MNXB01 Project

Thomas Eriksson, Emil Johansson, Albin Jonsson, Saeid Pirani

November 11, 2018

Lund University

1 Comparison of Coastal and Inland climates

For the task not listed in the project description, a comparison is chosen. The idea is to compare the measurements of Visby, located on the small island of Gotland, to another set of measurements taken at an inland location. The produced histogram should thus highlight the differences of the two locations.

One way in which the temperature of coastal regions differs from those more inland is that the temperature changes overall more slowly. This is due to the large body of water acting as either a massive heat reservoir in the autumn of taking large amounts of energy to heat up in spring. Thus, one way to compare the locations would be to plot the temperatures in these transitional months of the year and see the temperature difference.

The second location is chosen to be Boras. It is chosen as it is of approximately the same altitude as Visby and is not a coastal city.

1.1 Code structure

To achieve this, the first step is to read the data. The method used is the **getline()** command, that saves the important data to strings which are then converted to integers and doubles. Before this, in order to make the data in the .csv file more manageable, a bash script is called with the **system()** function. The goal is to trim away all unnecessary text and only leave measurements of green quality marking.

This is all done in a while-loop, which can also be used to extract the temperatures of desired dates. As the only distinction needed for this comparison is to pick out measurements originating from a chosen month, a simple if-statement separates these from unwanted ones. A small script inside the statement calculates the temperature average for each day. These averages are then added to the histograms. To demonstrate more clearly the differences in temperature between the locations each histogram is fitted to a Gaussian function.

1.2 Results

When plotting the results of the comparison in May month, figure 1 is created. As is easily seen, Boras tends to have higher temperatures than Visby during this period. While this coincides with the hypothesis, it alone does not prove it.

To further strengthen the hypothesis the temperature difference in autumn must also be investigated. In figure 2 the temperatures for the two locations in September are shown.

Here, as predicted, Boras has the lower temperature. The water retains the heat of summer, causing Visby to decrease in temperature more slowly than for inland locations.

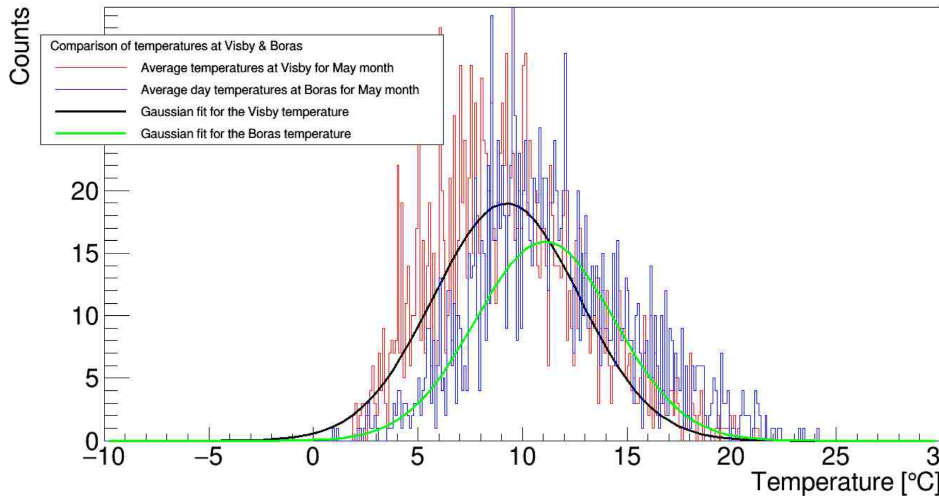

Figure 1: Comparison between the Visby and Boras temperatures in May

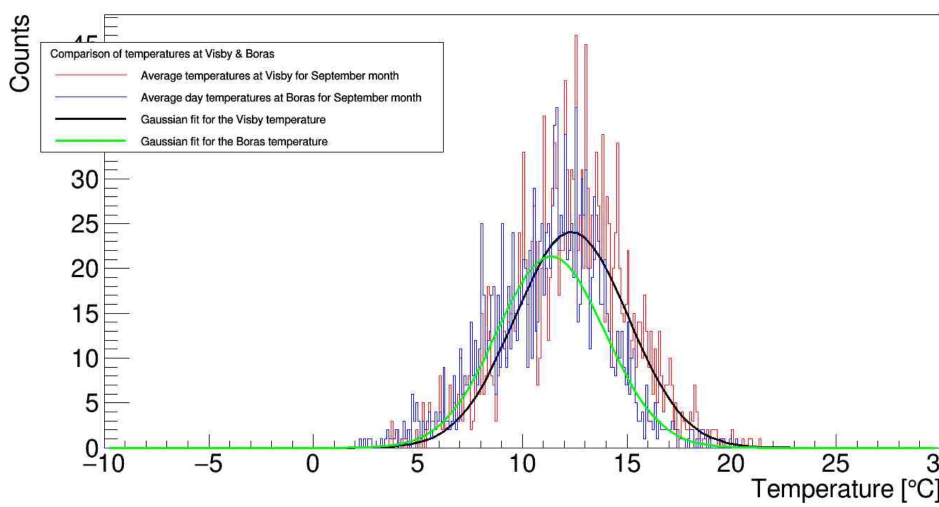

Figure 2: Comparison between the Visby and Boras temperatures in September

2 Mean Temperature Each Day of the Year

The purpose of this exercise, referred to as "3.2 The temperature for every day of the year" in the project instructions, is to create a program which returns a plot of the mean temperature for each day of a given year. The plot should also contain error bars representing the standard deviation of the corresponding mean values.

For this program a data set of temperature measurements in Lund was used. This data set ranges from 1961 to 2015. My objective was to write a program which could produce the desired plot for any of these years.

2.1 Code

The code is written such that it should be able to run in ROOT since that is where we want to produce the plots.

Firstly the SMHI data set file, containing lines of text and superfluous data, had to be trimmed down to just provide data interesting to this exercise. This was done using a bash script. This bash script trims down the data so that descriptive lines of text is excluded using the **cut**-command. Every line of data has a quality marker, and using the **grep**-command the unapproved lines of data could be removed. This altered data is then saved to a new textfile. This bash script is called with the **system()**-function and can therefore be executed within the main program.

This textfile is read using a **while** loop, where the **getline()** function is used to extract strings of information. These strings are later converted into integers/doubles. Since this loop reads the entire file, an input variable specifying the desired year that is to be plotted is used to create an **if**-statement where the mean temperature values for each day and their associated standard deviation values are determined and stored in a histogram previously defined. This histogram has 366 bins corresponding to the days of the year, each bin is therefore filled with only one value. This way only values for each day of a specific year is plotted.

Exiting the **while** loop the file is closed and using the **Draw()** function the filled histogram is drawn with error bars.

2.2 Results

Since the purpose of the exercise was to simply plot the mean temperatures for each day with error bars representing the standard deviation no further analysis of the data was performed. The written program takes a given year and plots the temperatures for that year. The program does not take leap years into account, on non-leap years the 366th bin will be empty. This is something that could be done if we would perfect the code further. Figure 3, 4, 5, 6 show plots for the years 1961, 1980, 2000 and 2011 respectively. Observe that 1980 and 2000 were leap years.

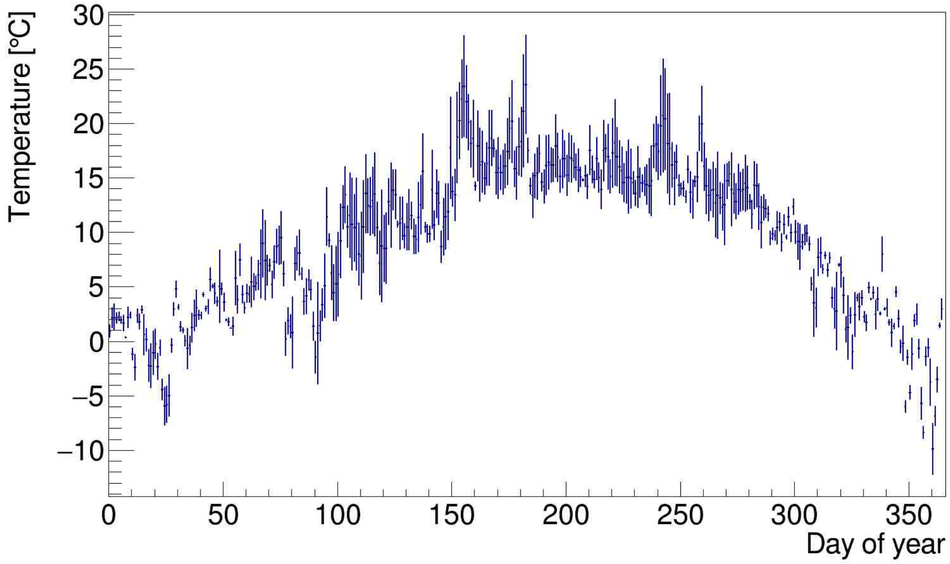

Figure 3: Mean temperatures for each day of 1961.

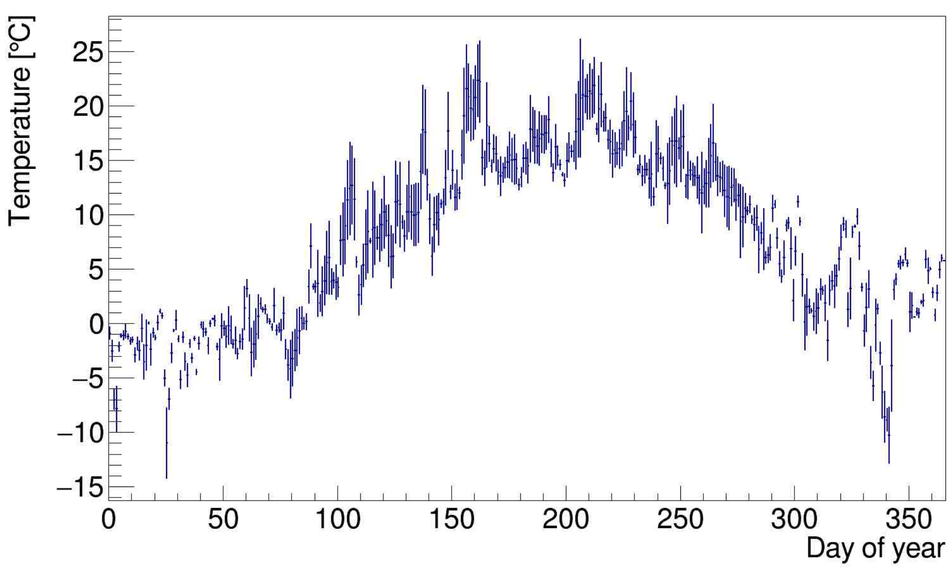

Figure 4: Mean temperatures for each day of 1980.

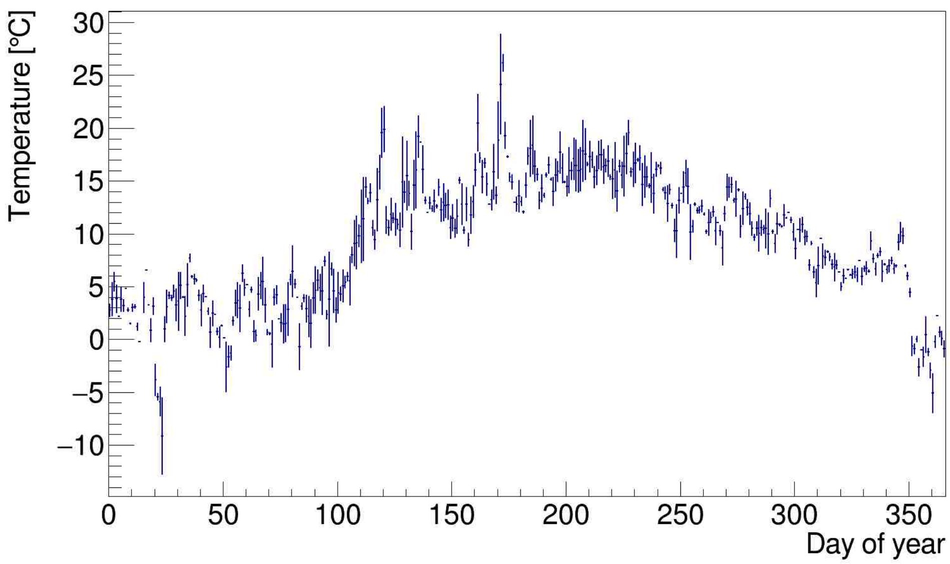

Figure 5: Mean temperatures for each day of 2000.

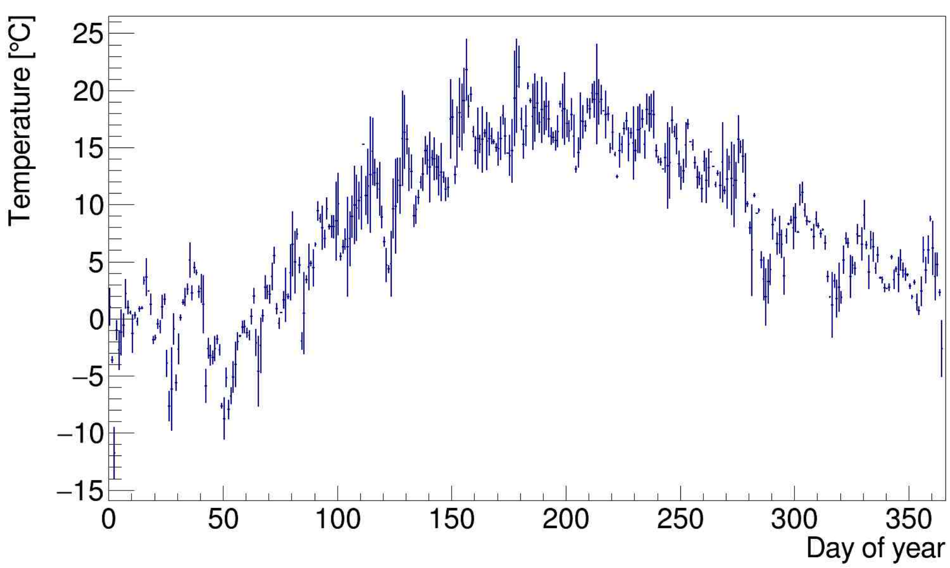

Figure 6: Mean temperatures for each day of 2011.

3 The warmest and coldest day of each year

In this work measured temperature data between 1961-2015 years for Lund have been gotten as input data then coldest and warmest days in each year are extracted and plotted by Root as the output of project in this section. It is illustrated that how appearance of coldest and warmest days are changed during the 1961-2015 period.

3.1 Code structure

In the 1st step, Bash scripting is performed in order to extract practical data from available "smhi-opendata-Lund.csv" file. Bash output provides required data in separate .txt files as: year.txt, month.txt, day.txt, time.txt and temperature-01.txt, this separate data file extraction strategy is for possible error checking. Only G data have been extracted and Y data considered as low quality data. In the 2nd step C++ coding is performed to process imported data in order to extract coldest and warmest days of each year. These data will tell us variation in appearance of coldest and warmest days in different years. C++ section outputs are stored in two .txt files as "WarmestDaysHist.txt" and "ColdestDaysHist.txt". In 3rd step provided .txt files from C++ are used to fill histograms, fitting histograms provide mean values as expected day for happening of coldest or warmest day in the year based on available statistics.

3.2 Results

Figure 1 shows plotted results in Root for Variation of coldest and warmest days in different years between 1961-2015. Black lines show Gaussian fit, for warmest days it lead to mean value=196.5 which it is the predicted warmest day during a year in Lund. Following bulleted issues which are requested to be covered by performing this project are done except the fitting part related to Coldest days and getting mean value for Coldest days.

- Create histograms of the warmest and coldest day each year.
- Predict when the warmest and coldest day is most likely to occur.
- Can you show both histograms in the same plot?
- Can you make a t function that wraps around as in Figure 3?

4 The mean temperature of each year

The general structure that I used to preform this task was the suggested with a class called tempTrender. Therefore a header file where the class is declared was used. The file **tempTrender.cpp** contains the implementations of the constructor and method used for my task. Instead of using the **project.cpp** file the

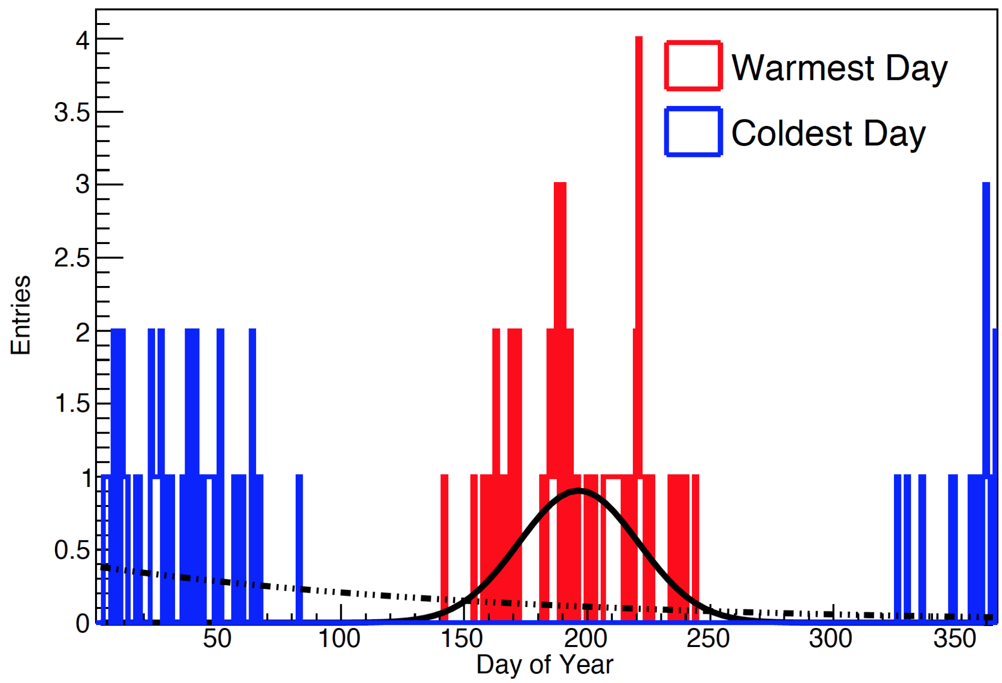

Figure 7: Variation of coldest and warmest days of year between 1961-2015. Black lines shows Gaussian fit

function within the file was moved to a common file for all group members. This function creates an instance of the class, and runs the method **tempPerYear** for that instance.

The wanted result is plots of histograms with the average temperature of each year. The first thing that needs to be done is to read in the data file. This was done in the implementation of the constructor in **tempTrender.cpp**. After that the wanted data needs to be extracted from the file. Looping over all strings, and with various if statements, the wanted data could be selected. Since this project wants the average yearly temperature, only the year that are completed were used. Also in some files there are only data points from 06.00 and 18.00 at later years, while at the start of the measurement there were three at 06.00, 12.00 and 18.00. To solve this only the data points from 06.00 and 18.00 were used. The temperature values were assigned to an array, were each data point corresponds to a year. A counter for the number of temperature measurements per year was also used. With these two a yearly average temperature could be calculated. Also the mean of the average yearly temperature from all the years could be calculated.

In the method **tempPerYear** a histogram of the average yearly temperature was made. After that a histogram of the mean of the average yearly temperature. A third histogram with either the mean of the average yearly temperature or average yearly temperature, depending on if it is larger or smaller then the mean of the average yearly temperature, was also made. These are to create the deviation from average effect that was requested. Together with these two lines can be seen. The black line is the moving average, which takes the closest 5 points including the year itself. The green line is a fitted arctan function. An arctan function was chosen because it seems to fit the data well. Since an increase around late 1980s can be seen, the arctan = 0 was set at 1987. Because of the chosen function to fit the data to, a unrealistic value for the mean temperature of year 2050 was calculated to 8.3°C. Although one could argue that more could not be concluded from the current data.

As the code was made to be able to handle any of the datasets but Uppsala. Lund is shown in the figure below.

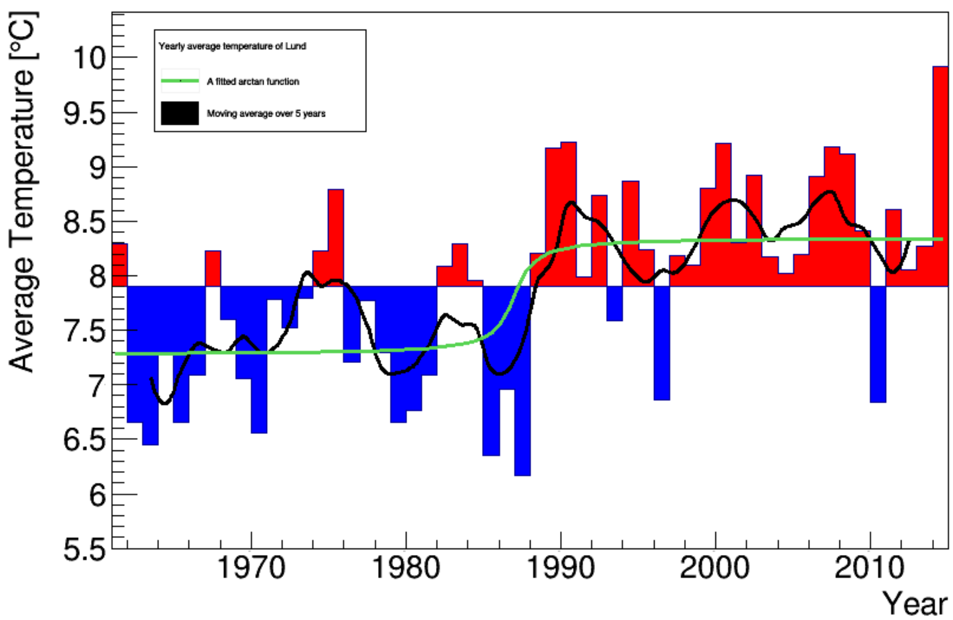

Figure 8: Yearly average temperature of Lund