MNXB11 - Final Project

Dominique de Boer, Kiam Persson, Marie-Philine Hebel, Ivana Varsányiová

30 oktober 2025

1 Project Goal

The goal of this project is to see whether the impacts of global warming can be observed with weather data in Lund. Two different datasets are used for this, one listing the temperatures at different times during the days from 1780-2023 and the other one, found on the `smhi.se` website, listing the amount of rainfall from 1863-2025. There were two sub-goals: the first one to see whether we can see a consistent rise in temperature over the years and the second one to observe whether we can see any trend happening to the amount of rainfall over the years.

2 Approach

2.1 Preprocessing data

2.2 Extracting data

Since the data is used slightly different when looking at the average per year or the average per day each year, and per sub-goal as well, the data is extracted in every source file separately. However, the general structure remains similar.

The constructor is called in the source file and defined to take the name of a file. This is used in the main file, so it can call the file in which the (cleaned) data is stored. The file name is saved as a variable to use in the function extracting the data.

The function starts with initializing the variables that are contained in the data file: the file, date, time and temperature. The first line of the data is a header describing what the numbers mean, but in order to create the lists, this header needs to be ignored, which is done with iostream's getline.

Then a while loop is created to iterate through the data file.

Using iostream's stoi function, the date can be separated into year, month and day. For instance, as the date is written as 2025-10-30, to get the year the four numbers from the first

number need to be used, thus 0, 4. These are saved as variables and can be used for if statements to filter the complete data file.
The contents of the while loop differ depending on the sub-goal.
Lastly, the file is closed to avoid a data leak.

2.3 Average temperature per year from 1780 until 2022 (Marie-Philine)

To create a histogram which displays the trend of the average yearly temperature a `class temp_vs_years` is created that includes all the relevant variables and methods.
For the purpose of this subtask the years and temperatures are extracted from the cleaned temperature dataset as described in 2.2. These values are stored in vectors.
The method `void GetYearlyAverage()` stores the yearly averages in a vector and calculates the standard deviation. A for-loop is employed to iterate through all the data. A variable `int current_year` is used to keep track of the current year. In this way all the temperatures measured in one year are summed and then divided by the amount of measurements each year to find the average. The standard deviation is given by

$$\sigma = \sqrt{\frac{\Sigma_i (x_i - \mu)^2}{N}}, \tag{1}$$

where x_i is a specific temperature, μ is the yearly average temperature and N is the amount of measurements per year. This is being calculated for each year with the help of `cmath` functions and stored in the `st_dev` vector. One has to be careful to reset the `count` and `sum` variables after a year is over. This is taken care of in the `else {}` statement.
The Histogram is created in the `void CreateHistogram()` method. The ROOt class `TH1D` is used to create a plot of one variable. The number of bins are defined to be the number of years and they are iteratively filled with the average yearly temperatures. The corresponding standard distributions are also added as bin errors.
The data is fitted linearly with ROOTs `TF1` class. The slope of of the fit was noted and then later added to the plot.
One plot was created without the standard deviation and another with it. This was done in order to have one 'clean' plot with just the averages but also showing the relevant information given by the standard deviations.

2.4 Average temperature on a given day for two periods (Dominique)

The goal is to find the temperature on a given day, set at February 15th, for two different periods, in order to see the possible effects of climate change on the mean temperature on 15/02 of these periods. The first period is from 1863 till 1913 and the second one from 1973 till 2023, because these periods have data available in the datasets for both temperature and rainfall, which makes a direct comparison between them easier.

To start, a class is created in a header file, which initializes a constructor, four functions and eight vectors. These are then implemented in the source file. Inside the source file, the data is first extracted in *ReadFile* as described in 2.2, creating a variable for year, month and day. Inside the while loop, an if statement is implemented to check if the day is the 15th and the month the 2nd, so the date is 15th of February. If this is true, there are two more if statements, targeting the years of the two periods. If these are true as well, then the year and temperature corresponding to the iteration are saved in the first four vectors initialized in the class. As there are two periods, there are two of each vector, corresponding to the first or second period.

In the temperature data there are often multiple measurements on the same date, but at different times. On those occurrences, the average temperature of the day is calculated in the second and third functions, *GetAverageTemp_1* and *GetAverageTemp_2*, which essentially do the same, but for both periods separately. Each function iterates through the two created lists, and if the year that is set as the first year is the same as the year that the iteration is at, it adds the temperature of that year to a sum and it counts how often the same year is repeated. Once the year the iteration is at is not the same as the current year anymore, it adds the sum divided by the count to a new vector and the year that it was done for to another new vector, both initialized in the class. It then changes the year to the next one in the list and adds the first value of that year already to the sum and count. If there are more measurements in the year, it is ready to repeat the previous loop and if there is only one, the values are added to the new vectors on its own. The last year needs to be added manually as the loop stops before the line in which the variables would have been added.

The last function creates three histograms of the last four vectors. All histograms have 50 bins and range from -15 degrees Celsius to 15 degrees Celsius, which is based on the contents of the lists. The first two are the histograms that show the entries of each temperature of both periods separately, in blue and red, including a text box that shows the statistics: the number of entries, the mean and the standard deviation. The third one shows them both at the same time, without statistics but includes a legend to show which is which. This is done using the TH1D, TLatex, TCanvas and TLegend libraries. They are all separately saved as pdf files with a given title.

2.5 Average daily rainfall each year from 1863 until 2022 (Kiam)

The goal is to examine the amount of rain from different years in Lund. The subgoals are to see whether the amount of rain each year has changed throughout the years, and to see if there is a relationship between the average temperature of a year and its rainfall.

A struct RainVsYears is created in the header file rain_vs_years.h. In the struct is declared an overloaded constructor taking the datafile as a parameter, a void function ReadAndDraw, and the necessary variables. The constructors and the ReadAndDraw functions are defined in src/rain_vs_years.cxx. The ReadAndDraw function reads a datafile containing the date and its rainfall, and with the data draws a histogram using ROOT. The histogram has an X-axis with the years 1863-2022. The Y-axis shows the daily average rainfall for each year in millimeters. A linear fit with it's calculated slope is also drawn.

In the ReadAndDraw function reads and extracts the year and the rainfalls for eahc year. A std::map is created where the key is the year, and the value is a std:.pair containing the sum of the rainfall that year, and the number of days that have been added to each year's rainfall. As the file is read, each years total rainfall and number of entries are put in the value corresponding to its key. Rainfall/entries, (or year's total rainfall/days) is done to get each year's average daily rainfall. The year and its avergae are put into two vectors. A histogram is then filled and draw it using ROOT's TH1D class and its functions. Using ROOT we will also do a linear fit to the histogram, and calculate the slope. TO BE CONTINUED

2.6 Average rainfall on a given day for two periods (Ivana)

The main aim is to get a histogram of the amount of rainfall on the same day (15th of February) over two different periods, both consisting of 51 consecutive years. This can be done in a very similar manner as the average temperature on a given day (see section 2.4). TBC

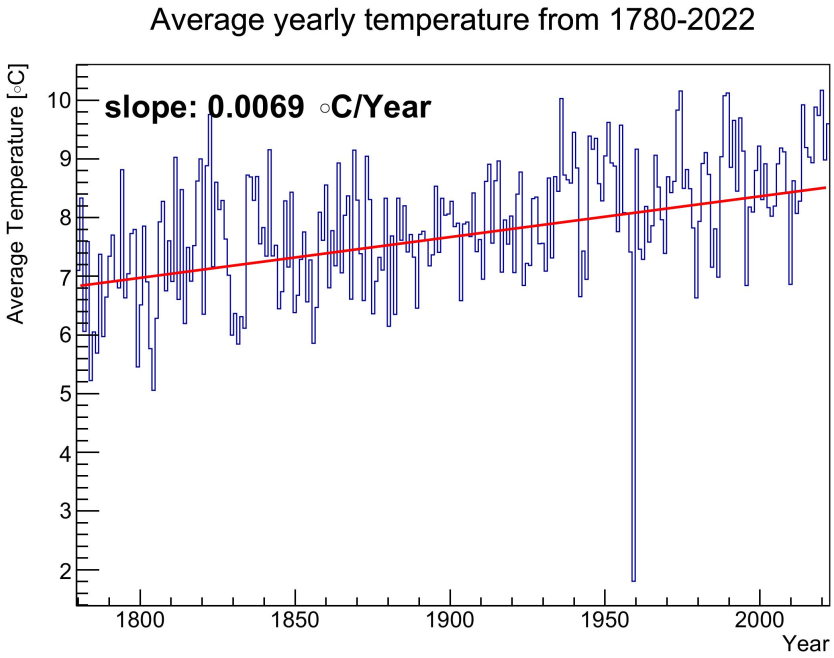
2.7 example for pic

??

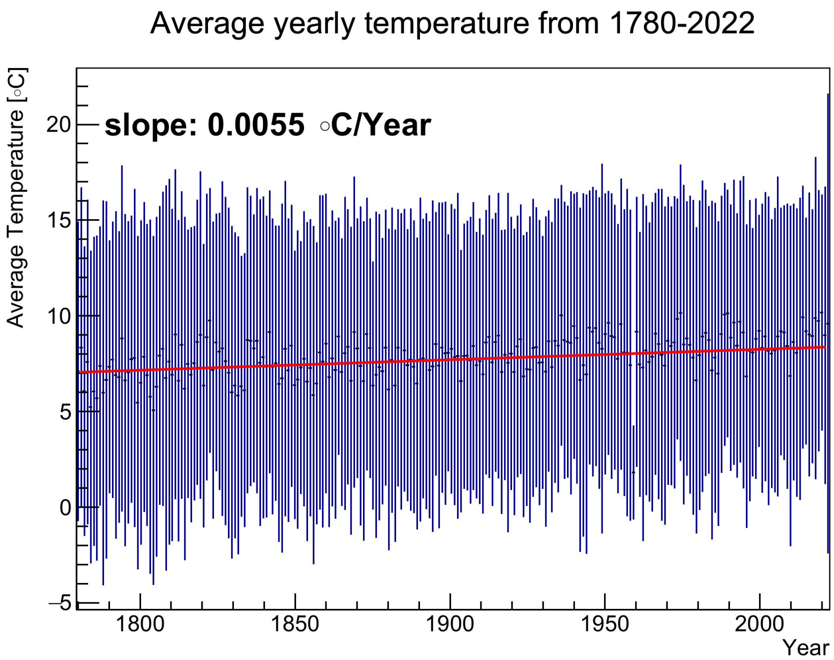
3 Results

3.1 Average temperature per year (Marie-Philine)

The average yearly temperature was plotted in Figure 1 and the same data was plotted with a standard deviation for each average value in Figure 2. The slope of the line is found to be 0.0069 °C/Year for the plot without standard deviations and with the weighted values from the standard deviation the slope is found to be 0.0055 °C/Year.



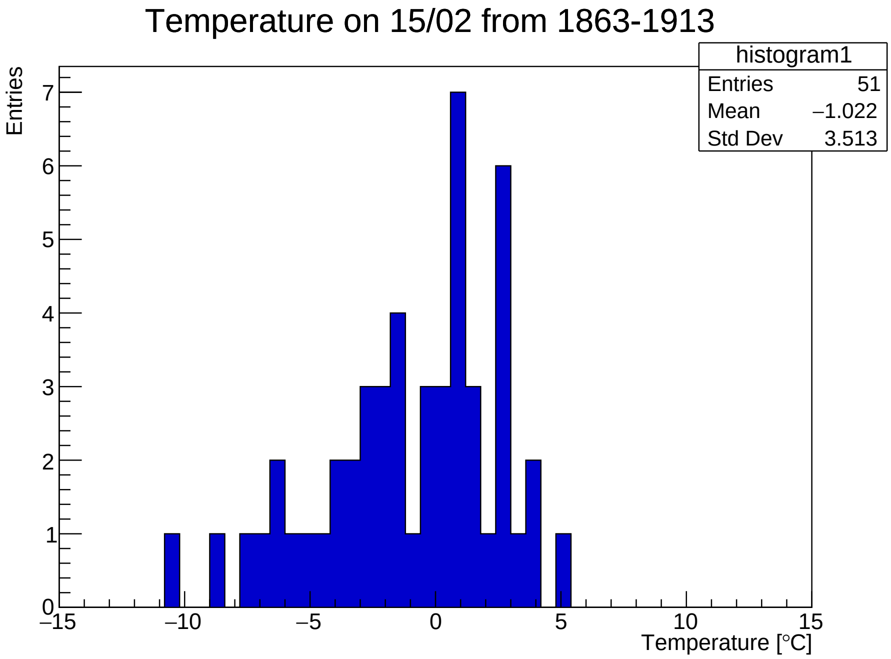
Figuur 1: The average temperature of each year plotted against years from 1780-2022. The data was fitted by a liner fit (red line).



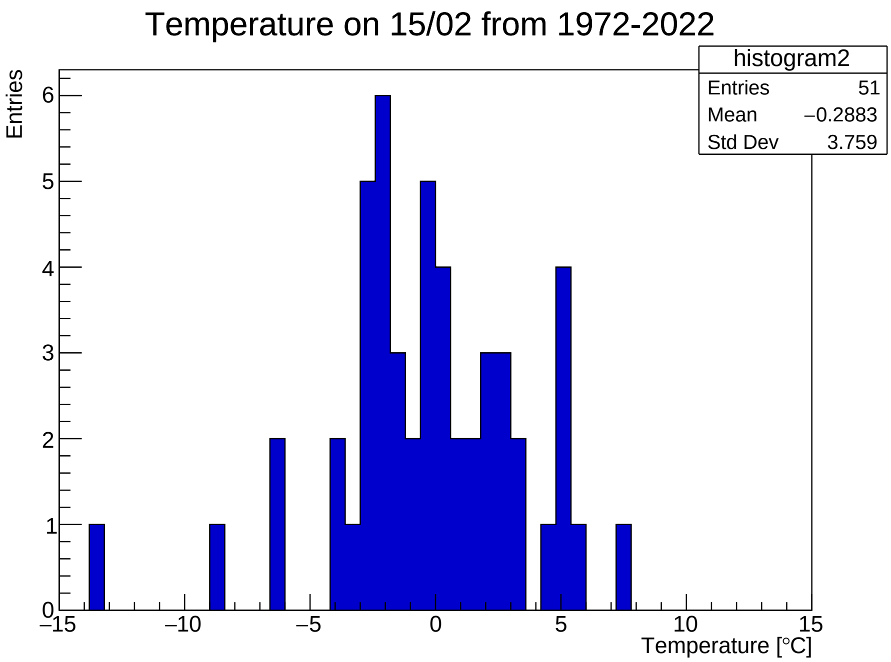
Figuur 2: The average temperature of each year plotted against years from 1780-2022. The standard deviation for each year is given as the vertical lines and the data is fitted linearly (red line).

3.2 Average temperature on a given day for two periods (Dominique)

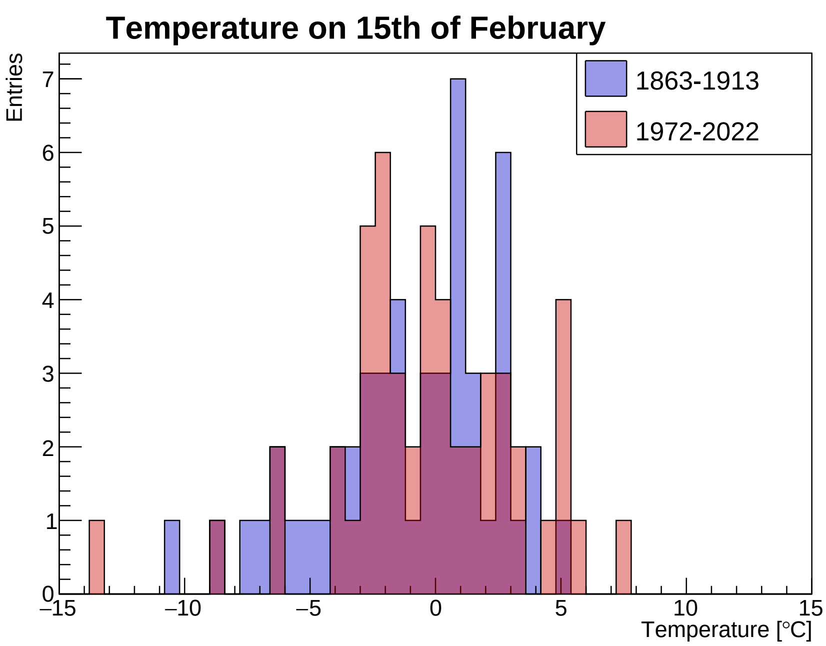
Figure 3 and 4 show the histograms of the first and second periods respectively. The temperature on the 15th of February from 1863 till 1913 has an average of -1.022 degrees Celsius. From 1972 till 2022 the average temperature was -0.288. Figure 5 shows the difference in shape of bins of figures 3 and 4.



Figuur 3: The temperatures measured in Lund on February 15th from 1863 till 1913

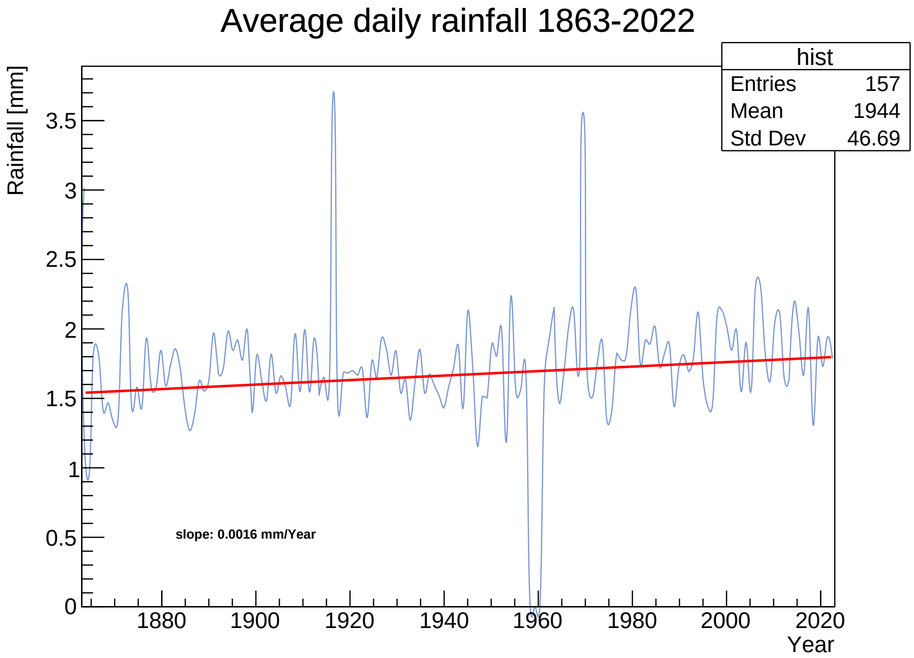


Figuur 4: The temperatures measured in Lund on February 15th from 1972 till 2022



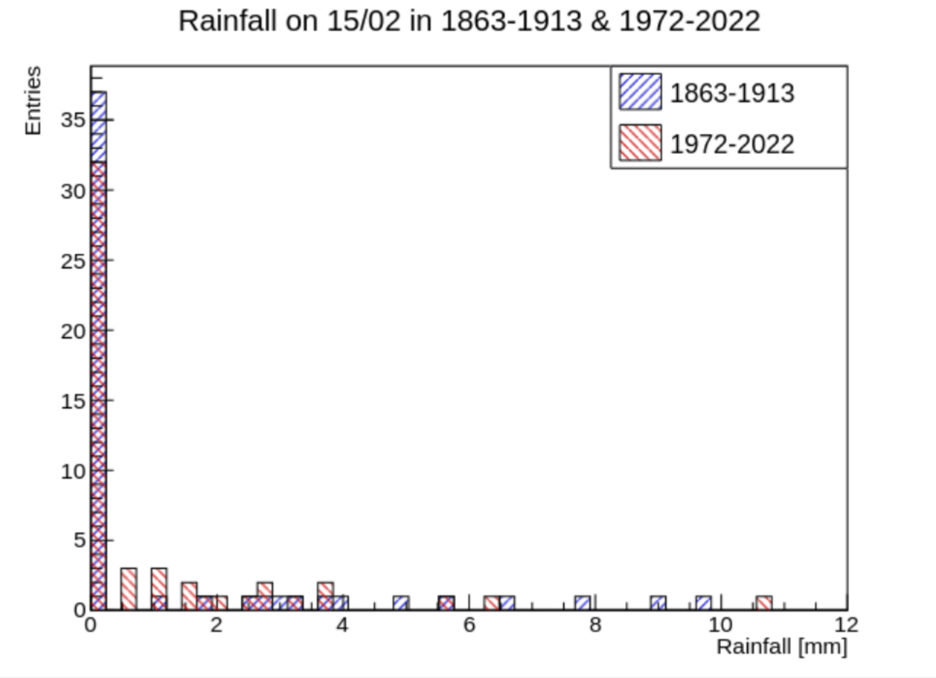
Figuur 5: The temperatures measured in Lund on February 15th from 1863 till 1913 and 1972 till 2022

3.3 Average daily rainfall each year from 1863 until 2022 (Kiam)

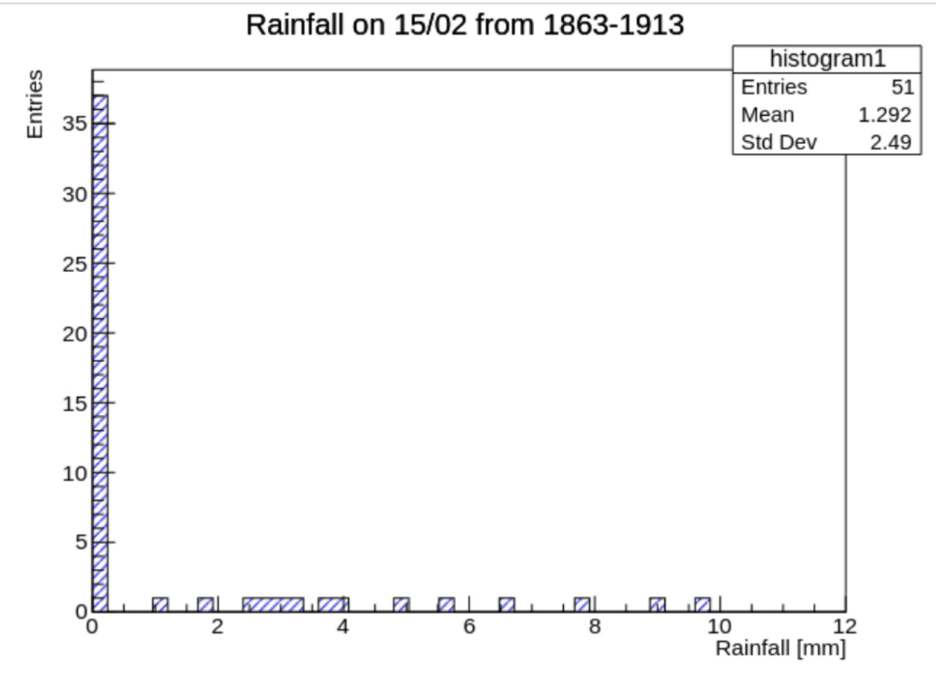


Figuur 6: Average daily rainfall each year in Lund from 1863 until 2022

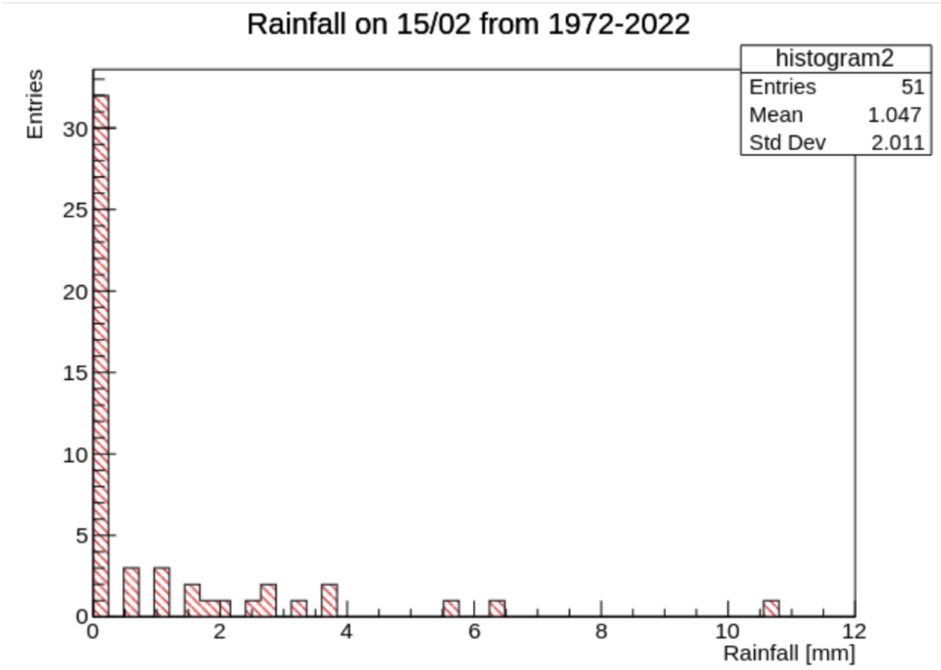
3.4 Average rainfall on a given day for two periods (Ivana)



Figuur 7: The rainfall measured in Lund on February 15th from 1863 until 1913 and 1972 until 2022.



Figuur 8: The rainfall measured in Lund on February 15th from 1863 until 1913.



Figuur 9: The rainfall measured in Lund on February 15th from 1972 until 2022.

4 Discussion

4.1 Interpretation of results

The plots that resulted from the temperature analysis showed a significant increase in temperature, which would prove climate change. On the other hand is there a slight increase in rainfall. The correlation between these two results is hard to find and understand if not done by a climate expert.

4.2 Unexpected data around 1960

In Figure 1 as well as Figure 6 one clearly sees an extreme drop in both temperature and rain. This is do to the data that was used, since there was no data recorded between the 1957.01.31 and 1961.01.01. Since the data recorded in 1957 is only from January, naturally there is a great drop in temperature. The drop in rainfall is somewhat surprising but could also be due to to the way the bins are filled in the missing years. This could possibly have been fixed by removing the data from 1957 and filling the bins in such a way that the bins corresponding to the missing years are skipped. The gap in data could maybe be solved by taking the data for those years from another weather station close by.

4.3 Deviation in slope when adding the standard deviations

When adding the standard deviations as error bars to the bins the slope decreased from 0.0069°C/Year to 0.0055°C/Year. This happened because the average values were weighted differently depending on the amounts of deviation. If the deviation was large the corresponding value contributed less to the linear fit.
In this context of average temperature in a year a larger standard deviation does not necessarily correspond to a less significant value which is why this automatic feature in the `TH1D` class is not very relevant. It could have been removed manually but it was decided against it in order to be able to discuss this feature.

4.4 Timing of temperature measurements

The time at which the temperature was measured is not the same for every day. Since the temperature changes throughout the day depending on the hour, this makes the results less reliable.
When there are multiple measurements at different times on one day, only the average is calculated, but the degree of spreading nor the amount of measurements was not incorporated, which can wrongfully reflect the average of a day.
Unfortunately, the data with which was worked can not be changed anymore.However, in future repetitions of this project, the approach to calculating the average and using the timing could be altered to include only those closest to one moment in the day.

4.5 Just one day

The analysis of the temperature on one day has only been done for 1 day, however results could be different if done for another day. Specifically days in different season would perhaps different results. If this project would be repeated, it would be interesting to take a day in each of the seasons, or at least in both winter and summer.

4.6 51 entries in a period

During the analysis of both the rainfall and the temperature over two different periods, only 51 entries (i.e. different years) were taken in both periods. This is nowhere near enough to get reliable results; however, this could not have been easily fixed, as the rain dataset does not have many more years. In order to make useful comparison, only years present in the rain dataset could be used from the temperature dataset.

5 Conclusion