# Data Insights and Design

As a team, we considered several data sources; we looked first at climate change data, then at criminal activity data, but both were discarded for incomplete and limited data pools. The data we settled on as a team, we chose both because of the large data pool (over 13,000 records) that was all complete. It allowed for meaningful comparison between the three areas it covered; however, we chose to trim down the data to around 1,300 records, as it made the data far less unwieldy to manipulate. Originally, the data was global, including data from sources all over the world (see fig. 1 below). We decided to focus on simply the data from and relating to the UK.

Diagram

Description automatically generated

*Fig. 1: Original data overview.*

The data we have chosen is weather data from the years 1900-2013, made up of average temperature measurements from the first of the month throughout each of the 113 years. It covers both land and ocean temperatures, with twelve records for each year in this period. Originally, the data was in three tables: Global Temperature, Country Land Temperature, and City Temperature (see fig. 2 below).

*Diagram

Description automatically generatedFig. 2: an entity relationship diagram for the original three tables, showing the fields, relationships, and linking keys for each.*

We decided to combine all of these into one table for ease of access and data cleanliness. When first analysing the data for eligibility, we split the data into equal chunks of 18 years, which made it easier to analyse. As such, we decided to take a similar approach when drawing up our list of requirements for the software design.

It was our intention that the user should be able to utilise the historical data throughout the period to be able to identify changes between and patterns in the temperatures, by allowing the user to search and filter different seasons and months for years. We needed the data to easy to manipulate, reflecting this in our chosen design: the data will be explored by the user mixing and matching various filters, including month, season, and year. The landing page will have options for all three, though within each the user will be able to add the other filters should they wish to do so. The end result will be a dynamic graph, either a pie chart or a bar chart depending on which is most appropriate, that changes with the data the user chooses.

# Software Requirements

Our requirements for the software:

* Allow the user to filter data by:
  + Month
  + Year
  + Season
* Produce graphs that changes with the data the user filters
  + Pie charts
  + Bar charts
* Be intuitive and easy to use
* Read from the .CSV file to obtain the data needed
* Allow the user to draw conclusions as to temperature patterns in the data
* EXT: write to a .txt file to produce a downloadable file of filtered data.

[more software requirements once we have decided upon a structure for our code]

# Algorithm Justification

We have decided to utilise the Matplotlib library to create our graphs, as it would allow us to do so dynamically by simply changing the values of the variables assigned, rather than needing to recode the graph each time a new filter is added. It is our hope that by making use of dynamic graphs, the user can easier draw conclusions from our data, thereby achieving point 5 in our software requirements while achieving point 2.

We have also decided to utilise semi-identical functions for each filter; as we intend to import the data into a 2d array, functions would streamline the code, allowing for a simpler and more efficient program. Each search function – one for seasons, one for months, and one for year – would be almost identical, simply swapping out variables as needed.

[more algorithm discussion once we have decided upon a structure for our code, include pictures of pseudocode/flowcharts]

# Description of implementation

[insert discussion of implemented features, how they match up to our software requirements, and our design. Include screenshots of code, flowcharts, the works.]

# Testing

[insert discussion of testing, including testing tables, test descriptions, results, and actions taken. Include screenshots of test code, the works.]

# Evaluation

[insert evaluation here.]