MNXB 11

Final Project Report

Authors
Marcus O'Riordan
German Luzin
Koshin Puri
Ana Rita Ferraz Rodrigues
Daniel Stålhammar

Date
October 30,2025

1 Introduction

This project aimed to look at trends in the weather throughout Sweden. That was achieved by using data provided by the Swedish Meteorological and Hydrological Institute (SMHI). This data consisted of the temperature taken at different times daily for several decades, or even centuries. These measurements were taken from stations across the country. With all of this available data, a large variety of relationships were possible to analyse.

As a group, 4 different analyses were selected for visualization. These were: to investigate the warmest and coldest days of the year, the yearly maximum average temperature in Söderarm, the difference in temperature on Christmas between Lund and Luleå, as well as finding the number of heatwaves and coldwaves in Lund and Uppsala.

In order to complete this as effectively as possible, a repository was created with a release manager, and then the other members of the groups created their separate forks. This method was chosen as it was most familiar to the group members. Sharing this repository allowed parts of the code to be shared, avoiding duplication.

The results of the four goals are presented individually, with each code having a method, result, and discussion section.

1.1 Corrupt Data

It is always a risk that data is corrupted due to lack of information. An example could be temperature for some date has not been noted or the date for some temperatures is missing or something other. That can cause trouble in the analysis and at worst affect the results, so they become very different than they had been if the data were not corrupted.
To avoid that happen is it important to control the data which should be used and secure it not is corrupted and that is what the file **corrupt-data** did.
When the data are processed, the data file should consist of lines. Every line should have year, date and temperature and the values are separated by the sign ,. The file **corrupt-data** control the lines. If the line not is empty the file splits the line at the , signs and puts all elements in an empty list. After that it control the length of the list. If the data on the line not is corrupt, the list should now have a length of three. If the length is three the file put the control list as an element in another list (endlist) and keeping the process with next line. When every line is investigated the file spit out the endlist which now should be free from corrupt data.

2 Detection of hottest and coldest days of the year

This part of the project describes the historical temperature data analysis with the goal of finding the distribution of hottest annd coldest days over the years as well as the most probable hottest and coldest days of the year. The analysis combined C++ programming, ROOT histogram plotting tools, parsing preprocessed CSV files. Analysis was focused on the data from Lund.

2.1 Method

Each CSV file was read line by line using `std::ifstream` and `std::stringstream`. A data structure `DataRow` was used to represent rows, and the data was stored in a vector of that structure. The function `calc_average_temp()` read the data stored in the vector and calculated the average temperature for each date present in data set, this was a function written by Marcus. Then the function `find_hottest_coldest_by_year()` returns the vector of structure `YearExtremes` that contains integers coldest and hottest day of the year.After that two histograms are created and filled with corresponding data, the Gaussian fits are done by using `FitGaussian()` function. This function draws histograms and prints out mean and sigma values of the fit, it also has a special feature that first rotates the domain of the histogram so that it starts with day 100, this allows successful fitting of the coldest days data, the fit values are then "rotated" back and the gaussian is built based on these values.

2.2 Results

The data analysis provided the following results. The average hottest day of the year is 196'th day which is the 15'th of July with a sigma value of 30 days and the coldest day of the year is 22'nd day which is the 22nd of January. The Figure 1 below represents the histogram of coldest and hottest days of the years as well as fitted gaussian curve.

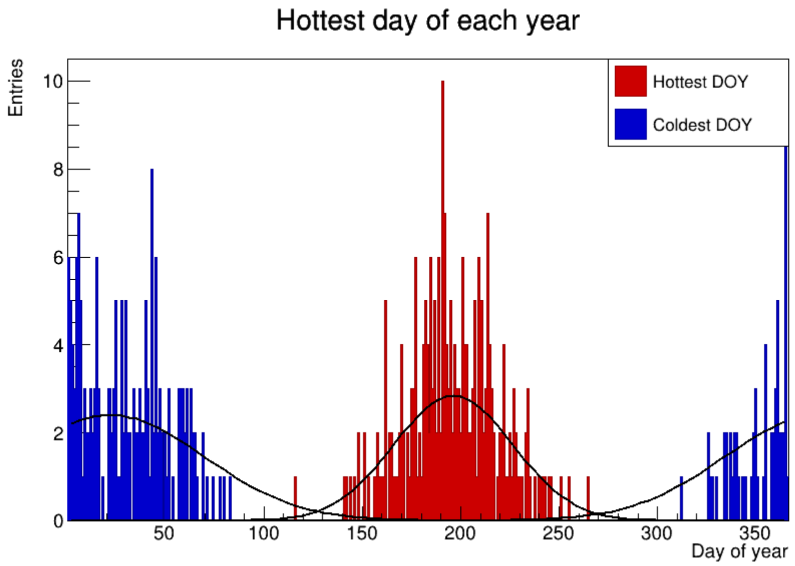

Figure 1: The histogram shows how often a given day of the year was the warmest or coldest. The lines on the histogram show Gaussian fits of the distributions.

See the obtained Gaussian fit values summarized in the Table 1 below.

	Mean	**Sigma**
Hottest day	196.283	30.4293
Coldest day	22.0019	47.7433

Table 1: Gaussian fit parameters for DOY distributions of temperature extremes.

2.3 Discussion

The hottest day is the middle of the summer, around 15'th of July while the coldest day occurs in late January, 22'nd of January. The spread of hottest and coldest days distributions is quite large, 30 and 47 days respectively. This is wide spead suggesting that the hottest and coldest day may occure anywhere in summer and winter in Lund. The wider spread for winter might occur for the following reason: The coldest day was calculated for eah year, not for each winter, that means that for a winter that was colder than its neighbouring ones, the dates from that one winter were counted twice. The analysis could be improved by taking that into account. Using C++ and root proved to be efficient, useful tools for temperature analysis.

3 Analysis of Yearly Maximum Average Temperature in Söderarm

The project is centred around using coding to find trends in Swedish weather data. I decided to focus on Söderarm, a lighthouse on the east coast of Sweden, roughly an hour drive in the North-East direction from Stockholm. My analysis is focused on finding the hottest average day in the year and comparing that value yearly in one location. The day has to be averaged as some dates have multiple temperature recordings. The goal of this analysis is to find if there is an overall warming trend to demonstrate local effects of climate change.

3.1 Method

The data was provided from a preprocessed CSV file. I then used code to read the data file due to a function created by Ana and then found the average temperature for each day using a function created by Marcus. Then a function I created finds the highest value, therefore the hottest temperature for each year and records it in a list of values that go chronologically listing the year and then the hottest average temperature in a day that year. Then I use another function that plots the data into a graph with a trend line and calculates the gradient to determine if there is an overall trend or not.

3.2 Results

The data analysis ranged from values from 1941 to 2023. The maximum average temperature on a given day in a year ranged from around 16 to 24 °C. The coldest temperature recorded was in 1962 being 16.4 °C compared to the hottest being 2018 with 24.1 °C. The average trend demonstrated a gradual increase in temperature as the gradient of the linear trendline was 0.0273922 in °C/year. This means over a century the hottest average temperature increased by 2.7 °C. The following figure shows the hottest average temperature per year in Söderarm, as well as a line of best fit:

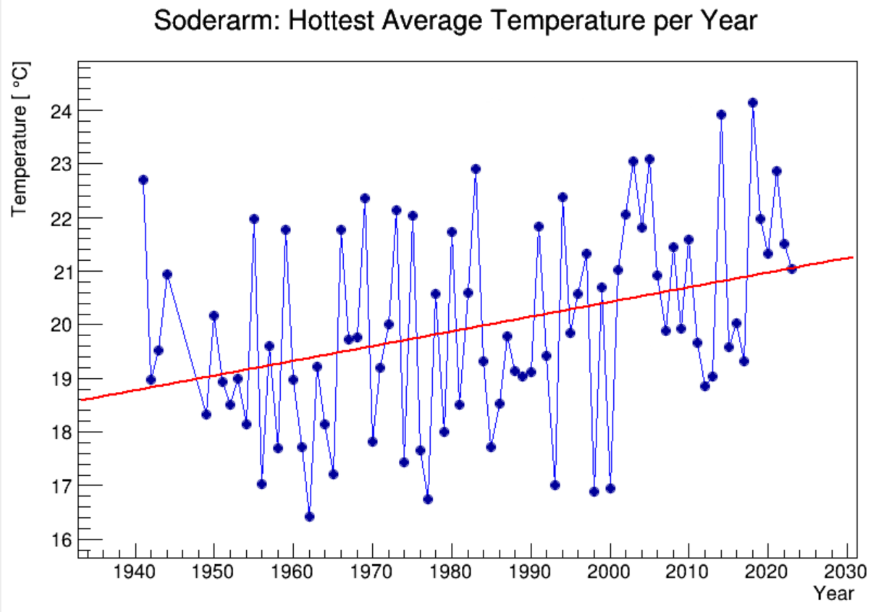

Figure 2: Hottest average temperature per year in Söderarm.

3.3 Discussion

Figure 2 demonstrated a warming of temperature in Söderarm over roughly 80 years. My results showed an average increase in temperature of 2.7°C whereas global averaging warming rate is around 1.8°C. This suggests that the temperature rise in Söderarm follows the upward global trend however is affected by harsher, regional weather typical to coastal Sweden and northern Europe. The variation between each point can occur due to multiple variables such as weather patterns and ocean effects. It is worth noting that there is data missing from 1944 to 1949 which may slightly skew results and add uncertainty to the results. Although uncertainties are present the positive gradient still signifies a clear warming pattern in the average hottest day per year.

4 Comparing the Average Temperature between Two Cities on Christmas

The purpose of this part of the project was to calculate the difference between the average temperature on Christmas day (December 25th) every year, between the cities of Lund and Luleå. The reason why this was chosen is that I have family near Luleå, while I live in Lund. Therefore, I thought that it would be interesting to see how different the temperature would be for us on one of the most special days of the year.

4.1 Method

Initially, the data was in large datasets in the form of CSV files, taken from SMHI. These CSV files contained a lot of metadata, therefore a bash script was used in order to remove this metadata, leaving just the date, time of measurement, and temperature for each CSV file. The CSV files that were used from this preprocessed data was: `Preprocessed_data/Lund.csv` and `Preprocessed_data/Lulea.csv`.

As the CSV files included data from every day over a span of over a decade for some cities, the next step was to filter this data. This was done using the `filterDate()` function, which is defined in the `filter_date.cxx` source file. This checks if the string in the date column of the CSV file matches that of the desired date, in this case 12-25. This resulted in the data consisting of several measurements from Christmas day from several years. However, the measurements were not always taken at the same time every year. Therefore, to obtain an accurate measurement that can be compared for several years, the average temperature was taken. This was done using the `calc_average_temp()` function which is defined in the `average_temp.cxx` source file. This function would return a map, which consisted of a string (which represented the year) and a double, which would be the average temperature. However, to be able to plot these values in ROOT, it would be easier to have them in the form of two vectors, one for temperature and the other for the year. This conversion was done using the `makeLists()` function.

The steps above are completed for both the data from Lund, as well as Luleå. As there are some years where the measurements were not taken, the data is once again filtered so that it only keeps the years where there is an average temperature measurement for both years. The differences in average temperature are then taken between the two cities, and in order to see if there has been any trends a 5 year moving average was added to the figure.

Finally, the results are plotted using ROOT and the figure was saved as a png in the `figures` directory.

4.2 Results and Discussion

The following figure was produced:

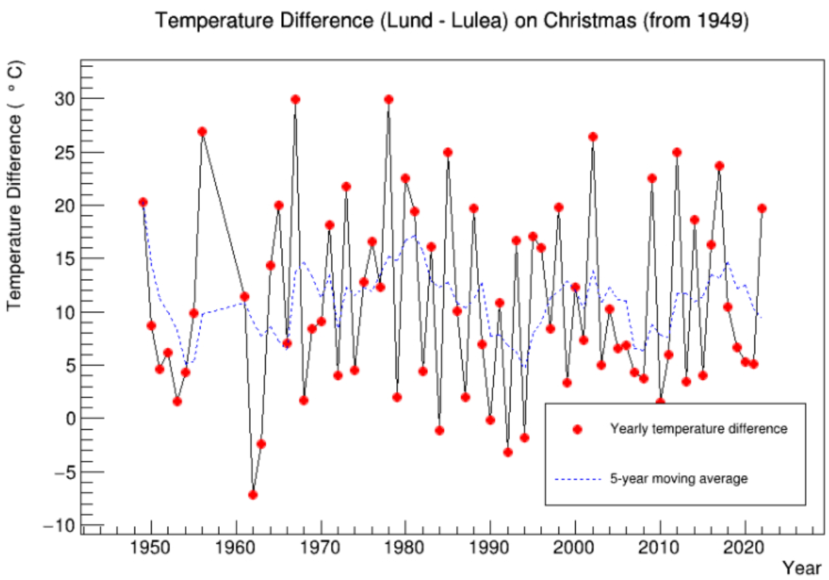

Figure 3: Difference in Average Temperature on Christmas between Lund and Luleå.

As seen in Figure 3 the largest temperature difference was in 1967 when Lund was approximately 30 °C warmer than Luleå. The other extreme occurred in 1962, where Lund was approximately 7 °C colder than Luleå.The 5-year moving average in Figure 3 illustrates a multi-year cyclical trend, which generally remains between a temperature difference of 5 and 15 °C.

This trend was unexpected, as one would expect Luleå to be consistently colder than Lund, as it is far more North. Therefore, negative temperature values, meaning that Lund was colder than Luleå on Christmas, is even more unexpected. These deviations could however be explained by short term weather anomalies, where Lund was either very cold, or Luleå was much warmer than usual. This explanation is supported as the trends do not last several years. A source of error could be that the temperature readings were not taken at the same time for both of the cities. This could bias the temperature readings, despite the average temperature being taken, as if more measurements were taken at night it would result in a colder value. One possibility for an improvement would be to limit the time in which the data is collected. For example only choosing data that is measured between noon and 3 pm for both sites.

One improvement that could be made to this code would be to show the average temperature of Lund and Luleå. This could have been plotted on a separate figure so as to not clutter the current plot. This would have allowed one to see if the temperature differences were due to large fluctuations in only one city, or if the average temperature simultaneously varied in both.

5 Heatwaves and cold waves for Uppsala and Lund

The purpose of this part of the project is to analyze the occurrence of heat and cold waves for the cities of Lund and Uppsala. The goal is to quantify the number of three consecutive days with temperatures over or under a certain arbitrary choice and how their frequency has changed over the decades.

1. Data and preprocessing

The analysis used preprocessed daily temperature data from SMHI for both stations. Each record contains a date, time, and temperature reading. The data were read from the CSV files `Preprocessed_data/Lund.csv` and `Preprocessed_data/Uppsala.csv` using a custom function `readCSV()`. This function reads each line of the file, splits it by commas, and stores the date, time, and temperature in a `Row` structure. The resulting vector of `Row` objects is then converted into the shared `DataRow` format for further processing.

Hourly temperatures were grouped by date to compute daily mean values through the shared function. The daily averages were calculated using the function `computeAveragePerDate()`. (defined in the common header "averagetemp.h"). The resulting datasets consisted of daily mean temperatures for each station.

2. Method

A heat wave can be defined as a period of at least three consecutive days with daily mean temperature above the one of ninety percent of all days (90th percentile). A cold wave can be detected similarly, but for temperatures lower than the 10th percentile.

We worked out the 90th and 10th percentile temperatures separately for Lund and Uppsala. To do that, we put all daily temperatures in order from coldest to hottest and then found the temperature that marks the top (or bottom) ten percent of days. When the percentile didn't fall exactly on a data point, the program took a value partway between two neighboring temperatures to make the result more precise.

Each daily time series was scanned sequentially to identify consecutive days that exceeded (for heat waves) or fell below (for cold waves) the determined threshold.

If a sequence of three days reaches the minimum or maximum temperature, it was counted as one event for that year.

We then store the number of detected events per year and then group them by decade. For each decade, the mean number of events per year was calculated. Finally, the results were visualised using ROOT's plotting tools, `TGraph` and `TCanvas`, in the function `create_plot()`. This function plots the decadal averages for Lund and Uppsala, adds linear trend lines, and saves the resulting figures as PNG images.

3. Results

Two plots were produced:

1. Mean number of heatwave events per year by decade

2. Mean number of coldwaves events per year by decade

Blue points represent Lund and red points represent Uppsala, with corresponding linear trend lines. Both series show a clear trend. The heatwave plot indicates an overall increasing trend in frequency for both cities, while the coldwave plot shows a decreasing or stable trend, consistent with a general warming of the Swedish climate.

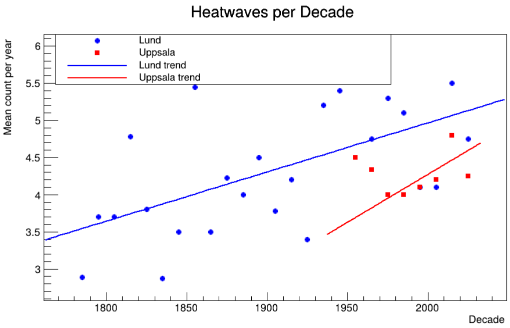

Figure 4: Caption for the first figure.

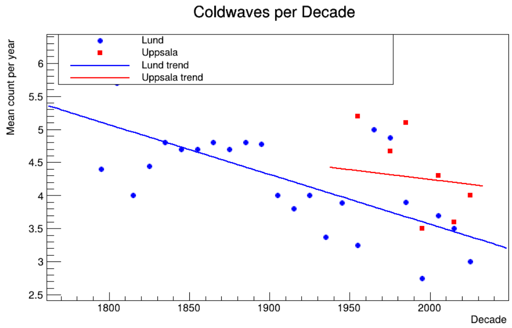

Figure 5: Caption for the second figure.

4. Discussion

Using percentile-based thresholds ensures that the definitions of “extreme” are relative to each station’s own climate, allowing a fair comparison between the two places. The decadal averaging reduces year-to-year variability and makes long-term trends clearer. When plotted with daily and yearly data, it wasn’t possible to observe a linear trend as clearly.
Also worth to mention that using the daily average temperature helps representing the overall temperature conditions of the day and smooths out changes like day–night cycles or single-hour anomalies. Secondly, we use the percentiles relative to all the data points we have, not just by year, since we want to detect long-term changes in the frequency of extremes, not changes in their definition. Using a fixed threshold lets us see how often those same “extreme” conditions occur over time.

However, there are two limitations in the current implementation which were realized later: Firstly, waves crossing New Year’s Eve may be attributed to the following year; Also, if the dataset ends during an ongoing wave, that event might not be counted, since we finalize the count when the code detects a non-extreme day. These issues can be resolved by tracking the wave’s start date and ensuring that ongoing waves at the end are counted. Overall, the analysis demonstrates an upward trend in heatwave frequency and a decrease in coldwave frequency over time, aligning with broader climate-change expectations.