MNXB11

Final Project Report

Sebastian Magnusson, Artis Vijups

2024-11-01

1 About the Project

In this project, the user provides a minimum and maximum temperature that establishes a range. Then, histograms are plotted summarizing three results:

- Each year, what was the ***count*** of days in the range?
- Each year, what was the longest streak of days within the ***range***?
- Each year, what was the longest streak of days within a range of the same ***amplitude***?

We create a cubic polynomial fit and determine the mean and standard deviation for each of these three results.

1.1 Collaboration Policy

We used *policy B* in this project: all members of the group were collaborators and pushed changes directly to the `main` branch.

We chose this policy because we had a small group (two people) and good communication through Discord, which made the commits more predictable.

2 Files, Classes, Methods

2.1 Cleaning the Data

A bash script takes the users inputted files, cleans them, and returns a .csv file for each file. It works for files formatted in the standard SMHI format.

The `clean_data()` function reads the input file, removes all unnecessary lines and outputs the columns of interest.

2.2 C++ Class

A class called `AnnualData`, written in C++, reads the file and the time the user wishes to consider, records the years that appear in the file and computes the necessary results.

`AnnualData::years()` is used to retrieve the vector containing the years.

`AnnualData::count(float low, float high)` returns a vector containing the number of days within the interval `[low, high]` for each year.

`AnnualData::range(float low, float high)` returns a vector containing the largest number of consecutive days within the interval `[low, high]`.

Finally, `AnnualData::amplitude(float width)` returns a vector containing the largest number of consecutive days within an interval of amplitude `width`.

2.3 ROOT Script

The ROOT script `newHistData.C` performs analysis and visual representation of the annual temperature data.

The `drawHist` function creates histograms of yearly temperature data using ROOT libraries and fits a polynomial curve to them.

The main `newHistData` function takes temperature range parameters and city information and then calls `drawHist` to generate and save histograms.

2.4 Building and Running the Project

`Makefile` is used to build the project by typing `make`.

`plot` is a Bash script used to simplify running the program. The standard usage is:

```
./plot <min_temperature> <max_temperature> [<city> <time>]
```

2.5 Other Files

`README.md` and `datasets/README.md` contain instructions for building and running the project and for using the cleaning script, respectively.

`CHANGELOG.md` contains the versions of the project and the changes made.

`Workplan.md` contains the initial work plan for the project.

`rootlogon.C` provides formatting instructions that ROOT runs by default.

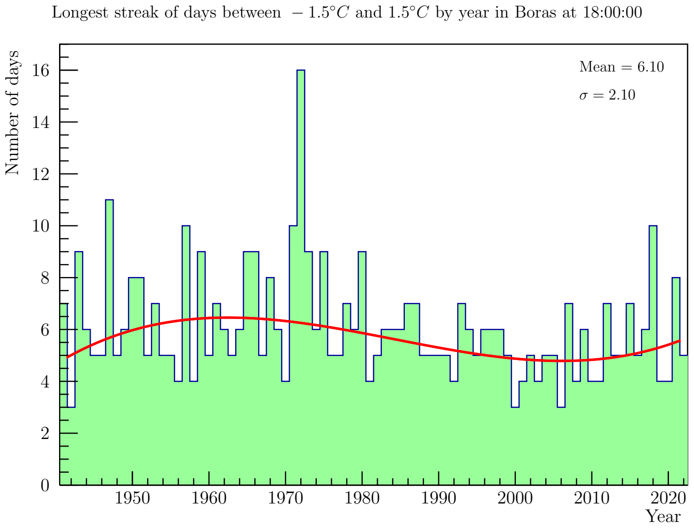
`.clang-format` defines the C++ formatting rules for the project.

`LICENSE` contains the used license and `AUTHORS` lists the project creators.

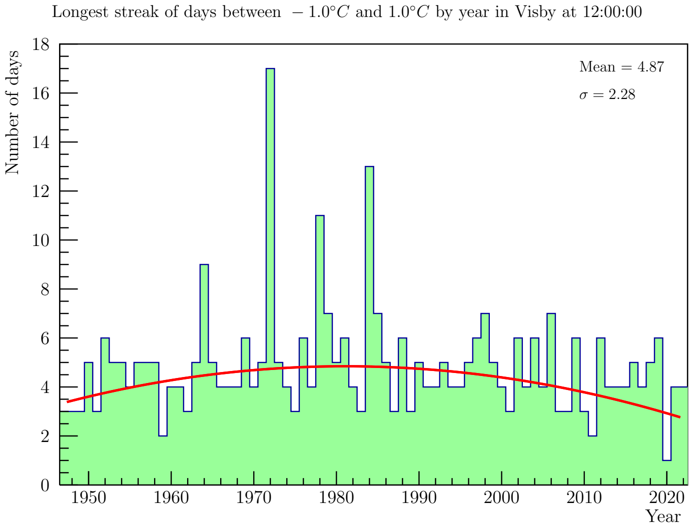
`Report.pdf` contains this report as exported with Overleaf.

3 Some Interesting Results

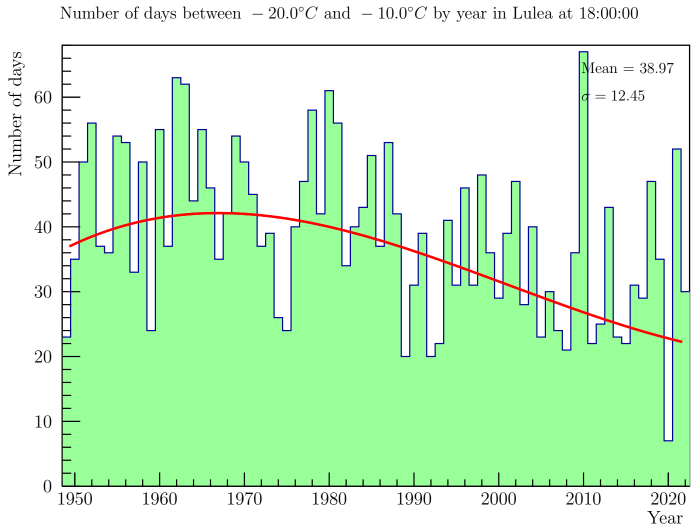
In 1972, there was a 16-day long streak in Borås where the temperature was around 0°C at 18:00:00.



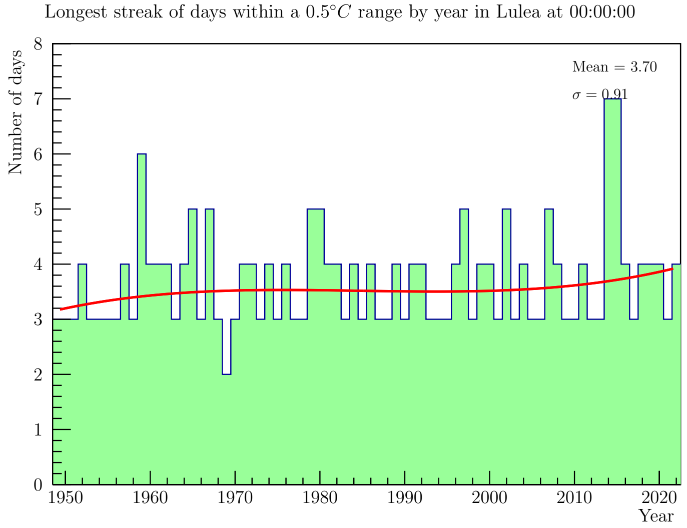
Similarly, in Visby, 1972 saw a long 17-day streak of around 0°C at 12:00:00.



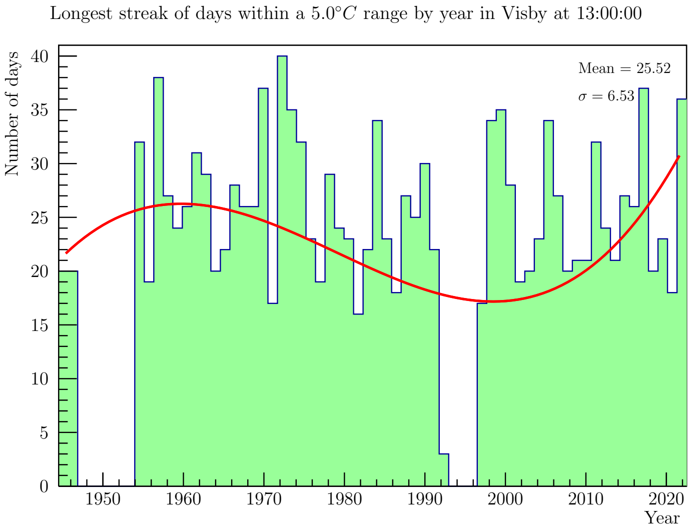
In Luleå, 2020 was an outlier with only 9 days between $-20°C$ and $-10°C$ at 18:00:00. All other years had at least 20 such days.



Also in Luleå, two consecutive years had a 7-day streak of temperatures within half-a-degree Celsius at midnight. No other years saw such a long streak of similar temperatures.



Finally, a histogram for Visby at 13:00:00 shows that this particular dataset had incomplete data for the provided time.



4 Conclusion

The project's goals were achieved: the annual temperature data was analyzed by generating histograms that depict the count of days, streaks and amplitude variations within a given range.

The created histograms revealed interesting patterns, such as notable streaks and outliers.

4.1 Possible Improvements

The main possible improvement is that an output text file could be provided, describing specifically when long streaks or other points of interest occur. As it stands, there is no way of knowing on which days these happen without scouring through the CSV files manually.