Contents

Foreword

Democracy Needs Statistical Literacy
Gerd Gigerenzer

1 Why Engage with Civic Statistics?
Jim Ridgway

Part I

Edited by Joachim Engel and James Nicholson

2 Back to the Future – Rethinking the Purpose and Nature of Statistics Education
Joachim Engel and Jim Ridgway

3 A Conceptual Framework for Civic Statistics and its Educational Applications
Iddo Gal, James Nicholson, and Jim Ridgway

4 Implementing Civic Statistics – An Agenda for Action
Iddo Gal, Jim Ridgway, James Nicholson, and Joachim Engel

Part II

Edited by Rolf Biehler and Peter Kovacs

5 Interactive Data Visualizations for Teaching Civic Statistics
Jim Ridgway, Pedro Campos, James Nicholson, and Sónia Teixeira

6 Data Sets: Examples and Access for Civic Statistics
Sónia Teixeira, Pedro Campos, and Anna Trostianitser

7 Lesson Plan Approaches: Tasks that Motivate Students to Think
Anna Trostianitser, Sónia Teixeira, and Pedro Campos
8 Seeing Dynamic Data Visualizations in Action: Gapminder Tools
   Peter Kovacs, Klara Kazar, and Eva Kuruczleki

9 Data Visualization Packages for Non-inferential Civic Statistics in High School Classrooms
   Daniel Frischemeier, Susanne Podworny, and Rolf Biehler

10 Civic Statistics and iNZight: Illustrations of Some Design Principles for Educational Software
    Chris J. Wild and Jim Ridgway

11 Exploring Climate Change Data with R
    Nuno Guimarães, Kimmo Vehkalahti, Pedro Campos, and Joachim Engel

12 Covid-19 Shows Why we Need Civic Statistics: Illustrations and Classroom Activities
    Jim Ridgway and Rosie Ridgway

Part III

Edited by Iddo Gal and Daniel Frischemeier

13 Critical Understanding of Civic Statistics: Engaging with Important Contexts, Texts, and Opinion Questions
   Iddo Gal

14 Implementing Civic Statistics in Business Education: Technology in Small and Large Classrooms
   Peter Kovacs, Klara Kazar, and Eva Kuruczleki

15 Civic Statistics for Prospective Teachers: Developing Content and Pedagogical Content Knowledge through Project Work
   Susanne Podworny, Daniel Frischemeier, and Rolf Biehler

16 Civic Statistics for Prospective Teachers: Developing Critical Questioning of Data-based Statements in the Media
   Achim Schiller and Joachim Engel

17 Civic Statistics at School: Reasoning with Real Data in the Classroom
Christoph Wassner and Andreas Proemmel

18 Preparing for a Data-rich world: Civic Statistics Across the Curriculum
Joachim Engel, Josephine Louie, and James Nicholson

19 Dynamic, Interactive Trees and Icon Arrays for Visualizing Risks in Civic Statistics
Laura Martignon, Daniel Frischemeier, Michelle McDowell, and Christoph Till

Part IV

Edited by Pedro Campos and Achim Schiller

20 Reflections on Civic Statistics — A Triangulation of Citizen, State and Statistics: Past, Present and Future
Karen François and Carlos Monteiro

21 Connecting Data Science, Data Movements, and Project-based Learning with a Social Impact
Leid Zejnilović and Pedro Campos

Jim Ridgway, Pedro Campos and Rolf Biehler

23 Civic Statistics in Context: Mapping the Global Evidence Ecosystem
Jim Ridgway and Rosie Ridgway
Chapter 11
Exploring Climate Change Data with R

Nuno Guimarães, CRACS-INESCTEC and University of Porto, Portugal
nuno.r.guimaraes@inesctec.pt
Kimmo Vehkalahti, University of Helsinki, Finland
kimmo.vehkalahti@helsinki.fi
Pedro Campos, LIADD-INESCTEC and University of Porto, Portugal
pcampos@fep.up.pt
Joachim Engel, Ludwigsburg University of Education, Germany
engel@ph-ludwigsburg.de

Abstract

Climate change is an existential threat facing humanity and the future of our planet. The signs of global warming are everywhere, and they are more complex than just the climbing temperatures. Climate data on a massive scale has been collected by various scientific groups around the globe. Exploring and extracting useful knowledge from large quantities of data requires powerful software. In this chapter we present some possibilities for exploring and visualising climate change data in connection with statistics education using the freely accessible statistical programming language R together with the computing environment RStudio. In addition to the visualisations, we provide annotated references to climate data repositories and extracts of our openly published R scripts for encouraging teachers and students to reproduce and enhance the visualisations.

Keywords: R; Coding; Climate Data; Data Visualisation; Multivariate Data

11.1 Introduction

We live in a world awash with data and freely available software. Skills to use information technology, search for information, and explore data are of increasing importance. Digital literacy is a central goal for preparing students for the digital age and providing them with 21st century skills (OECD, 2019). Chapter 9 of this book presents various data visualisation packages designed for learning statistics and data analysis, suitable for high school. Chapter 10 discusses design principles for educational software and introduces the data analysis tool iNZight for visualizing Civic Statistics data. iNZight is based on a graphical user interface (GUI), allowing its use without any knowledge of programming code – a great advantage, especially for learners, because it avoids the additional cognitive load of handling a programming language. However, GUI-based systems do not have the same level of functionality and granular control as command line interface systems (CLI) which provide greater flexibility and can be used to easily do things that are difficult or even impossible to do with a GUI.
For some burning problems facing the world today, such as global warming, the analysis involves massive complex datasets, and hence requires flexible use of professional software. Therefore, in this chapter we go beyond GUI and illustrate how students with increasing digital skills can be encouraged to explore and visualise complex data on climate change using the command-based professional software R. Although visualisation is a great place to start with R programming, data visualisation by itself is typically not enough (Wickham and Grolemund, 2017). Data transformation and modelling is an important part of the exploratory process, since users get involved with selecting important variables, filtering out key observations, creating new variables, and computing summaries. This is a clear advantage of using R for data analysis.

We provide annotated references to climate data repositories and extracts of our openly published programming scripts for encouraging teachers and students to reproduce and enhance the visualisations. The challenge to provide students with computational tools and to enhance their data-related capacities has been emphasized by various statistics educators (e.g., Horton et al., 2014; Nolan & Temple Lang, 2010).

The purpose of this chapter is to provide an accessible entry to exploring climate change data for learners who have little previous experience with professional environments meant for statistical modelling and programming. This includes addressing some technical issues concerning coding and handling of software. Annotated references to climate data repositories and extracts of openly published programming code are provided. The reader can replicate all the visualisations presented here and may create additional representations that may look at the data from different perspectives.

Some burning problems facing the world today such as global warming require the analysis of massive complex datasets and the flexible use of professional software

In the recent Future of Humanity survey of over 10,000 young people in 22 countries aged 18-25 years (also called Generation Z), climate change was the most commonly cited issue facing the world¹. From Prince Charles² and Pope Francis³ to Antonio Guterres⁴, and many other prominent figures or world leaders, there is a consensus that climate change is the

³ https://catholicclimatecovenant.org/encyclical
biggest threat to this planet. Indeed, scientists have warned for decades of the global threat that global warming and climate changes pose to planet Earth. Effects on the environment, such as loss of sea ice, accelerated sea level rise and longer and more intense heat waves are already being observed. Extreme weather events are observed in higher frequency. In addition, glaciers have shrunk, ice on rivers and lakes is breaking up earlier, plant and animal ranges have shifted, and trees are flowering sooner.

For many people, however, climate change still seems an abstract and faraway phenomenon. One way of raising students’ awareness and engagement in the climate change topic is to let them explore and analyse climate change datasets by themselves, with support from their teachers. We approach this option by illustrating possibilities of exploring and visualising openly available climate data, by working with the powerful statistical programming language R (R Core Team, 2020) using the freely accessible computing environment RStudio.

To raise students’ awareness and engagement about climate change we encourage them to explore and analyse climate change datasets by themselves.

Einstein is reported to have said that Everything should be made as simple as possible, but no simpler. We believe in learning by doing and following ready-made examples, but there are also challenges to our approach. As weather changes rapidly in time and with local fluctuations while climate comprises long-range patterns, the study of climate requires the exploration of complex spatial time series data. Climate change data not only have a massive scale but also high dimensionality and complicated dependency structures, making the analysis task challenging. However, using a script-based programming language, such as R, offers certain advantages: users can learn the functions and code fragments step-by-step, and the whole analysis is under full user control with all the results easily reproduced. The students can play around with the data, ask new questions and modify the code to get new representations. Another advantage of using the provided R script is the fact that by running the code, the freshest data sets are retrieved (e.g., most recent data on CO₂ concentration in the atmosphere) and made available for the analysis. Our aim is to encourage both teachers and students in using a hands-on approach that assumes no specific prior knowledge of R (Shah, 2020).

Why R? During its early years (around year 2000), R was mainly used in the (Mathematical) Statistics community (where it originated in the early 1990s), but nowadays R has become
one of the most popular programming languages at large. Indeed, R is the primary tool reported in data analysis (Lai et al. 2019, p.1). Theobold and Hancock (2019) conclude in their review of R and other computing possibilities in the environmental sciences: Statistical computing has become a foundational aspect of research. In many other fields there is similar evidence of the fact that R has begun to take over the role from traditional commercial software packages, such as SAS or SPSS.

An introduction to R - even for the specific area of climate change - is beyond the scope of this chapter - huge number of excellent introductory materials (e.g., slides, books, websites, blogs, and interactive courses) are freely available. It is generally wise to consult quite recent materials, because R is under constant development and it has gone through several major changes and updates during the 2010s. One example of a modern textbook is R for Data Science (Wickham & Grolemund, 2017) that helps users to learn the most important data science tools in R: importing, cleaning, transforming, and visualising data. The book is fully accessible online.6 Another example of a useful resource is an online textbook on Open Data Science that covers the basics of R and RStudio as well as the state-of-the-art tools for reproducibility and version control, namely, Git and GitHub (see also Lowndes et al., 2017). We may also mention DataCamp7 (a commercial company) that offers a popular, free interactive course Introduction to R. Since 2016, almost two million people worldwide have taken the course.

In our examples, we focus on illustrating the potential of R for visualising climate change data. For further details, we refer the interested reader to Shen (2017 and Shen & Somerville (2019). See also the hands-on YouTube videos related to Shen and Somerville’s (2019) book8.

The structure of the chapter is as follows. Section 11.2 provides guidance to trustworthy data repositories with visualisations to retrieve up-to-date climate data and sources for deeper background information. Section 11.3 illustrates selected visualisations of climate data supported with technical hints and activities related to the R language. Section 11.4 presents some ideas and suggestions of how to alleviate the technical challenges in implementing the R code for learners, and section 11.5 concludes.

11.2 Sources for Climate Data and Visualisations

There are many comprehensive sources for climate data and their visualisation, collected and compiled by top research institutions, international organisations and government agencies. In this section, we provide a brief guidance to some of these sources with a summary provided in Table 11.1.

The “Five Most Important” Datasets of Climate Science

---

5 https://r4ds.had.co.nz/  
6 https://ohi-science.org/data-science-training/  
7 https://www.datacamp.com  
8 https://www.youtube.com/channel/UC92gEJfTpsVcE92fCeSRbJg/videos
These datasets are based on the presentation made by Stefan Rahmstorf to the Arctic Expedition for Climate Action⁹. They include the Vostok Ice Core (Antarctica), Rise of CO₂ (“Keeling curve”), the Global Average Temperature, Sea Level Rise and Sea Ice Retreat. Despite the potential subjectivity of the selection, those datasets represent undeniably important climate science phenomena, such as temperature, atmospheric carbon dioxide (CO₂), sea level, and sea ice. These datasets are explored using R in the next section.

weatherData R package

The weatherData R package¹⁰ is a library of functions that helps in fetching weather data from websites. Given a location and a date range, these functions help fetch weather data (temperature, pressure, humidity, wind speed etc.) for any weather related analysis. Examples of the different types of data are: underground stations data, data of US weather stations, daily minimum (maximum) temperatures for a given weather station, and data from international weather stations.

Copernicus Climate Change Service

The Copernicus Climate Change Service (C3S), powered by Copernicus, the European Union's Earth Observation Programme, provides information about the past, present and future climate, as well as tools to enable modelling climate change mitigation and adaptation strategies by policy makers and businesses¹¹. Here, it is possible to find extensive information (including didactic material) and illustrations in the form of explanations, tables, graphics, animations and videos. The Climate Data Store delivers many datasets, ranging from monthly data on pressure levels to data on glacier elevation and mass change data from 1850 to the present from the Fluctuations of Glaciers Database. It is freely available and functions as a one-stop shop for exploring climate data.

NASA Global Climate Change and GISS Data

NASA offers many different resources, such as educational videos, charts, tables and many explanations about climate change¹². In particular, the Goddard Institute for Space Studies (GISS) provides some datasets and images¹³ such as the one from the ISCCP (International Satellite Cloud Climatology Project) with 3-hourly weather state data at 1-degree horizontal resolution, covering the period from July 1983 to June 2015.

Deutscher Wetterdienst

Here you can find the CDC (Climate Data Center)¹⁴ and explore “bad weather days”, i.e. days when the weather makes work on construction sites difficult or impossible. Another

---

¹⁰ https://ram-n.github.io/weatherData/
¹¹ https://climate.copernicus.eu
¹² https://climate.nasa.gov/
¹³ https://data.giss.nasa.gov/
¹⁴ https://cdc.dwd.de/portal/
new product is the wind climatology “QuWind100”, which leads to a comprehensive improvement in the accuracy of the yield estimation of wind turbines in Germany.

**National Geographic**

“Seven things to know about climate change”\(^{15}\) is a climate hub created by National Geographic to let us know about diverse changes occurring in climate related topics, such as global warming, the shrinking of ice in the Arctic Sea and the vanishing of some animals and plants.

**International Research Institute (IRI) for Climate & Society Data Library**

The IRI Data Library\(^{16}\) is a powerful and freely accessible online data repository and analysis tool that allows users to view, analyse, and download hundreds of terabytes of climate-related data through a standard web browser. Datasets are available which offer the opportunity to create analyses of data, ranging from simple averaging to more advanced analyses using the Ingrid Data Analysis Language. In addition, there is a climate and society map room, a collection of maps and other figures that monitor climate and societal conditions at present and in the recent past. The maps and figures can be manipulated and are linked to the original data. Examples of datasets include Climate and Agriculture Data, Fires, El niño, Food Security, etc.

**WorldClim Database**

The WorldClim Database ([www.worldclim.org](http://www.worldclim.org)) provides current, future, and past data in a 1km x 1km grid format (and other resolutions). Data is formatted to be easily accessible and adapted to be used in *ArcGIS* and other standard formats. It contains maps, graphs, tables, and data on the global climate such as the historical monthly weather data, global climate and weather data, bioclimatic variables, future climate data, Paleo-climate, etc.

**RClimate**

*RClimate*\(^{17}\) provides a set of problems to be solved with R related to the analysis of flash floods based on the United States Geological Survey (USGS). The National Water Information System provides data on the Schuylkill River at Philadelphia\(^{18}\) regarding several variables, such as water temperature, discharge (in cubic feet per second), suspended sediment concentration, suspended sediment discharge, etc. D. Kelly O’Day, the author of this blog, provides several charts and data that allows the use of these data and R to understand climate change.

Table 11.1 Summary of sources for climate change data and visualisations

<table>
<thead>
<tr>
<th>Resource name</th>
<th>Data sources</th>
<th>Data/variables available</th>
<th>link</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="https://waterdata.usgs.gov/nwis/inventory/?site_no=01474500">https://waterdata.usgs.gov/nwis/inventory/?site_no=01474500</a></td>
<td>(USGS) Schuylkill River data</td>
<td>Water temperature, discharge, suspended sediment concentration and discharge</td>
<td></td>
</tr>
<tr>
<td><a href="https://rclimate.wordpress.com/">https://rclimate.wordpress.com/</a></td>
<td>RClimate problems</td>
<td>Flash flood analysis results for the United States</td>
<td></td>
</tr>
<tr>
<td><a href="https://iridl.ldeo.columbia.edu/">https://iridl.ldeo.columbia.edu/</a></td>
<td>IRI Data Library</td>
<td>Comprehensive climate data repository and analysis tool</td>
<td></td>
</tr>
</tbody>
</table>


\(^{16}\) [http://iridl.ldeo.columbia.edu/](http://iridl.ldeo.columbia.edu/)

\(^{17}\) [https://rclimate.wordpress.com/](https://rclimate.wordpress.com/)

\(^{18}\) [https://waterdata.usgs.gov/nwis/inventory/?site_no=01474500](https://waterdata.usgs.gov/nwis/inventory/?site_no=01474500)
<table>
<thead>
<tr>
<th>The “Five Most Important” Datasets of Climate Science (by Stefan Rahmstorf)</th>
<th>Stefan Rahmstorf, based on agencies, such as NASA/GISS, ERSST, etc.</th>
<th>Vostok Ice Core (Antarctica), Rise of CO₂ (“Keeling curve”), the Global Average Temperature, Sea Level Rise and Sea Ice Retreat.</th>
<th><a href="https://tamino.wordpress.com/2018/11/01/the-5-most-important-data-sets-of-climate-science/">https://tamino.wordpress.com/2018/11/01/the-5-most-important-data-sets-of-climate-science/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>weatherData</strong> R package (a library of functions that helps in fetching weather data from websites)</td>
<td>Weather underground (<a href="https://www.wunderground.com/">https://www.wunderground.com/</a>) and others</td>
<td>Weather data (temperature, pressure, humidity, wind speed etc.) for any weather related analysis;</td>
<td><a href="https://ram-n.github.io/weatherData/">https://ram-n.github.io/weatherData/</a></td>
</tr>
<tr>
<td>Copernicus Climate Change Service (Copernicus - the European Union's Earth Observation Programme)</td>
<td>Copernicus</td>
<td>The Climate Data Store delivers many datasets, from monthly data on pressure levels, through to data of glaciers' elevation and mass change data from 1850 to present from the Fluctuations of Glaciers Database.</td>
<td><a href="https://climate.copernicus.eu">https://climate.copernicus.eu</a></td>
</tr>
<tr>
<td>NASA Global Climate Change and GISS Data</td>
<td>NASA and the Goddard Institute for Space Studies (GISS)</td>
<td>Datasets and images where you can find data from the ISCCP (International Satellite Cloud Climatology Project) with 3-hourly weather state data at 1-degree horizontal resolution, covering the period from July 1983 to June 2015.</td>
<td><a href="https://climate.nasa.gov/">https://climate.nasa.gov/</a></td>
</tr>
<tr>
<td>Deutscher Wetterdienst</td>
<td>Climate Data Center</td>
<td>Explore “bad weather days”, i.e. days when the weather makes work on construction sites difficult or impossible, and wind climatology (wind turbines in Germany).</td>
<td><a href="https://cdc.dwd.de/portfolio/">https://cdc.dwd.de/portfolio/</a></td>
</tr>
<tr>
<td>National Geographic Climate Hub</td>
<td>National Geographic</td>
<td>Many data visualisations, such as the shrinking of Arctic Sea ice, the vanishing of some animals and plants, etc.</td>
<td><a href="https://www.nationalgeographic.com/magazine/2017/04/seven-things-to-know-about-climate-change">https://www.nationalgeographic.com/magazine/2017/04/seven-things-to-know-about-climate-change</a></td>
</tr>
<tr>
<td>International Research Institute (IRI) for Climate &amp; Society Data Library</td>
<td>University of Columbia, NY, USA</td>
<td>Datasets are available that facilitate creating analyses and maps. Examples of datasets include Climate and Agriculture Data, Fires, El niño, Food Security, etc.</td>
<td><a href="http://iridl.ldeo.columbia.edu">http://iridl.ldeo.columbia.edu</a></td>
</tr>
<tr>
<td>WorldClim Database</td>
<td>WorldClim</td>
<td>Maps, graphs, tables, and data on global climate such as historical monthly weather data, global climate and weather data, bioclimatic variables, future climate data, Paleo-climate, etc.</td>
<td><a href="http://www.worldclim.org/">www.worldclim.org/</a></td>
</tr>
<tr>
<td>RClimate (a blog from D. Kelly O’Day)</td>
<td>United States Geological Survey (USGS) and others including flood data</td>
<td>Variables such as temperature, water, discharge, suspended sediment concentration, suspended sediment discharge etc.</td>
<td><a href="https://rclimate.wordpress.com/">https://rclimate.wordpress.com/</a></td>
</tr>
</tbody>
</table>

### 11.3 Using R in Exploring Climate Change Data

In this section, we present some impressive visualisations together with data and code for creating them in R. We will use the *Five Most Important Datasets for Climate Change* by Stefan Rahmstorf (see Table 11.1). Our R code for visualising the five datasets is adapted...
with modifications from a blog\textsuperscript{19} published in 2018. The graphs of the blog were created using R and its \texttt{ggplot2} graphing package\textsuperscript{20}, making the process reproducible and thus easier to share and study. In addition to those five datasets, we show how to generate interactive geographical maps with R to illustrate rising temperatures in different geographical regions of the earth.

We shall present the visualisations in dynamic form using the \texttt{plotly} package of R (Sievert, 2020), that provides several advantages when compared to traditional static plots. For example, visualisations created in \texttt{plotly} allow interactive manipulations, such as zoom in/out, range selection, and trace removal with axis adjustment. In addition, \texttt{plotly} is easy to use and thus serves as a good introductory tool for visualising data in R. Of course, the graphs here are displayed only in static form, but the dynamic versions can be reproduced using the R code provided in our \textit{GitHub} repository\textsuperscript{21}. We encourage interested readers to do so. For working with R code, we recommend installing and using the free \textit{RStudio}\textsuperscript{22} software available in the \textit{GitHub} repository. Additional recommendations on how to involve students in using the R language follow in Section 11.4.

The five datasets are:

1. Vostok Ice Core (Antarctica)
2. Rise of CO\textsubscript{2} (“Keeling curve”)
3. Global Average Temperature
4. Sea Level Rise
5. Sea Ice Retreat

In the following, we shall walk through each of the five datasets summarizing the phenomena behind them, displaying views of the data and R code, plotting the graphs, and discussing key points related to the R code.

Our aim is to show and discuss the possibilities of exploring and visualising climate change data with R. We emphasize reproducibility by providing the complete R scripts online. Although our views of R code may look quite technical, we think that the views and the online R scripts will help the interested reader to get to grips with R. Those readers who are not interested in the programming issues can largely ignore the R codes and the more technical sections of the text and instead concentrate on the visualisations to help them understand the practical implications of exploring the climate change data with R. The visualisations are also directly accessible through an app with the link \url{https://nrguimaraes.shinyapps.io/climateChanger/}.

### 11.3.1 Vostok Ice Core (Antarctica)

The first dataset is entitled the Vostok Ice Core. Vostok is a research station in Antarctica and one of the coldest places on this planet. Snow accumulates very slowly there, and an ice core

\textsuperscript{19} \url{https://rethinking.rbind.io/2018/11/16/the-top-five-climate-charts-using-ggplot2/}

\textsuperscript{20} \url{https://ggplot2.tidyverse.org/}

\textsuperscript{21} \url{https://github.com/nrguimaraes/climateChangeR}

\textsuperscript{22} \url{https://rstudio.com/}
contains a long, accurate record of the temperature at Vostok, and of the atmospheric composition, because air bubbles trapped in the ice are little samples of the old atmosphere.

In the 1970s and 1980s, a French-Russian team drilled a 2083-meters-long ice core at the Vostok station, revealing the CO$_2$ concentration in the atmosphere and temperature at Vostok for about the past 400,000 years. The Vostok Ice Core data (Barnola et al., 2003) consists of two time series: `vostok_co2` and `temperature`. They are measured from the ice core so that the CO$_2$ concentration measurements are best estimates from each depth level of the ice core (Barnola et al., 1987) and the temperatures refer to the variation of the Vostok isotope temperature records as differences from the Vostok’s modern surface-temperature mean value of -55.5 degrees in Celsius (Jouzel et al., 1987). Both time-series are included in an R list object `datasets` that contains all the datasets (as R data frames) needed in producing the five graphs of this section. In the R code, the Vostok Ice Core datasets are referred to with their specific names as `datasets$vostok_co2` and `datasets$temperature`, respectively. Calling those names in the R script gives brief views of so-called tibbles containing the datasets: the dimensions, the names of the variables, and the first ten records (see Dataview 11.1).

```
> datasets$vostok_co2
# A tibble: 363 x 4
  depth age_ice age_air  co2
   <dbl>   <dbl>   <dbl> <dbl>
1   149.    5679    2342  285.
2   173.    6828    3634  273.
3   177.    7043    3833  268.
4   229.    9523    6220  262.
5   250.   10579    7327  255.
6   266    11334    8113  260.
7   303.   13449   10123  262.
8   321.   14538   11013  264.
9   332.   15208   11326  238.
10  342.   15922   11719  238.
# ... with 353 more rows
```

```
> datasets$vostok_temperature
# A tibble: 3,311 x 4
  depth age_ice  deuterium  temp
     <dbl>    <dbl>     <dbl> <dbl>
1      0       0     -438   0
2      1      17     -438   0
3      2      35     -438   0
4      3      53     -438   0
5      4      72     -438   0
6      5      91     -438   0
7      6     110     -438   0
8      7     129     -438   0
9      8     149     -443 -0.81
10     9     170     -438  0.02
# ... with 3,301 more rows
```

Dataview 11.1. First records of the Vostok Ice Core data sets

Fig. 11.1 shows the graph plotted using the two datasets. The graph is a combination of two time-series plots, the CO$_2$ concentration and temperature variation, with a shared time scale

---

23 https://r4ds.had.co.nz/tibbles.html
of 400,000 years. It clearly reveals the last four, so-called glacial cycles, each lasting about 100,000 years. Details of the phenomenon and how it is linked to the variations in Earth’s orbit, are explained in Rahmstorf’s presentation. Put in perspective, compare these graphs with the Keeling curve (Fig. 11.2), indicating today’s CO$_2$ value above 410 ppm. Never in the past 400,000 years has the CO$_2$ concentration been nearly as high as it is now.

![Paleoclimate: The Link Between CO2 and Temperature](image)

Fig. 11.1. The Vostok Ice Core graphs, displaying time series for atmospheric CO$_2$ concentration (above, 11.1a) and for temperature variation (below, 11.1b) Codeview 11.1 shows the R code used to create the plot in Fig. 11.1. The plot is built from two R objects that are both results of plotting a scatter diagram of two variables from the two datasets using the `plot_ly` function with various graphical parameters. As the next step, the two objects are combined in one (Fig. 11.1) using the `subplot` function that also gives a common header for the resulting plot. Finally, the plot is displayed simply by calling its name.

```r
# Build the first plot with respect to co2
figla <- plot_ly(datasets$vostok_co2, x = ~age_ice, y = ~co2, type = 'scatter', mode = 'lines', name = ~"Co2") %>%
  layout(xaxis = list(autorange = 'reversed', range = c(420000, 0), showticklabels = FALSE, title = ""),
yaxis = list(title = 'CO2 concentration'))

# Build the second plot with respect to temperature
figlb <- plot_ly(datasets$vostok_temperature, x = ~age_ice,
y = rollmean(datasets$vostok_temperature$temp, 8, na.pad = TRUE),
type = 'scatter', mode = 'lines', name = ~"Temperature") %>%
  layout(xaxis = list(title='Years before present', autrange = 'reversed', range=c(420000, 0),
showticklabels = FALSE, title = ""),
yaxis = list(title = 'Temperature (C)'))

# Aggregate both with a shared X value (for better visualisation interaction)
fig1 <- subplot(nrows=2, figla, figlb, shareX = TRUE, shareY = FALSE) %>%
  layout(title = "Paleoclimate: The Link Between CO2 and Temperature")
```

http://www.pik-potsdam.de/~stefan/5datasets_rahmstorf.pdf
# Display the plot
fig1

Codeview 11.1. R code for the Vostok Ice Core graphs

## Activity 11.1: Exploring the relationship between CO₂ level and temperature

1. Looking at Fig. 11.1: What kind of relationship do you expect between CO₂ level and temperature?
2. Imagine the two time series for CO₂ and temperature in one plot by overlaying the two curves. What do you observe? For creating such a plot with R, go to GitHub and download the file `activities.R` for help.
3. Paleoclimatologists - these are scientists who study the climate during Earth's different geologic ages - see the cause for the variation in temperature and CO₂ as being initiated by different orbital changes. Search the Internet to find out more about the interaction between temperature, CO₂ level and orbital changes. What questions come to your mind? What are your conclusions?
4. Compare the variation in CO₂ and temperature in years before present with today’s changes (see the CO₂ curve in section 11.3.2 and temperatures in section 11.3.4.).

### 11.3.2 Rise of CO₂ (“Keeling curve”)

The second dataset is another well-known one, related to the CO₂ concentration in the atmosphere (measured in parts per million by volume, ppmv). In 1958, Charles David Keeling began the measurements on Mauna Loa, Hawaii, and they have been continued on a regular basis. Mauna Loa Observatory is located at an elevation of 3397 meters above sea level and it is one of the most important atmospheric research facilities due to the strong marine inversion layer presented in that location. This allows not only a minimum interference of vegetation and human activity in the measurements but also a clearer separation between the polluted lower portions of the atmosphere and the much cleaner troposphere²⁵. Dataview 11.2 displays the first ten records of monthly CO₂ averages.

```
> datasets$co2
# A tibble: 756 x 8
     year month date       average de_seasonalized days st.devofdays unc_of_mon_mean
   <dbl> <dbl> <date>       <dbl>           <dbl>        <dbl>            <dbl>            <dbl>
1   1958   13 1958-01-21    318.2           318.1         13.3           0.725           0.0847
2   1958   14 1958-01-22    318.2           318.1         13.3           0.725           0.0847
3   1958   15 1958-01-23    317.9           317.9         13.3           0.725           0.0847
4   1958   16 1958-01-24    317.8           317.8         13.3           0.725           0.0847
5   1958   17 1958-01-25    318.3           318.3         13.3           0.725           0.0847
6   1958   18 1958-01-26    319.3           319.3         13.3           0.725           0.0847
7   1958   19 1958-01-27    320.4           320.4         13.3           0.725           0.0847
8   1958   20 1958-01-28    321.6           321.6         13.3           0.725           0.0847
9   1958   21 1958-01-29    322.2           322.2         13.3           0.725           0.0847
10  1958   22 1958-01-30   322.9           322.9         13.3           0.725           0.0847
```

²⁵ [https://www.esrl.noaa.gov/gmd/obop/mlo/](https://www.esrl.noaa.gov/gmd/obop/mlo/)
7 1958 9 1958-09-01 313. 316. -1 -9.99 -0.99
# ... with 746 more rows

Dataview 11.2. First records of the “Keeling curve” dataset

Fig. 11.2 displays the famous “Keeling curve” for the past 60 years (Tans and Keeling, 2020) that shows the increasing upward trend and the seasonal cycle in atmospheric CO₂. The figure shows the data with their seasonal variation and a trend curve, as provided in the data. More details are available in Rahmstorf’s presentation.

![Keeling Curve](http://www.pik-potsdam.de/~stefan/5datasets_rahmstorf.pdf)

Fig. 11.2 The “Keeling curve” from 1958 to 2020

With the dataset embedded into an R environment (e.g., RStudio), there are multiple possibilities to visualise the data and explore more of the data interactively (see Cleveland, 1993). For example, the function used to aggregate data (currently the \texttt{mean} function), could easily be changed. Some common examples are the maximum, minimum, or median (depending on what type of studies the user is conducting). In addition, the function can be displayed side by side in the current visualisation by using the \texttt{add trace} function. We encourage the readers to modify the script to enhance the visualisation by adding a trace to show, for example, the maximum CO₂ value for each year.

\footnote{http://www.pik-potsdam.de/~stefan/5datasets_rahmstorf.pdf}
Chapter 11: Exploring Climate Change Data with R

Codeview 11.2 shows the R code used to create the plot in Fig. 11.2. First, outliers (denoted by “-99.99”) are removed and then the data is aggregated by year and month, computing the means. Finally, the scatter plot is constructed using the time on the horizontal axis and adding the trend (one of the variables) with graphical details.

```r
# Remove outliers
datasets$co2 <- datasets$co2[datasets$co2$average != -99.99, ]

# Aggregate data
agg <- aggregate(x = datasets$co2,
                 by = list(datasets$co2$year, datasets$co2$month),
                 FUN = mean)
agg <- agg[order(agg$year), ]

# Build plot with respect to co2 in atmosphere
fig2 <- plot_ly(agg, x = ~ date, y = ~ average,
               type = 'scatter',
               mode = 'lines', name = "co2") %>%
               add_trace(y = ~ de_seasonalized, name = "trend") %>%
               layout(title = 'Rising Atmospheric CO2 (Keeling Curve)')
# Display the plot
fig2
```

Codeview 11.2. R code for the “Keeling curve”

The notation `%>%` in Codeview 11.2 refers to a “pipe” that takes the result (output) from a previous function as the input of the next function. In this case, the result of the `plot_ly` function is taken via the pipe to the `add_trace` function and further to the `layout` function, hence constructing the graph step-by-step. After that, the result (the final graph) is saved in the object `fig2`. It could be given similarly with a pipe, but it is a conventional style in R to use the left arrow. The pipe represents a more recent and straight-forward style of writing R code. The new style has become popular due to `tidyverse`, a versatile collection of R packages designed for data science, standardising common data structures, and an underlying design philosophy, and grammar27.

Activity 11.2: Exploring the Keeling curve

1. Fit a linear regression line to the data and predict the atmospheric CO₂ level for the year 2030 (help for implementation in R is on GitHub in file `activities.R`).
2. Draw a residual plot with respect to the linear fit (help on GitHub).
3. Explore the seasonal variations. What are they caused by? (Note: This requires exploring plant physiology.)
4. Is a linear model an appropriate description for the data? Explain!

11.3.3 Global Average Temperature

The next dataset provides information on the global land-ocean mean temperature, measured by weather stations and ships around the world for each year from 1880 to the present

27 https://www.tidyverse.org/
(NASA GISTEMP Team, 2020). The data represents so called temperature anomalies or deviations that indicate how much warmer or colder it is than normal for a particular place and time. For the provider of the data, normal always means the average over the 30-year period 1951-1980 for that place and time of year.

The first ten records of the dataset are displayed in Dataview 11.3.

```r
> datasets$land_ocean_temp
# A tibble: 141 x 2
  date       annmean
  <date>       <dbl>
1 1880-01-01  -0.16
2 1881-01-01  -0.08
3 1882-01-01  -0.1
4 1883-01-01  -0.16
5 1884-01-01  -0.28
6 1885-01-01  -0.32
7 1886-01-01  -0.3
8 1887-01-01  -0.35
9 1888-01-01  -0.16
10 1889-01-01  -0.1
# ... with 131 more rows
```

Dataview 11.3. First records of the global temperature data set

The data visualised in Fig. 11.3 includes both the curve for the annual mean temperature anomaly (deviation from the 1951 to 1980 means) and a non-linear trend. Since the 1970s, the anomalies are positive, which shows that the warming has been strong and ongoing, and even accelerating faster in recent years.

![Global Land-Ocean Temperature Index (LOTI)](image)

Fig. 11.3 The global average temperature rise
Codeview 11.3 shows the R code used to create the plot in Fig. 11.3. First, the data is smoothed using a suitable span (found by experimenting visually with some typical smoother like loess, a locally estimated scatterplot smoother) and then the means are plotted as a time-series, adding the smoothed values as a non-linear trend line with some graphical details. Here, this trend is computed using the loess.smooth function which receives the $x$ and $y$ data and computes the loess smoothing. The output of this function is a data frame with the respective $x$ and $y$ for the trend, which makes it easy to add it to an existing plot (using the add_trace function), as we have done in Fig. 11.3.

```r
# Create smoothing data
loess_smooth <- loess.smooth(y = datasets$land_ocean_temp$annmean,
                             x = datasets$land_ocean_temp$date, span = 0.2)

# Build plot of the global land-ocean temperature as well as the smooth line
fig3 <- plot_ly(data = datasets$land_ocean_temp, x = ~ date, y = ~ annmean,
                type = 'scatter', mode = 'lines', name = "Annual mean") %>%
                add_trace(data = loess_smooth, x = loess_smooth$x, y = loess_smooth$y,
                           name = "Loess smoothing") %>%
                layout(title = 'Global Land-Ocean Temperature Index (LOTI)',
                       yaxis = list(title = 'Temperature Anomaly (C)'),
                       xaxis = list(title = 'Date'))

# Display the plot
fig3
```

**Codeview 11.3 R code for the temperature rise graph**

**Activity 11.3: Can the number of record temperatures be explained as random fluctuations?**

1. First, we have to define what we mean by a record in a time series: given a sequence of numbers $x_1, x_2, \ldots, x_k$, we call $x_k$ a record, if $x_k$ is bigger than all of its predecessors, i.e. if $x_k > x_v$ for $v < k$. For example, in the sequence 5, 3, 4, 7, 5, 11, 13, the number 7 is a new record because 7 is larger than the three preceding figures. For the same reason, 11 and 13 are records. The initial figure, here 5, is always a record. The above sequence contains 4 records. If we notice in a time series too many records, we doubt that the sequence is completely random. Too many records are indicative of an upward trend, too few records let us suspect a downward trend.

2. Now download the R code from GitHub and reproduce Fig. 11.3. The plotly implementation allows you to zoom in on specific areas of the graph. Look at the temperature values between the years 1980 and 2019 and count the number of records. Could you identify 11 records?

3. Assume for a moment that the temperature data between 1980 and 2019 were completely random, without any trend over time. How likely is it to observe 11 records in a random sequence of length 40? To figure that out you could resort to somewhat sophisticated combinatorial mathematics, or you can address the problem via simulations. The file activities.R on GitHub gives you the code for finding the probability of 11 records in a random sequence of length 40 via simulation.
11.3.4 Sea Level Rise

The fourth data set concerns the rise of the sea level (Nerem et al., 2018). It consists of two distinct parts. The older part is based on tide gauge records at coastal stations, and it offers the reconstructed data from 1880 to 2009 as described in Church and White (2011). The newer records (since 1993) are accurate satellite measurements. The unit here is GMSL (global mean sea level). The data are reported as changes relative to January 1, 1993 and are 2-month averages. The first six and last six records of the combined dataset are displayed in Dataview 11.4, using R functions head and tail, respectively.

```r
> head(datasets$sea_level)
# A tibble: 6 x 3
  date               method                          gmsl
<dttm>              <chr>                      <dbl>
1 1880-01-16 00:00:00 Coastal tide gauge records -181.
2 1880-02-16 00:00:00 Coastal tide gauge records -170.
3 1880-03-17 00:00:00 Coastal tide gauge records -163.
4 1880-04-17 00:00:00 Coastal tide gauge records -157.
5 1880-05-17 00:00:00 Coastal tide gauge records -157.
6 1880-06-17 00:00:00 Coastal tide gauge records -158

> tail(datasets$sea_level)
# A tibble: 6 x 3
  date               method                          gmsl
<dttm>              <chr>                  <dbl>
1 2017-11-18 00:00:00 Satellite observations  83.4
2 2017-11-28 00:00:00 Satellite observations  84.8
3 2017-12-08 00:00:00 Satellite observations  83.3
4 2017-12-18 00:00:00 Satellite observations  83.9
5 2017-12-28 00:00:00 Satellite observations  81.7
6 2018-01-06 00:00:00 Satellite observations  82.0
```

Dataview 11.4. First and last records of the sea level data set.

Fig. 11.4 shows the data combined in one time-series plot. The sea rise since 1880 is over 200 mm (or 20 cm) and it has accelerated in the 2000s.

---

29 http://sealevel.colorado.edu/files/2018_rel1/sl_ns_global.txt
Codeview 11.4 shows the R code used to create the plot in Fig. 11.4. First, the variable name of the series is updated to reflect the measurement type (tide gauge records vs satellite observations). Then the data is plotted as a time-series, where the colour depends on the measurement type.

```r
# Replace the method values for a more comprehensive visualisation
datasets$sea_level <- datasets$sea_level %>%
  mutate(method = replace(method, method == 'gmsl_tide', "Coastal tide gauge records")) %>%
  mutate(method = replace(method, method == 'gmsl_sat', "Satellite observations"))

# Build the plot for sea_level data
fig4 <- plot_ly(datasets$sea_level, x = ~date, y = ~gmsl,
                color = ~method, type = 'scatter', mode = 'lines')

# Display the plot
fig4
```

Codeview 11.4. R code for the sea level rise graph

**11.3.5 Sea Ice Retreat**

The last one of *The Five Most Important Data Sets of Climate Science* concerns the sea ice and its retreat, with the focus on the polar region (Fetterer et al., 2017). It is a known fact that the ice extent is shrinking faster than predicted by climate models. Global warming is amplified, as the white ice (that reflects about 90% of the incident solar energy back into
space) will be replaced by the dark ocean (that in turn absorbs about 90% of the solar energy). Dataview 5 shows the first ten records of the dataset.

```
> datasets$polar_ice
# A tibble: 41 x 6
  year                  mo `data-type` region extent  area
  <dttm>                <dbl> <chr>       <chr>   <dbl> <dbl>
1 1979-01-01 00:00:00   9 Goddard     N       7.05  4.58
2 1980-01-01 00:00:00   9 Goddard     N       7.67  4.87
3 1981-01-01 00:00:00   9 Goddard     N       7.14  4.44
4 1982-01-01 00:00:00   9 Goddard     N       7.2  4.43
5 1983-01-01 00:00:00   9 Goddard     N       7.39  4.7
6 1984-01-01 00:00:00   9 Goddard     N       6.81  4.11
7 1985-01-01 00:00:00   9 Goddard     N       6.7  4.23
8 1986-01-01 00:00:00   9 Goddard     N       7.41  4.72
9 1987-01-01 00:00:00   9 Goddard     N       7.28  5.64
10 1988-01-01 00:00:00  9 Goddard     N       7.37  5.36
# ... with 31 more rows
```

Dataview 11.5. First ten records of the sea level data set

Fig. 11.5 displays the time-series (the measurements connected by a line) with a linear trend from the end of the 1970s to present. One might ask, whether fitting a linear trend is reasonable as there is quite much variation around the regression line. It is also good to remember that the dataset includes only 41 observations, one per each year. The higher variability in some periods might be caused by chance, or there could be another factor explaining it. The linear trend is probably oversimplified, at least for a predictive model.

Fig. 11.5 Sea ice retreat from 1979 to 2019
Codeview 5 shows the R code used to create the plot in Fig. 11.5. First, a linear model is fitted to find the linear trend based on years. Then, the data is plotted as a time-series, adding the trend and fine-tuning some of the graphical parameters.

```r
# Fitting with linear model
lm_e <- lm(datasets$polar_ice$extent ~ datasets$polar_ice$year)

# Build the plot for the arctic sea ice minimum with the fitted values
fig5 <- plot_ly(datasets$polar_ice, x = ~year, y = ~extent,
   type = 'scatter', mode = 'lines', name = "measure") %>%
   add_trace(x = datasets$polar_ice$year, y = lm_e$fitted.values,
     name = "Linear Regression") %>%
   layout(title = 'Arctic Sea Ice Minimum',
     yaxis = list(title = 'million square km'),
     xaxis = list(title = 'Year'))
```

**Codeview 11.5. R code for the sea ice retreat graph**

Activity 11.4: Exploring the trend

1. Determine by eyeball analysis the slope of the fitted linear function describing the decline in arctic sea ice over the last 40 years.
2. Based on this trend, make a prediction of the ice volume for the year 2030.
3. In chapter 5.3.1 you encountered a different graphical representation of the loss in arctic sea ice over time. Compare the two visualizations and assess their pros and cons.
4. To investigate the quality of the fit, it is helpful to investigate the residual plot, i.e. the plot of time versus the difference between data and fitted values. Draw a residual plot. The file `activities.R` on GitHub provides you with technical details. Can you discern a pattern in the residuals?
5. Another possibility to check if the trend is linear is by adding a smooth curve to the graph (again, see the file `activities.R` on GitHub for help). The span is a smoothing parameter, governing how closely the resulting curve follows the data. By default, we have set `span=0.3`. Change the value of the smoothing parameter, observe the resulting curve and describe what you see.
11.3.6 Using Geographical Maps

Recent developments in computer graphics and animation have shaped the way in which data are presented in the media as images or as animated simulations. Visual representations play a central role in public communication and aim to represent the relevant dynamics and content in an easily understandable way. Choropleth maps are coloured maps which are a popular method to display spatially distributed data: the colour makes it easy to see the differences between areas. Many types of data can be placed in one picture in a comprehensible way. This is true in particular for animated maps which illustrate changes over time.

We provide the reader with resources and demonstrations of the R language to represent data through animated interactive maps, by adding a component that visualises the change over time. We continue using plotly as the visualisation tool, as it allows the mapping and visualisation of geospatial data while also providing an animation component that is easily integrated in the visualisations.

To facilitate the replicability without exposing the reader to a detailed explanation of the syntax and functionality of the R language, we provide a complete, well-documented R script (utils_map.R) in our GitHub repository. The repository also contains the data files for the visualisations in csv format. In the following, we provide some guidelines on how the R script can be used, modified and customized.

There are three variables in the script that allow different visualisation outputs:

- **YEAR** - restrains the data to a user-specific interval of time
- **TYPE** - refers to the type of function to be used in the aggregation process of the variable temperature. Possible values are the average or maximum temperature. Additional aggregation functions can be implemented by the user.
- **REGION** - allows the user to define which regions of the maps will be shown. There are three different values that the user can input referring to the totality of the world, a set of specific countries, or the states of the USA.

Changing the combination of these variables will result in different outcomes. Fig. 11.6 presents some of the possible visualisation examples.

31 [https://github.com/nrguimaraes/climateChangeR](https://github.com/nrguimaraes/climateChangeR)
Fig. 11.6 Geographical maps showing average temperatures in selected parts of the world; snapshots from the animated displays

The script starts by loading the necessary libraries to perform operations such as reading the data and plotting the map. It also loads additional information that may be required to work
with data from the states of the USA. Then, we introduce the three variables that can be modified by the user (YEAR, TYPE, REGION). The data will be transformed based on their values. For example, if the reader intends to plot the temperature in the states of the USA, the data must be loaded and the names of the states must be converted to their abbreviated codes. All this will happen by running the ready-made R code presented in Codeview 11.6.

```r
clim_data <- read.csv("GlobalLandTemperaturesByState.csv")

# Allow only the complete cases
clim_data <- clim_data[complete.cases(clim_data),]

# Conversion from date to different columns
clim_data %>% separate(col = dt, into = c("Year", "Month", "Day"), convert = TRUE) -> clim_data

# Apply the function getCode to the "States" column in dataset
codes <- unlist(lapply(clim_data$State, getCode))

# Create a new column (named variable) with the codes
clim_data$variable <- codes

# Set the location type of the map to USA-states (so the function know how to interpret the data)
location_mode <- 'USA-states'
```

Codeview 11.6 R code for loading and transforming the data from the states of the USA

After reading the file with the information from the states of the USA, we ensure using the `complete.cases` function that we only want to use those entries in the dataset where there is information in all columns. Next, we use the `separate` function to decompose the collection date into three separate columns, containing the day, month, and year for better reading.

The function `getCode` is responsible for ensuring the conversion between state names and abbreviations. This conversion is necessary for the `plotly` package to be able to map the states to their respective positions. Finally, we set our `location_mode` variable to “USA-states” so that the `plotly` library knows how to interpret the data.

The `YEAR` and `TYPE` variables are handled next in the script. The `YEAR` variable allows us to filter the data frame if the type of information provided on the variable is an interval of years. When the `TYPE` is set to “average”, the code in Codeview 11.7 will be executed.

```r
clim_data %>%
  select(Year, AverageTemperature, variable) %>%
  group_by(Year, variable) %>%
  summarise(value = mean(AverageTemperature)) -> clim_dataf
```

Codeview 11.7 R code for aggregating the data by average

The code again makes use of pipes (```%>%```) for better comprehension. The data frame is passed on to the `select` function, which will only use the `Year`, `AverageTemperature`, and `variable` (representing the US state codes). It is important to highlight that this `AverageTemperature` is the measure provided in the dataset while `TYPE=“average”` refers to the function used for aggregation of the data (since there are multiple measurements in each
year). The `group_by` function will group identical values in the `Year` and `variable` column and `summarise` will apply the mean (i.e. the average) to those grouped values. Finally, everything is stored in a new R data frame (`clim_dataf`). We opt to use a new name for the data so the reader can have a better perception on the transformations that occur in the data.

The code is very similar when the `TYPE` variable is set to “max”, the only difference being the function used to summarise the data. With that in mind, the reader can easily add conditions to work with other functions, e.g., minimum temperature. The only requirement would be to copy the `else if` condition, change the condition to `TYPE=="min"` and set the `value` argument of the `summarise` function to `value=min(AverageTemperature)`.

Due to space constraints and since it is not the goal of this chapter to provide a complete introduction to R, we will conclude this section by focusing on the part of the script responsible for the visualisation of the data. Codeview 11.8 presents the code that uses `plotly` functions to present the transformed dataset in map format.

```r
fig <- clim_dataf %>%
  plot_ly(
    locationmode=location_mode,
    z = ~Temperature,
    locations=~variable,
    frame = ~Year,
    type = 'choropleth',
    showlegend = TRUE,
    colorscale='bluered',
    zmax=max_value,
    zmin=min_value
  )

fig <- fig %>% layout(
  title= plot_title<-paste0("Evolution of the Average Temperature (C) through the years ",clim_dataf$Year[1],"-",clim_dataf$Year[nrow(clim_dataf)],""),
  geo=g
)
fig <- fig %>%
  animation_opts(
    frame=100
  )
fig
```

Codeview 11.8 R code for the visualisation of the map

The code is quite straightforward. The transformed data frame (`clim_dataf`) is passed down to the `plot_ly` function responsible for displaying the map. In this function, certain arguments must be provided. We shall briefly explain our choices of arguments to help the user in understanding how to build such visualisations.

The `z` argument corresponds to the variable we want to fill the map with (in this case the temperature column), with the `locations` argument referring to the location of that data. The `showlegend`, `colorscale`, `zmax` and `zmin` refer to the visibility of the legend, its colour scheme and the maximum and minimum value of it. Some of these arguments can be omitted with `plotly` adjusting to the default values. However, in the case of `zmax` and `zmin` it is important
to maintain the same values in the animation so that the visualisation does not lead to misinterpretations (the default value of these arguments adjust the scale according to the data in each year). Finally, to add animation to the visualisation, we simply have to define what is the variable we want to change with each frame (in our case the years). This is achieved using the frame argument.

Optional configurations are then added to achieve a cleaner and easy-to-interpret visualisation. We use the layout function to define the title and configure some geographic settings. The animation_opts is used to slow the default duration of each frame to 100 milliseconds. Please refer to the comments on the script and the official plotly documentation in R for additional information in these functions and arguments.

With the provided script and the information on this section, we do hope that readers will gain some appreciation of the flexibility and ease of use of the R language for visualising geospatial and time-dependent climate data.

Activity 11.5: Creating an animated map

Create an animated map for your country (or any land of your dreams) and its neighbours to explore trends in the maximum temperatures across a chosen time period. See the file activities.R on GitHub for help.

11.4 Recommendations for implementation in a classroom environment

The primary aim of this chapter has been to facilitate the use of sophisticated software and engage students in exploring and visualising data about global warming and climate change. In this final section, we provide some technical guidelines for instructors to introduce the subject and replicate the visualisations provided in a classroom context.

The first requirement is to install R. It is very easy, because R is freely downloadable from the official R project website. R has a simple command-based interface, but in order to facilitate the visualisations and code comprehension, we highly recommend also downloading and installing RStudio software. It provides an integrated development environment (IDE) that allows an easier interaction between the R code, data, and visualisations.

As we have stated earlier, all the R code and data presented in this chapter are available for download from our GitHub repository. GitHub is a huge platform that allows development teams to store and share their code as well as update and manage version control on the code.

---

32 https://www.r-project.org/
33 https://rstudio.com/
34 https://github.com/nrguimaraes/climateChangeR
It also allows advanced functionalities such as methods for code merging and development trees.

With R and RStudio installed, the user may download the code from the GitHub repository, by finding the “Code” button and selecting “Download as Zip” (in the current layout of GitHub). When the download is complete, the user should extract the files to a folder of their preference and launch RStudio.

The next thing to do is to open one of the scripts that build the visualisations. To do that, just select File -> Open File and then select the preferred script. When the script is loaded and since some external datasets are required it is necessary to “tell” R what the current working directory is. In our case, we want our working directory to be the same as the script (since it is where the datasets are also located). Thus, we select Session -> Set working directory -> To source file location. This way we set the R environment ready to run the scripts. The first time each script is loaded, the necessary libraries are installed (it may take some minutes). After that, everything should work exactly as described in the previous sections.

The scripts provided in this chapter are highly customisable. This allows students to adapt the script to better comprehend the data and the R language itself. Let us begin by focusing on the script utils.R that is responsible for several visualisations presented in section 11.3. The first suggestion we present to readers is to visualise the plots themselves in RStudio. To do that, users should study the code to see the parts of the code associated with each plot and correctly call the plot_ly function. For example, to create the plot presented in Fig. 11.1, the user should execute all the code until the variable “fig” is created. To do this, one can select all the code to this point and use the key combination Ctrl+R (Cmd+R in Mac) or select Code -> Run Selected Line(s). After that, the plot will appear by simply typing the name of the variable (that is, fig) below the selected code and running it.

The process is similar in the rest of the script. Readers can also run the full script and then type the different plot variables (fig, fig2, …) to visualise the corresponding plot.

The plots are displayed in the plot window (usually located in the bottom right of RStudio). It is where users can see the advantages of using the plotly library since the embedded manipulation tools of the library allow a more fine-grained analysis of each visualisation. Also, RStudio allows (through the “Zoom” button in the top of the plot window) users to visualise each plot in a separate window.

We encourage instructors and students to modify the current script to explore several different advantages of manipulating and visualising climate change data with the R language. For example, when building the plot in Fig. 11.1b, we use the rolling mean function (rollmean) to smooth the temperature values in the dataset. Some parameters of this function can be changed, allowing a more fine-tuned visualisation. To find the possible options of this function, one can type ?rollmean on the console and select the zoo::rollmean function from the help window. This can be done with any function used in this script and
we encourage lecturers and readers to use the help option (by writing the name of the function preceded by a question mark) anytime they feel it necessary to better understand the code. In the help window of the `rollmean` function, each parameter of the function is explained, thus allowing users to customize how to fill the missing values or the type of alignment that they wish to do. We encourage users to understand these parameters and change the function to modify the plot in Fig. 11.1b.

Similarly, in Fig. 11.3 the Keeling curve consists of a trend plus a superimposed seasonal component, the latter reflecting the absorption of CO$_2$ by plants. The seasonal component is caused by the CO$_2$ being absorbed into plants via photosynthesis in spring and summer, leading to a decrease in the atmospheric CO$_2$ level during these seasons. In winter, some plants are deciduous (so do not photosynthesise at all); in regions where winter is associated with low temperatures, photosynthesis is reduced even in non-deciduous plants. The trend is modelled via data smoothing, and in this case the seasonal component can be described well through trigonometric functions.

As was mentioned in section 11.3.6, the script `utils_map.R` provides a more high-end customisation where the user can change the predefined variables to plot different geographical visualisations and animations concerning countries, time intervals, and function aggregation types (mean, max, …). We encourage lecturers to use this script in a classroom environment and challenge the student to enhance the script by providing new ways to visualise the data or transform the data to provide new visualisations.

We mentioned the addition of a function `min` (minimum temperature) in the `TYPE` variable by introducing a new `if` condition. Moreover, new data transformations can be added to visualise filtered parts of the dataset. For example, it can be interesting to study the effect of climate change in a particular season. Therefore, users can filter the data before the visualisation to only contemplate spring or winter (by using the month column). We encourage lecturers to propose this challenge in the classroom and require that the modifications made in the script must be additions and thus, the current functionalities must be kept. In this way, students will better understand the base script and how the global variables `TYPE`, `YEAR`, and `REGION` work, instead of just coding by a trial-and-error approach. We do believe that the current base script `utils_map.R` provides a good skeleton for a large number of additional exercises and challenges to transform and visualise climate data and thus, we encourage lecturers to create such activities and use them in a classroom context.

**11.5 Conclusions**

Studying climate change requires the exploration of complex spatial time series datasets, often with special tools and advanced methods. However, the basic knowledge of climate change can be understood from a few visualisations obtained with freely available data and software.
In this chapter we have given a general view on exploring climate change data, including ready-made online visualisations (such as images, maps, and animations). We have also shown a couple of examples of how to access and use climate change data from different data sources and repositories with R, the freely accessible statistical programming language and computing environment (R Core Team, 2020).

We encourage the reader to actually run our examples and to display and explore the dynamic versions of the graphs and maps. All that is required is to install R and RStudio, then download and use the R code shown in section 11.3, available in complete form in our GitHub repository. We note that a beginner of R does not need to understand every detail of the code. It is satisfying to see the results, especially the graphs, appearing in the RStudio window and to make some small modifications. Indeed, learning more R typically happens best as a process of trial-and-error, by working with ready-made examples and trying to repeat them, possibly modifying some details with the help of freely available materials, such as books and websites.

Acknowledgments

Nuno Guimarães thanks the Fundação para a Ciência e Tecnologia (FCT), Portugal for the Ph.D. Grant (SFRH/BD/129708/2017)

References


35 https://github.com/nrguimaraes/climateChangeR


