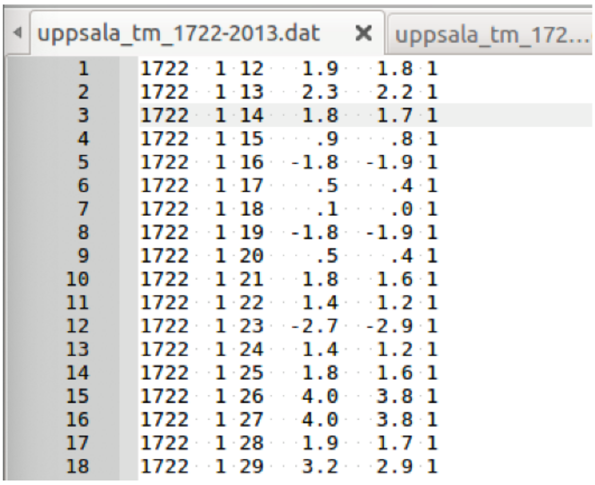
Report for final project, MNXB01

by Estelle Sernoux

November 9, 2019

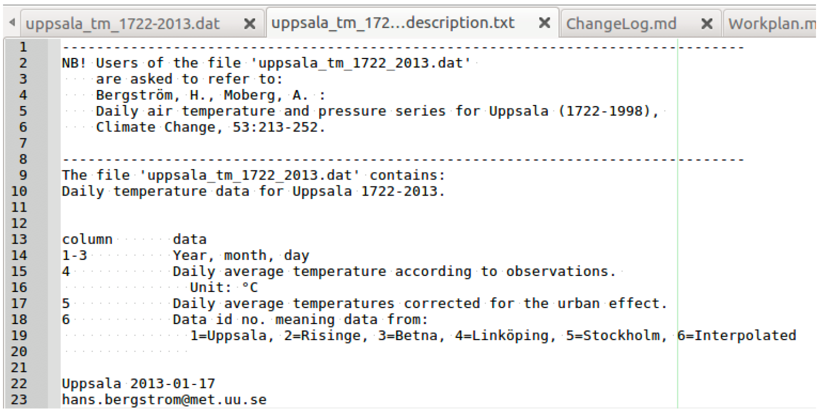
The goal of the project was to analyse data from the Swedish Meteorological and Hydrological Institute (SMHI). The data consisted of air temperature measurements from Uppsala in Sweden. The file format containing the data was .dat. And another file containing the description on datasets is in .txt. Figure 1 show a screenshot of the file associated with Uppsala temperature and figure 2 show a screenshot of the descritpion of the file assodiated with Uppsala data.



```
1722  1 12   1.9   1.8 1
1722  1 13   2.3   2.2 1
1722  1 14   1.8   1.7 1
1722  1 15    .9    .8 1
1722  1 16  -1.8  -1.9 1
1722  1 17    .5    .4 1
1722  1 18    .1    .0 1
1722  1 19  -1.8  -1.9 1
1722  1 20    .5    .4 1
1722  1 21   1.8   1.6 1
1722  1 22   1.4   1.2 1
1722  1 23  -2.7  -2.9 1
1722  1 24   1.4   1.2 1
1722  1 25   1.8   1.6 1
1722  1 26   4.0   3.8 1
1722  1 27   4.0   3.8 1
1722  1 28   1.9   1.7 1
1722  1 29   3.2   2.9 1
```

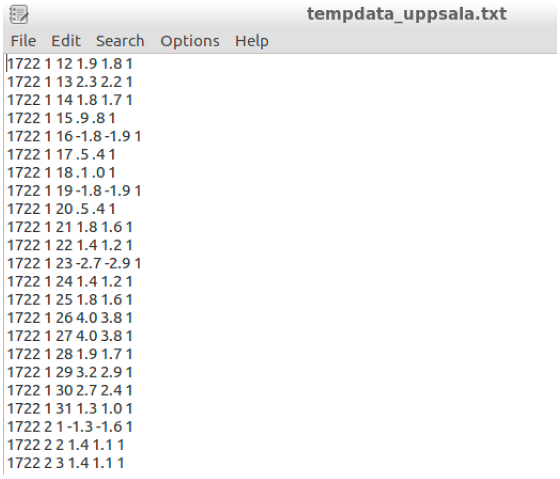
Figure 1: The raw data in Uppsala-tm-1722-2013.dat

First, I clean the file by writing a temporary code which when it's executed, replace the irregular space groups with a single space separating the data. This script would extract the data and save it to a new data file called tempdata-uppsala.txt. The result is shown in Figure 3.



```
---------------------------------------------------------------------------------------
NB! Users of the file 'uppsala_tm_1722_2013.dat'
    are asked to refer to:
    Bergström, H., Moberg, A. :
    Daily air temperature and pressure series for Uppsala (1722-1998),
    Climate Change, 53:213-252.

---------------------------------------------------------------------------------------
The file 'uppsala_tm_1722_2013.dat' contains:
Daily temperature data for Uppsala 1722-2013.


column      data
1-3         Year, month, day
4           Daily average temperature according to observations.
              Unit: °C
5           Daily average temperatures corrected for the urban effect.
6           Data id no. meaning data from:
              1=Uppsala, 2=Risinge, 3=Betna, 4=Linköping, 5=Stockholm, 6=Interpolated


Uppsala 2013-01-17
hans.bergstrom@met.uu.se
```

Figure 2: Desciption of Uppsala data file

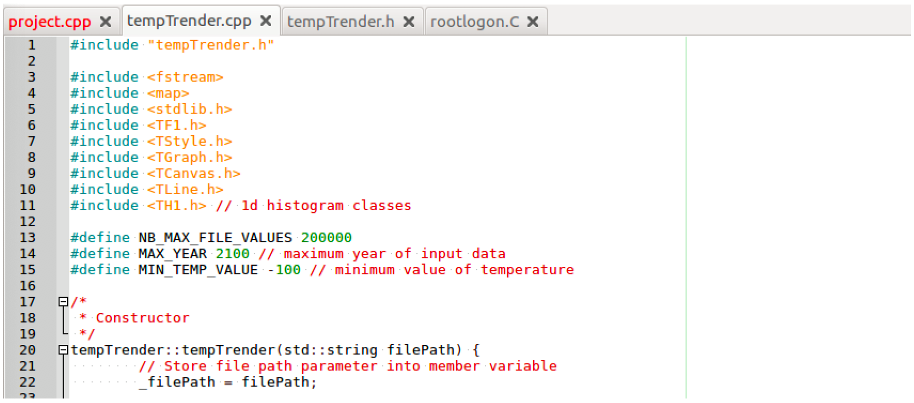


```
1722 1 12 1.9 1.8 1
1722 1 13 2.3 2.2 1
1722 1 14 1.8 1.7 1
1722 1 15 .9 .8 1
1722 1 16 -1.8 -1.9 1
1722 1 17 .5 .4 1
1722 1 18 .1 .0 1
1722 1 19 -1.8 -1.9 1
1722 1 20 .5 .4 1
1722 1 21 1.8 1.6 1
1722 1 22 1.4 1.2 1
1722 1 23 -2.7 -2.9 1
1722 1 24 1.4 1.2 1
1722 1 25 1.8 1.6 1
1722 1 26 4.0 3.8 1
1722 1 27 4.0 3.8 1
1722 1 28 1.9 1.7 1
1722 1 29 3.2 2.9 1
1722 1 30 2.7 2.4 1
1722 1 31 1.3 1.0 1
1722 2 1 -1.3 -1.6 1
1722 2 2 1.4 1.1 1
1722 2 3 1.4 1.1 1
```

Figure 3: New file, tempdata-uppsala.txt

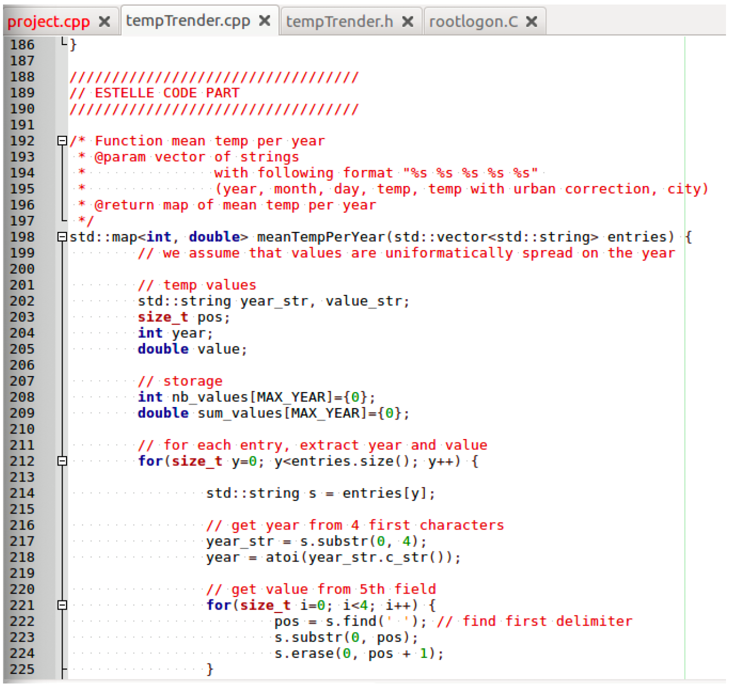
I will use the year of the file (first column in figure 3) and daily average temperatures (corrected by the urban effect, penultimate column), in order to show a histogram plotting average values of temperatures in Uppsala each year over a period from 1722 to 2013.

I create a function that allow us to read this file and retrieve all the temperatures per year in order to calculate the mean temperature per year. So I extract the first 4 characters which correspond to the year, then the 5th field which corresponds to the temperature. I sum the values as seen below figure 4 and 5.



```cpp
#include "tempTrender.h"

#include <fstream>
#include <map>
#include <stdlib.h>
#include <TF1.h>
#include <TStyle.h>
#include <TGraph.h>
#include <TCanvas.h>
#include <TLine.h>
#include <TH1.h> // 1d histogram classes

#define NB_MAX_FILE_VALUES 200000
#define MAX_YEAR 2100 // maximum year of input data
#define MIN_TEMP_VALUE -100 // minimum value of temperature

/*
 * Constructor
 */
tempTrender::tempTrender(std::string filePath) {
        // Store file path parameter into member variable
        _filePath = filePath;
```

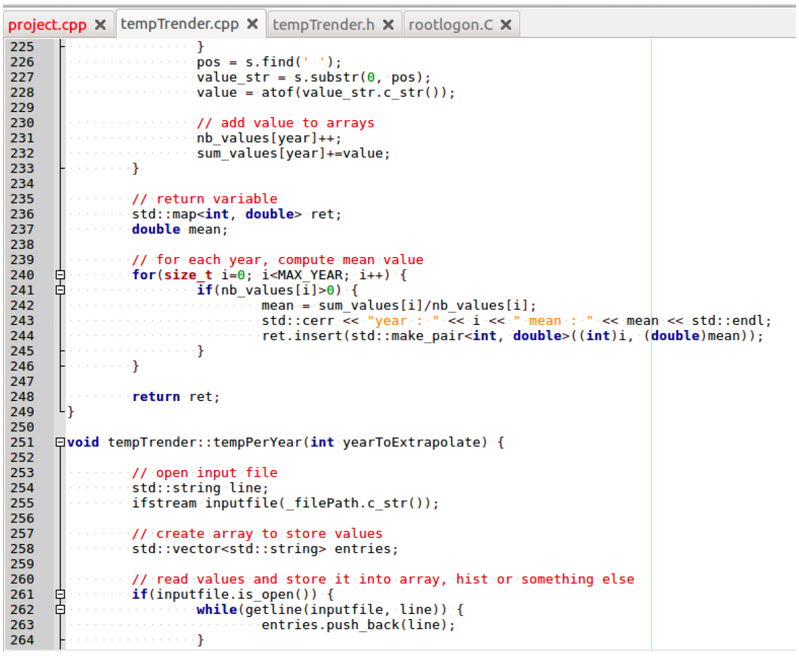
Figure 4: Extract (a) from tempTrender.cpp



```cpp
}

//////////////////////////////////
// ESTELLE CODE PART
//////////////////////////////////

/* Function mean temp per year
 * @param vector of strings
 *               with following format "%s %s %s %s %s"
 *               (year, month, day, temp, temp with urban correction, city)
 * @return map of mean temp per year
 */
std::map<int, double> meanTempPerYear(std::vector<std::string> entries) {
        // we assume that values are uniformatically spread on the year

        // temp values
        std::string year_str, value_str;
        size_t pos;
        int year;
        double value;

        // storage
        int nb_values[MAX_YEAR]={0};
        double sum_values[MAX_YEAR]={0};

        // for each entry, extract year and value
        for(size_t y=0; y<entries.size(); y++) {

                std::string s = entries[y];

                // get year from 4 first characters
                year_str = s.substr(0, 4);
                year = atoi(year_str.c_str());

                // get value from 5th field
                for(size_t i=0; i<4; i++) {
                        pos = s.find(' '); // find first delimiter
                        s.substr(0, pos);
                        s.erase(0, pos + 1);
                }
```

Figure 5: Extract (b) from tempTrender.cpp

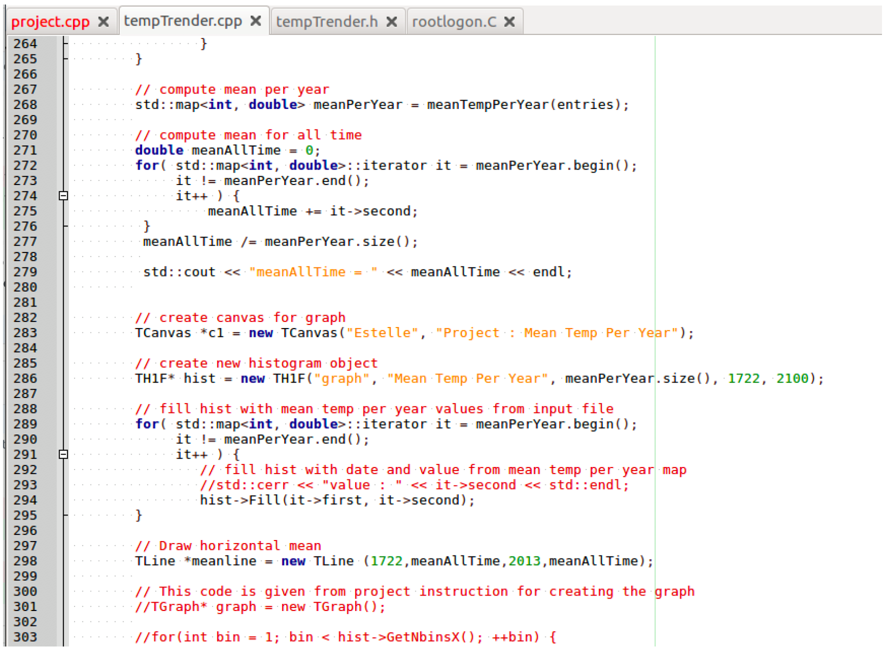
Then I calculate the mean temperature per year and the mean temperature of all the years. And finally I create a canvas object and draw the histogram, as you can see in extracts (d) and (e).



```cpp
        }
        pos = s.find(' ');
        value_str = s.substr(0, pos);
        value = atof(value_str.c_str());

        // add value to arrays
        nb_values[year]++;
        sum_values[year]+=value;
    }

    // return variable
    std::map<int, double> ret;
    double mean;

    // for each year, compute mean value
    for(size_t i=0; i<MAX_YEAR; i++) {
        if(nb_values[i]>0) {
            mean = sum_values[i]/nb_values[i];
            std::cerr << "year : " << i << " mean : " << mean << std::endl;
            ret.insert(std::make_pair<int, double>((int)i, (double)mean));
        }
    }

    return ret;
}

void tempTrender::tempPerYear(int yearToExtrapolate) {

    // open input file
    std::string line;
    ifstream inputfile(_filePath.c_str());

    // create array to store values
    std::vector<std::string> entries;

    // read values and store it into array, hist or something else
    if(inputfile.is_open()) {
        while(getline(inputfile, line)) {
            entries.push_back(line);
        }
```

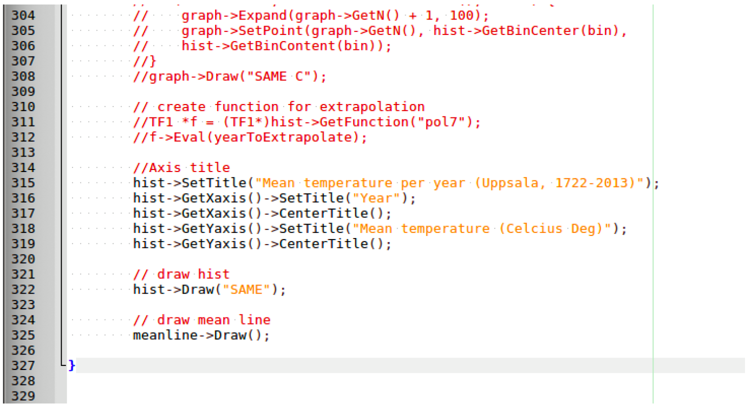
Figure 6: Extract (c) from tempTrender.cpp

After compilation, we are able to plot a histogram of mean temperature each year and the mean of all years, over a period from 1722 to 2013.



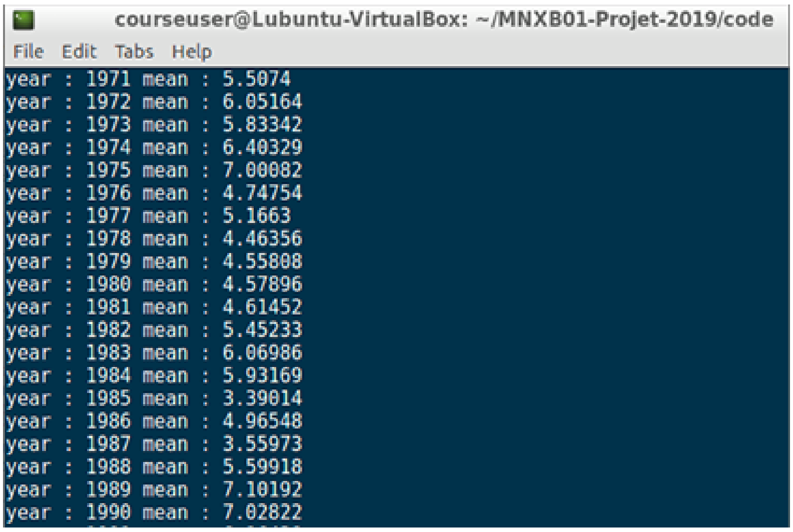
```
                }
        }

        // compute mean per year
        std::map<int, double> meanPerYear = meanTempPerYear(entries);

        // compute mean for all time
        double meanAllTime = 0;
        for( std::map<int, double>::iterator it = meanPerYear.begin();
             it != meanPerYear.end();
             it++ ) {
                 meanAllTime += it->second;
         }
         meanAllTime /= meanPerYear.size();

         std::cout << "meanAllTime = " << meanAllTime << endl;


        // create canvas for graph
        TCanvas *c1 = new TCanvas("Estelle", "Project : Mean Temp Per Year");

        // create new histogram object
        TH1F* hist = new TH1F("graph", "Mean Temp Per Year", meanPerYear.size(), 1722, 2100);

        // fill hist with mean temp per year values from input file
        for( std::map<int, double>::iterator it = meanPerYear.begin();
             it != meanPerYear.end();
             it++ ) {
                // fill hist with date and value from mean temp per year map
                //std::cerr << "value : " << it->second << std::endl;
                hist->Fill(it->first, it->second);
        }

        // Draw horizontal mean
        TLine *meanline = new TLine (1722,meanAllTime,2013,meanAllTime);

        // This code is given from project instruction for creating the graph
        //TGraph* graph = new TGraph();

        //for(int bin = 1; bin < hist->GetNbinsX(); ++bin) {
```

Figure 7: Extract (d) from tempTrender.cpp



```
        //    graph->Expand(graph->GetN() + 1, 100);
        //    graph->SetPoint(graph->GetN(), hist->GetBinCenter(bin),
        //    hist->GetBinContent(bin));
        //}
        //graph->Draw("SAME C");

        // create function for extrapolation
        //TF1 *f = (TF1*)hist->GetFunction("pol7");
        //f->Eval(yearToExtrapolate);

        //Axis title
        hist->SetTitle("Mean temperature per year (Uppsala, 1722-2013)");
        hist->GetXaxis()->SetTitle("Year");
        hist->GetXaxis()->CenterTitle();
        hist->GetYaxis()->SetTitle("Mean temperature (Celcius Deg)");
        hist->GetYaxis()->CenterTitle();

        // draw hist
        hist->Draw("SAME");

        // draw mean line
        meanline->Draw();

}
```

Figure 8: Extract (e) from tempTrender.cpp



```
year : 1971 mean : 5.5074
year : 1972 mean : 6.05164
year : 1973 mean : 5.83342
year : 1974 mean : 6.40329
year : 1975 mean : 7.00082
year : 1976 mean : 4.74754
year : 1977 mean : 5.1663
year : 1978 mean : 4.46356
year : 1979 mean : 4.55808
year : 1980 mean : 4.57896
year : 1981 mean : 4.61452
year : 1982 mean : 5.45233
year : 1983 mean : 6.06986
year : 1984 mean : 5.93169
year : 1985 mean : 3.39014
year : 1986 mean : 4.96548
year : 1987 mean : 3.55973
year : 1988 mean : 5.59918
year : 1989 mean : 7.10192
year : 1990 mean : 7.02822
```

Figure 9: Extract from terminal

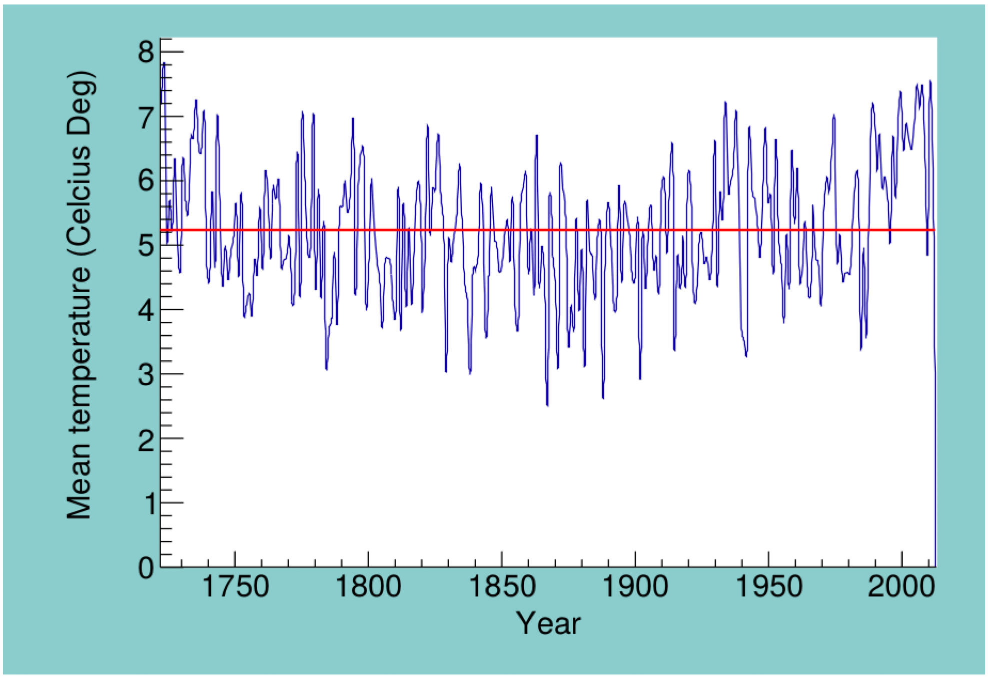


Figure 10: The graph shows the mean temperature of each year since 1722 in blue, and the mean of all years in red color