MNXB11 Project Report

Vera Grönvik Hende, Marta Villalba, Astrid Lopez

November 1, 2023

Check out our repository by clicking here.

1 Introduction

The aim of this project was to conduct three separate analysis on air temperature data retrieved from the Swedish Meteorological and Hydrological Institute (SMHI), using C++, ROOT, and bash scripts. SMHI keeps an extensive amount of temperature data recorded at a number of stations, and this study only bases its analyses on the data for only one station: Lund. The reason why Lund was chosen as the primary study area is due to the long temporal extent of the data available for that station, as well as it being marked as good quality for the majority of the data set. The file contains information on the air temperature, the date, the time, and the quality over a time period of approximately 1780-2023 (SMHI, n.d. a). This report aimed to conduct the following three analyses: a) the temperature of a given day, b) the temperature change in Lund from 1990 to 2022, and c) the warmest and coldest day of each year.

All scripts used in this project can be viewed in our GitHub repository. The structure of the script can be divided into two parts: data preparation and analysis.

2 Data preparation

2.1 Bash cleaner script

The first step was to create a cleaner.sh file using bash script. The purpose was to clean the SMHI datasets by eliminating the headers, the semi colons and the dashes and replacing them with spaces to make it easier for C++ to process the information. Additionally, the script selected only pertinent columns to streamline the data, generating a concise and more interpretable format. The result were csv files that C++ could read and understand in the Data Extraction section.

2.2 Data Extraction C++ script

After the csv files had been cleaned, they were ready to be extracted. The purpose of the data extraction script was to parse the information from the csv file and store it in a ROOT TTree within a TFile. For this script we made use of a date library (Howard Hinnant) as well as a csv parser library (Ben Strasser). These libraries are further referenced within the GitHub repository.

The script reads in the line for each column and stores it as a variable for the TTree. The columns for date and time was slightly more complicated than the others, and were split up in different ways. The parsing for the date column uses the date.h library and saves each part of the date (year, month, day) into separate variables (branches for the TTree). The time column proved to be trickier to use the date.h library for. Instead, only the hour within the time column was stored as a variable for the TTree. This was done by first saving the time part as a string (with the format hh:mm:ss) and then removing everything after the hour, before turning it into an integer. The TFile created by this script is named output.root, and works as the basis for each analysis.

2.3 Weather Data C++ script

The aim of this script was to work as a check to filter out data points that have not been marked as good quality by SMHI. The suspicious data is marked with a Y and the good data with a G. The script

defines a boolean function to filter all data out that is not marked with G. This function was then to be called on in the analysis scripts in order to only include good quality data within the analysis. This script did not, however, get implemented in the final version of the analyses. These shortcomings are further discussed in Section 4.1.

3 Analysis

All analyses have their separate set of translation units with a header and source file. They are all called Analysis [underscore] number [dot] h/cxx.

3.1 Analysis 1: The temperature of a given day

In the first analysis we tried to create a histogram to show the temperature data for a given day, specifically December 25th. In this way, the graph could8 show how many times a certain temperature had been observed throughout the years. The code that can be seen in the repository extracts and collates pertinent details encompassing the year, month, day, and associated temperatures from this particular date to create the histogram in graph 1. Our results show that the temperature was mostly around 1 to 4 degrees Celsius on Christmas day. An important distinction between the example given in the project's instructions and our graph is that we rounded the temperature values to integers instead of doubles.

The histogram created had some limitations. For instance, it is quite unlikely that the temperature never dropped below 0 degrees Celsius. Also, the x axis could not show the whole numbers and instead had a lot of decimal places. These problems could not be overcome despite trying.

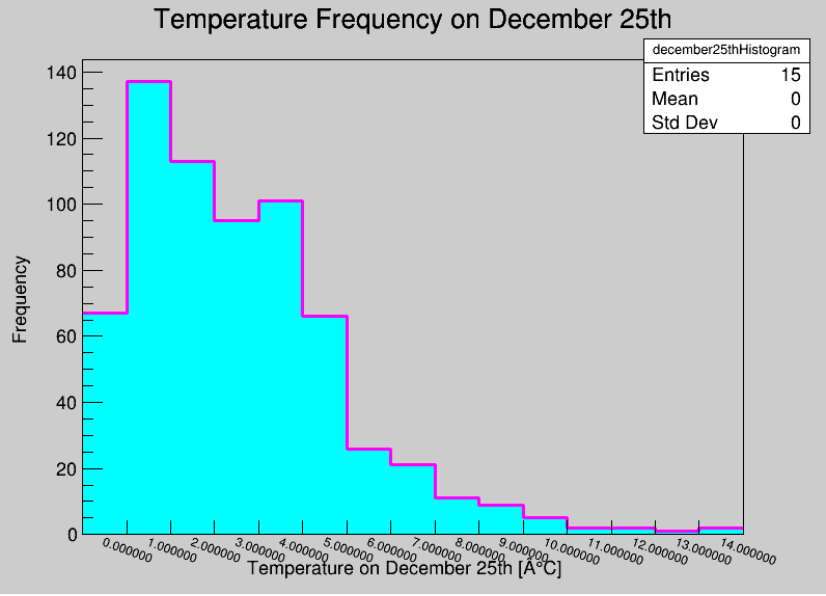

Figure 1: Temperature on December 25th in Lund.

3.2 Analysis 2: Temperature change in Lund from 1990 to 2022.

The second analysis script plots a TGraph for the yearly mean temperature from 1990 to 2022. It also plots a trend line using a linear regression. The legend displays the equation for the linear regression. The range 1990-2022 was chosen for two reasons: 1) there was missing data around 1960, and therefore any analysis covering that time period would be unreliable. 2) 30 years is typically used as the range for a climate period, and thus the last 30 years seemed a reasonable choice for investigating temperature change.

Just like the first analysis, the script for analysis 2 retrieves its data from the ROOT TTree created in the data extraction script. This analysis however only makes use of three branches from the TTree:

air temperature, year, and quality. The script creates two maps to store the sum of temperatures and the count of data points for each year. It then loops through the data in the data tree, calculating the sum and count of temperatures for each year within the range 1990 to 2022. A TGraph is then created, in which the mean temperature for each year is plotted. The final calculation of the means thus occurs in the function where the graph is being populated. The trend line is being calculated using the ROOT TF1 class. After performing the fit, the script extracts the intercept and slope, representing the y-intercept and the rate of change in the data, respectively.

The results displayed in Figure 2 appear reasonable. The linear regression fit is a quite simple model to use for this data set but it gives an adequate approximation of the trend in the data, considering the rather limited scope of this study. The trend indicates a temperature increase, corresponding with the expected results for this analysis. The temperature ranges seem reasonable as well, when they are compared to yearly air temperature statistics from SMHI (SMHI, n.d. b). Not all years included in this analysis were available at the previously mentioned source, but those that were seem to have their air temperature within the same range as the produced results.

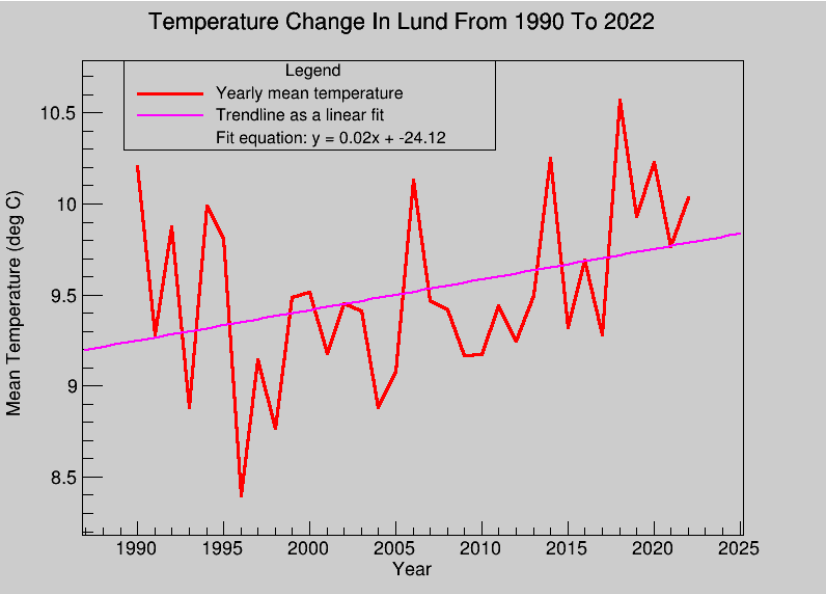

Figure 2: Temperature change in Lund from 1990 to 2022.

3.3 Analysis 3: The warmest and coldest day of each year

In the third analysis we want to do a histogram that shows the warmest and coldest day of every year in the city of Lund. We will need to include both the coldest and warmest day of the year histograms in the same plot. For that we used a Gaussian function. The variables used from the data in the ROOT TTree are temperature, day and year. Unfortunately the code was not working correctly so we could not get any histogram.

We think that the problem with the code could be related with the incorrect order of the code, as it shows a big amount of problems with the parenthesis. Also it can be because of not perfectly defining the Gaussian function, based on the code provided by the pdf of the project.

4 Discussion points

4.1 WeatherData translation units

Analysis 2 (long-term temperature plot) tried to implement the boolean function defined in WeatherData. However, when implementing it, the graph plots very strange results. This could in theory be due to the WeatherData function filtering out large amounts of data. When looking at the data it is marked as 'G' quality for all data points included in the analysis, so the issue does not seem to be that the WeatherData simply removes data points due to the quality being other than 'G'. We could not

however determine a fault in the boolean function itself. Further troubleshooting might determine the cause as to why it causes the data to behave strangely. The faults could not however be determined within the time constraints of the project. Therefore, the function is not included in the final script.

4.2 Macros

The inclusion of ROOT macros would have potentially allowed us to create better looking plots. However, due to time constraints, this step was skipped.

4.3 Analysis 3

The code is not working correctly. Therefore, in order to compile the script as is, the parts in the main() function in main.cxx for analysis 3 has to be commented out. The line indicated for the object file for Analysis 3 should also be removed from the makefile in order to compile. This way, the scripts for analysis 1 and 2 will still compile.

4.4 Potential additional issues

The raw data set could have been filled with irregularities in terms of placements of semi-colons between columns, or spaces within columns, that might have caused issues for our scripts as the bash cleaner file used these characters as a basis for the cleaning process. This would in turn have caused issues for the C++ scripts, and it may for example have been the cause as to why the weather data function did not produce expected results.

Another potential issue is if the number of measurements per day, month, and year varies throughout the data set. This may have caused complications for the calculations in the analysis scripts.

5 Conclusion

Thanks to this project we were able to learn new things about ROOT and C++ and we were also able to show all the knowledge that we acquired during this course.

This project managed to clean the data in bash and successfully extract it in C++ and produce an output file that can be viewed in ROOT. All analyses did not however produce the expected results and thus not all of the research questions could be answered. We ran into several known complications during the course of the project, and it is possible that the scripts contains further issues not directly visible to the authors. Some of the shortcomings of this project could have been improved given a longer timescale, but a better structure for the work within the project group might also have improved the results. A structured work plan was never created at the start of the project. However, given the relatively short time scope, as well as the relatively limited initial knowledge of both C++ and ROOT at the start of the project, the results produced could be considered adequate.

6 References

1. Swedish Meteorological and Hydrological Institute (SMHI) (n.d. a) *Lufttemperatur (h): SMHIs stationsnät: Lund*. [Data set]. SMHI Öppna Data. Retrieved October 23, 2023, fromhttps://www.smhi.se/data/meteorologi/l
ner-meteorologiska-observationer/param=airtemperatureInstant,stations=core,stationid=53430

2. Swedish Meteorological and Hydrological Institute (SMHI) (n.d. b) *Års och månadsstatistik*. [Data set]. SMHI Öppna Data. Retrieved October 29, 2023, from https://www.smhi.se/klimat/klimatet-da-och-nu/manadens-vader-och-vatten-sverige/manadens-vader-i-sverige/ars-och-manadsstatistik