MNXB01-project

Josefine Frid

November 2022

Introduction

The project of this course was to use data from SMHI (Swedish Meteorological and Hydrological Institute) to create graphs using the content of the course. By using what we learned in the course, bash, C++ and ROOT. The graph done in this project is showing the mean temperature in each month of the year that the user types in. The graph was supposed to be the group's own idea.

Observations

Down below are the graphs for the mean temperature in each month of the year 1863, 1900, 1950 and 2020.

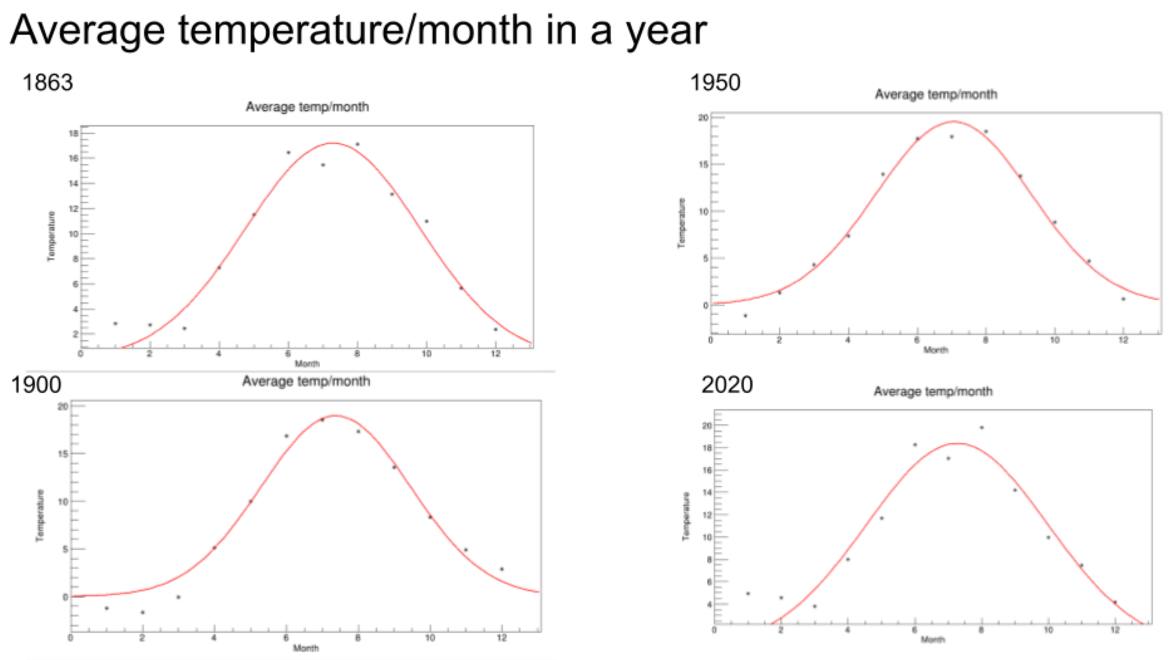


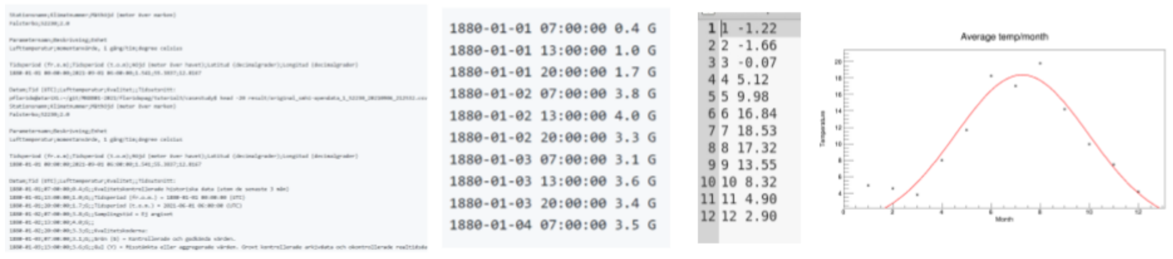
Figure 1. Graphs generated by data from SMHI, in Lund. Each point in a graph represents the average temperature that month. The red line represents the gaussian of that year.

The graphs show how much the temperature differs each month and which month is the coldest and warmest. We could also see that in different years the hottest and coldest month differs.

How it works

Program structure

- Using the bash clean up from tutorial 3
- Making my own bash program to make the final data file for ROOT
- The data in ROOT generates the graph



```
[illegible]
```

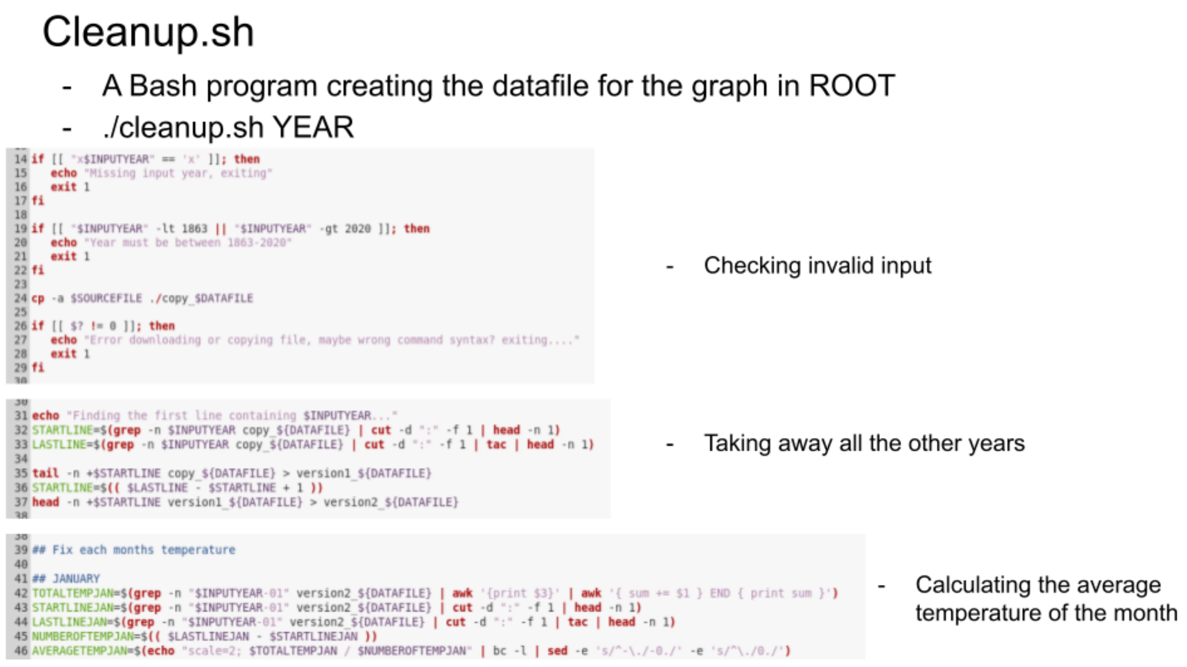
```
1880-01-01 07:00:00 0.4 G
1880-01-01 13:00:00 1.0 G
1880-01-01 20:00:00 1.7 G
1880-01-02 07:00:00 3.8 G
1880-01-02 13:00:00 4.0 G
1880-01-02 20:00:00 3.3 G
1880-01-03 07:00:00 3.1 G
1880-01-03 13:00:00 3.6 G
1880-01-03 20:00:00 3.4 G
1880-01-04 07:00:00 3.5 G
```

```
1 -1.22
2 -1.66
3 -0.07
4 5.12
5 9.98
6 16.84
7 18.53
8 17.32
9 13.55
10 8.32
11 4.90
12 2.90
```

Figure 2. Shows the output in each stage of the program from left to right (figure 2.1-2.4).

The program starts by using the file from tutorial 3 to clean up the data (figure 2.1) and takes away irrelevant data (eg. explanation of data and the data layout) and generates a file with a new data structure (figure 2.2). The output file then goes into the next part of the program where the user types in the year (between 1863-2020) as an argument that it wants to generate a graph for. That will generate an output file (figure 2.3) with the calculated average temperature each month and that will be used as the data for the graph. In the root program the user inserts the data file to generate the graph (figure 2.4).

Important functions



Cleanup.sh

- A Bash program creating the datafile for the graph in ROOT
- ./cleanup.sh YEAR

```bash
if [[ "x$INPUTYEAR" == 'x' ]]; then
   echo "Missing input year, exiting"
   exit 1
fi

if [[ "$INPUTYEAR" -lt 1863 || "$INPUTYEAR" -gt 2020 ]]; then
   echo "Year must be between 1863-2020"
   exit 1
fi

cp -a $SOURCEFILE ./copy_$DATAFILE

if [[ $? != 0 ]]; then
   echo "Error downloading or copying file, maybe wrong command syntax? exiting...."
   exit 1
fi
```

- Checking invalid input

```bash
echo "Finding the first line containing $INPUTYEAR..."
STARTLINE=$(grep -n $INPUTYEAR copy_${DATAFILE} | cut -d ":" -f 1 | head -n 1)
LASTLINE=$(grep -n $INPUTYEAR copy_${DATAFILE} | cut -d ":" -f 1 | tac | head -n 1)

tail -n +$STARTLINE copy_${DATAFILE} > version1_${DATAFILE}
STARTLINE=$(( $LASTLINE - $STARTLINE + 1 ))
head -n +$STARTLINE version1_${DATAFILE} > version2_${DATAFILE}
```

- Taking away all the other years

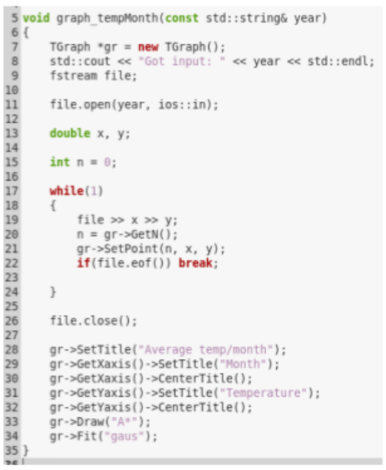
```bash
## Fix each months temperature

## JANUARY
TOTALTEMPJAN=$(grep -n "$INPUTYEAR-01" version2_${DATAFILE} | awk '{print $3}' | awk '{ sum += $1 } END { print sum }')
STARTLINEJAN=$(grep -n "$INPUTYEAR-01" version2_${DATAFILE} | cut -d ":" -f 1 | head -n 1)
LASTLINEJAN=$(grep -n "$INPUTYEAR-01" version2_${DATAFILE} | cut -d ":" -f 1 | tac | head -n 1)
NUMBEROFTEMPJAN=$(( $LASTLINEJAN - $STARTLINEJAN ))
AVERAGETEMPJAN=$(echo "scale=2; $TOTALTEMPJAN / $NUMBEROFTEMPJAN" | bc -l | sed -e 's/^-\./-0./' -e 's/^\./0./')
```

- Calculating the average temperature of the month

Figure 3. Some functions in the cleanup file. On github the file is named cleanup2.

The cleanup program starts checking for invalid inputs, it can’t be empty and it needs to be between the years 1863-2020. The input year will then be used to take away the data from the other years. To do this it searches for the first line with the year of the data that you want. Then it finds the last line of the year by reversing the data file. By using the first line of the year it takes away all the data above that year. To take away the data beneath the year it uses the newly generated file and calculates the difference between the first and last line in the year and adds one. The lines below that number will be erased. The last part of the program calculates the average temperature of each month. To find the data for the month it does the same procedure as it did to find the data for the year. In the end it adds all the temperatures of the month together and divides it by the total number of temperatures taken that month.

ROOT



```
void graph_tempMonth(const std::string& year)
{
    TGraph *gr = new TGraph();
    std::cout << "Got input: " << year << std::endl;
    fstream file;

    file.open(year, ios::in);

    double x, y;

    int n = 0;

    while(1)
    {
        file >> x >> y;
        n = gr->GetN();
        gr->SetPoint(n, x, y);
        if(file.eof()) break;

    }

    file.close();

    gr->SetTitle("Average temp/month");
    gr->GetXaxis()->SetTitle("Month");
    gr->GetXaxis()->CenterTitle();
    gr->GetYaxis()->SetTitle("Temperature");
    gr->GetYaxis()->CenterTitle();
    gr->Draw("A*");
    gr->Fit("gaus");
}
```

- Takes the filename as input

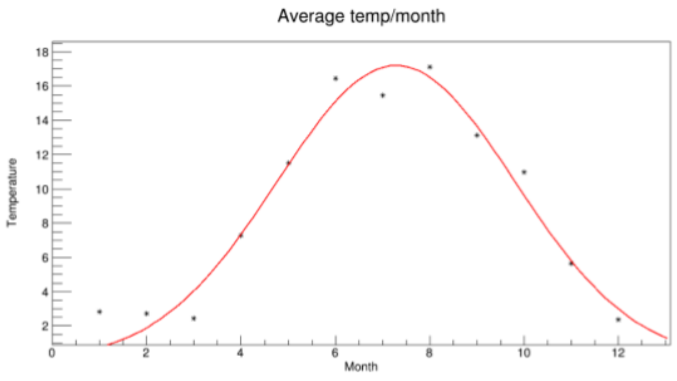


Figure 4. The ROOT program used to generate the graph.

I created a ROOT program that uses the finished data file (figure 2.4) from the cleanup program. Each point in the graph represents the month (x-value) and the average of the temperature in that month (y-value).

Shortcomings

It would be more correct to calculate the average temperature of the date first and then calculate the average of the month. The bash code in cleanup is very repetitive and could be way nicer with a loop for each month, but what would be even better is to do the calculations in cpp. A nice development of the program would be to draw a graph that is over a longer period of time, for example a decade or more.