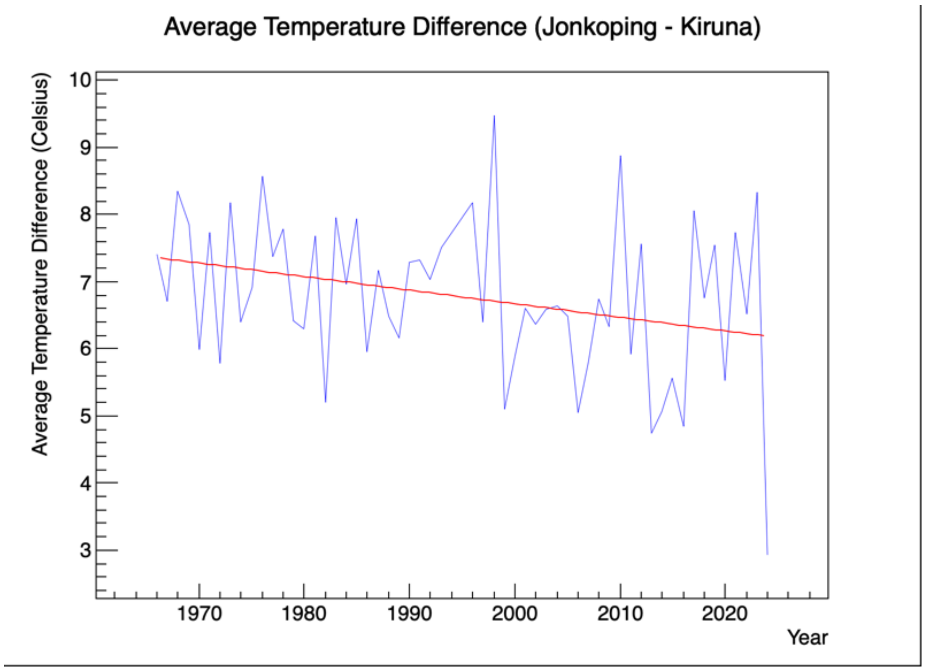

Figure 5: simpleTemperatureDifference.png

Problems

The project was mostly handled by two students since one fell sick and another disappeared. This meant the workflow was severely affected. The one who disappeared was in charge of data cleaning and the one who was sick was originally in charge of doing some analysis of the data. Due to it being exam week the project was done pretty late, meaning it had to be rushed to a certain extent. This resulted in the two remaining people having to do their own data cleaning, analysis and figures, since there wasn't much time to collaborate or synchronize their work. Making the once collaborative task into what amounts to two separate tasks. All this combined to magnify the amount of problems that would occur in the cleaning, coding and analysis. For the data for Gothenburg, getting the code to read correctly was difficult along with removing the other columns. Another difficulty with that data set was changing to the correct format for date and temperature, and especially the date. When it comes to the raw data for Kiruna and Jonkoping, it was cleaned manually instead of a with code since there wasn't enough time to figure out the method to extract the data. That is why there is no code for the cleaning of Kiruna and Jonkoping. This problem continued even with the cleaned data when the year was to be extracted, leading to the method of dividing the date by 10,000. This suggestion was one that was made by AI. AI was used to a certain extent when certain tasks became too hard to solve in remaining time. This means some of the code is bit out of the ordinary compared to the other code, since this was the only way to actually complete the task.