HCC Custom Charts for RAWGraphs

Bastian Kandlbauer, Laura Pessl, and Vera Tysheva

DAT.C406UF Information Visualisation SS 2025
Graz University of Technology

30 June 2025

Abstract

This project describes the creation of a suite of custom charts for the RAWGraphs data visualization framework. Building on the work of previous student groups, four existing charts were refined and improved. In addition, two new charts were developed from scratch. All charts follow RAWGraphs custom chart structure and also include some sample datasets in order to demonstrate their usage. The aim is to collect a number of custom charts for RAWGraphs into a convenient bundle for easy access and distribution.

Contents

List of Figures

List of Listings

Chapter 1

Introduction

RAWGraphs is an open-source data visualization application, which allows non-programmers to create charts and visualisations with their own data [DensityDesign et al. 2025; Mauri et al. 2017]. It closes the gap between spreadsheets and vector graphics editors. The platform allows users to create customizable charts directly within a web browser, without requiring any extensive coding knowledge. One of the key features of RAWGraphs is its flexibility to support other custom chart types, enabling users or developers to extend its visualization capabilities beyond its built-in templates.

This report describes the HCC suite of six custom charts for RAWGraphs. Building on the foundation laid by previous groups of students, four existing chart implementations were reviewed and refined: Paired Bar Chart, Connected Scatter Plot, Polar Area Chart, and Similarity Map. In addition, two entirely new chart types were implemented from scratch: Bullet Chart and Pareto Chart.

The charts follow the RAWGraphs custom chart template, and so can easily be loaded into RAWGraphs by end users. The charts are provided both individually and as a bundle of all six charts. Each chart is provided with two corresponding sample datasets, demonstrating the charts usage and capabilities.

Chapter 2

RAWGraphs

RAWGraphs is an open-source data visualization tool designed to make it easy for anyone to create charts and visualisations of their own data. Its user interface can be seen in Figures 2.1 and 2.2. To create a chart, a user proceeds through five steps:

1. Load your data: Options to paste, upload, use a data sample, use a SPARQL query, use a URL, or to open a project.
2. Choose a chart: Choose an available chart type, or load a custom chart.
3. Mapping: Map dimensions to chart variables.
4. Customize: Customise the look of the chart.
5. Export: Export the chart in one of four formats (`.svg`, `.png`, `.jpg`, and `.rawgraphs`). SVG is a scalable, vector graphics format, which can then be further edited in applications like Inkscape or Adobe Illustrator.

RAWGraphs currently supports 32 chart types out of the box, catering for a variety of different data types, as can be seen in the lower half of Figure 2.1. Additionally, it supports the option to load custom charts, each of which is a bundle of JavaScript code conforming to RAWGraphs' custom chart template. RAWGraphs is built in JavaScript with the React framework. Since it is an open-source project, developers can integrate into their own tools and workflows and extend or customise its charting capabilities.

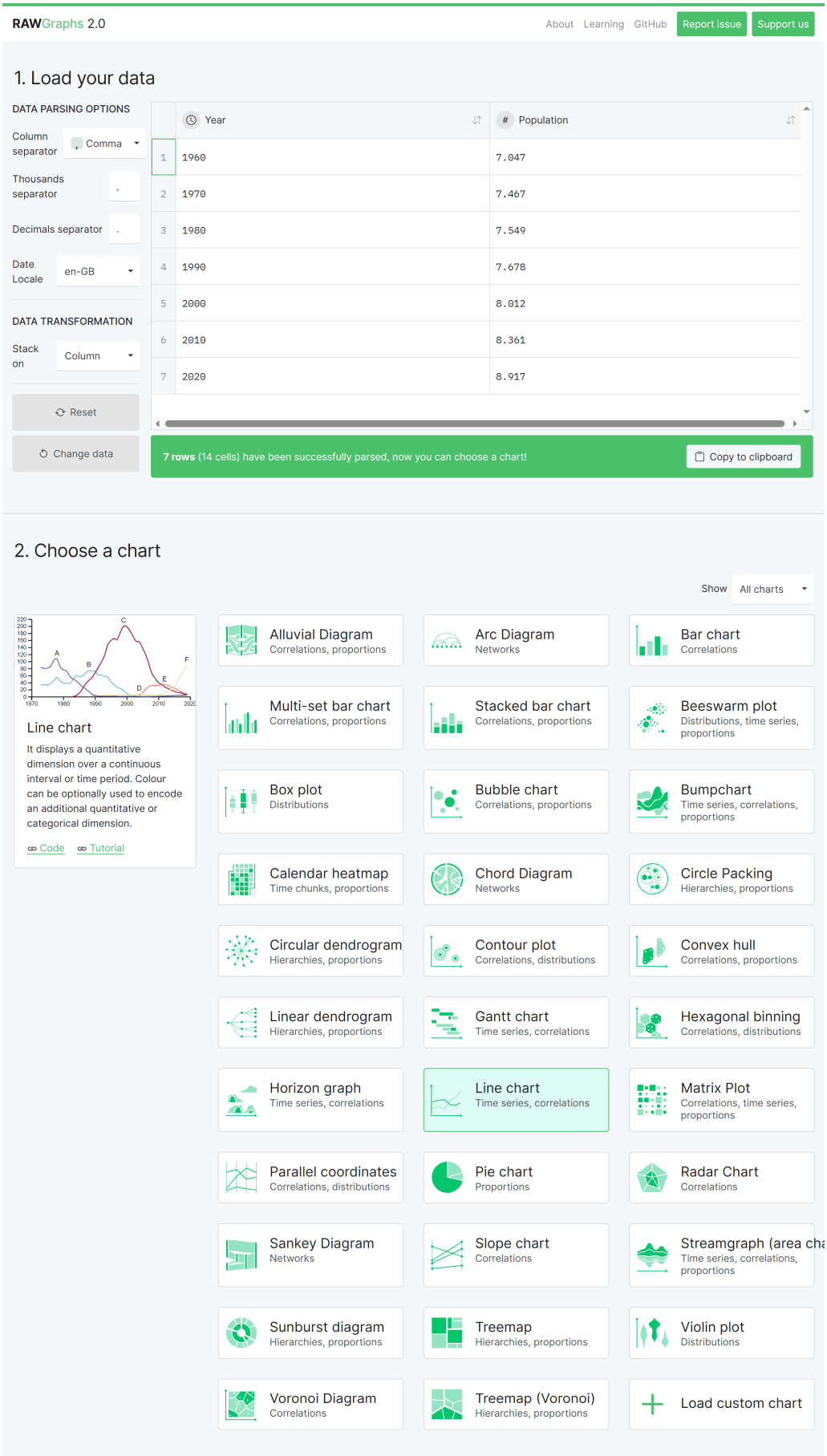

Figure 2.1: RAWGraphs: Full user interface (Part 1). Step 1: Load your data, and Step 2: Choose a chart.
[Screenshot taken by Keith Andrews using RAWGraphs [DensityDesign et al. 2025]. Used with permission.]

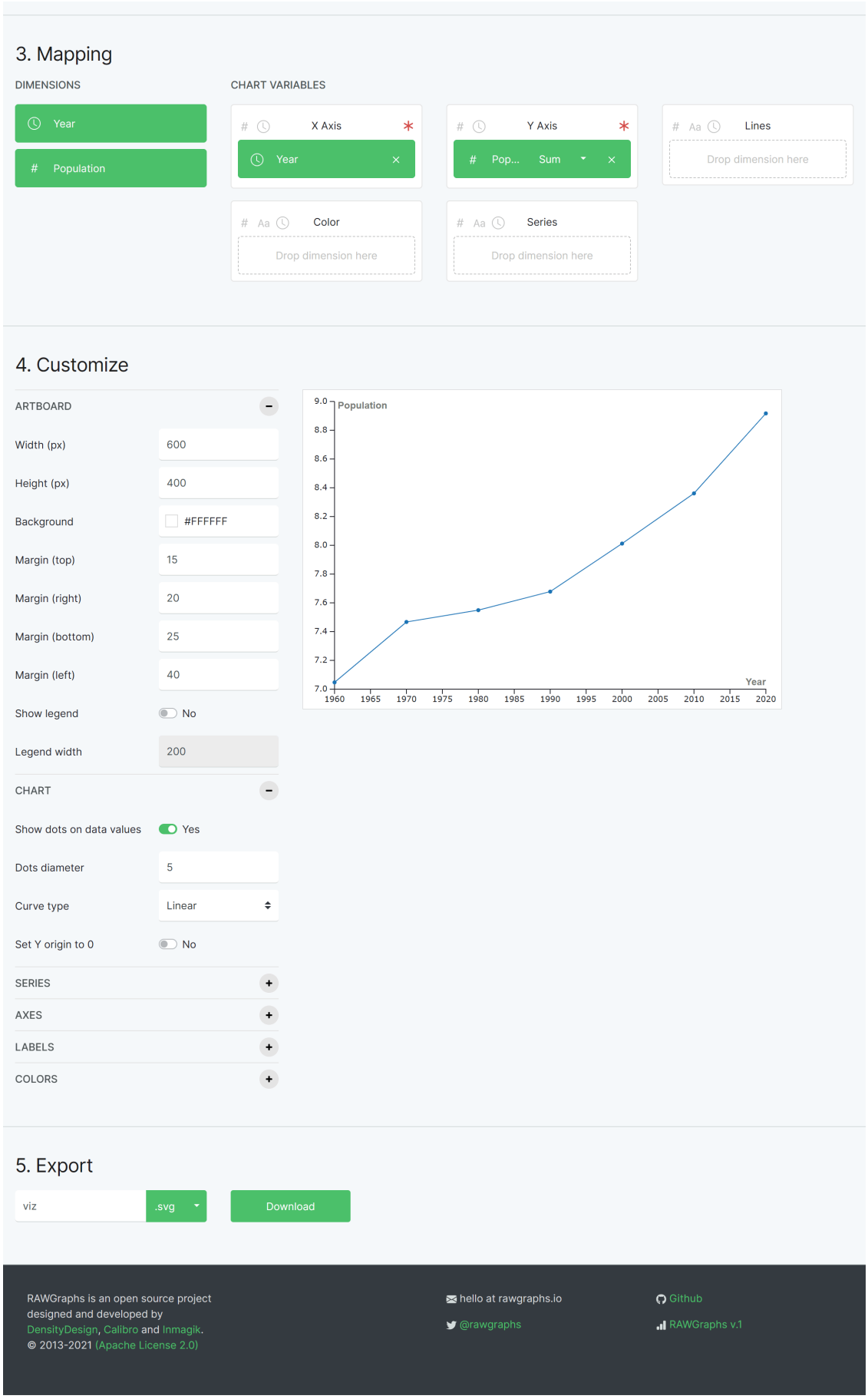

Figure 2.2: RAWGraphs: Full user interface (Part 2). Step 3: Mapping, Step 4: Customize, and Step 5: Export. [Screenshot taken by Keith Andrews using RAWGraphs [DensityDesign et al. 2025]. Used with permission.]

Chapter 3

Creating a Custom Chart

To create a custom chart for RAWGraphs, a specific structure and guidelines have to be followed. These structure and guidelines are provided in the official custom chart template [RAWGraphs 2025]. This ensures compatibility with the RAWGraphs interface and guarantees consistent behaviour across charts. When building a custom chart, begin by duplicating the `myChart/` folder located in the `docs/` directory of the template. Rename it according to the new chart's name and adjust the contents accordingly. The folder structure is shown in Listing 3.1.

The files contained in the folder are:

- `dimensions.js`: Specifies the required data dimensions for your chart, for example, X Axis, Y Axis, Size, and Color. These chart variables define how input data maps to the visual properties of the chart.
- `index.js`: Makes the chart available.
- `mapping.js`: Contains the code to transform the input data into a flat array of objects in the format required by the chart's rendering logic. This transformation is essential and must return a flat structure.
- `metadata.js`: Contains descriptive information about the chart, including its name, category, and description. This information is used to display and organise charts in the RAWGraphs interface.
- `render.js`: Contains the rendering logic to draw the chart, typically using the D3 library. This is where the chart is actually drawn, based on the mapped data and visual options.
- `visualOptions.js`: Lists all configurable visual options for the chart, such as width, height, font size, dot radius, or sorting rules. These options appear in the UI and allow users to customise the chart's appearance.
- `<chartName>.js`: Acts as a wrapper that aggregates all components of the chart into a single object to be exported by `index.js`.
- `<chartName>.svg`: The main icon of the chart, shown in the chart selector within the RAWGraphs UI.
- `<chartName>_thumb.svg`: A smaller thumbnail version of the icon used for preview purposes in the UI.

When developing a custom chart, refer to the documentation provided in the `docs/` folder of the template, in particular:

- `add-a-new-chart.md`: A step-by-step guide on how to add a new chart to the project. Follow this document closely to ensure proper integration.

```
<chartName>/
├── dimensions.js
├── index.js
├── mapping.js
├── metadata.js
├── render.js
├── visualOptions.js
├── <chartName>.js
├── <chartName>.svg
└── <chartName>_thumb.svg
```

Listing 3.1: The file and directory structure for a custom chart.

- `best-practices.md`: Tips and conventions for maintaining code quality, usability, and consistency across custom charts.

Chapter 4

HCC Custom Charts

During this course of this project at the Institute of Human-Centred Computing (HCC) at Graz University of Technology, a suite of custom charts for RAWGraphs has been assembled. It currently consists of six different charts: Paired Bar Chart, Connected Scatter Plot, Polar Area Chart, Similarity Map, Bullet Chart, and Pareto Chart. The first four are based on existing code which was reviewed and refined. The latter two were implemented from scratch. The suite is collectively called the HCC Custom Charts.

The charts are provided both individually and as a bundle of all six charts, via the project's GitHub repository [Kandlbauer et al. 2025]. Each chart is provided with two corresponding sample datasets, demonstrating the charts usage and capabilities. The original sources and permissions for the datasets are documented in the file `datapermissions.md` in the repository.

Every chart has a set of default customisation options (width, height, background, margins, and a legend where available) in the Artboard section, which are mandated by RAWGraphs and cannot be turned off by the custom chart creator. Other options in the customisation panel can be configured for each custom chart.

4.1 Paired Bar Chart

A paired bar chart is used to compare two sets of data by visualising them on either side of a common, central y axis. It is often used to show demographic data, such as the age pyramid shown in Figure 4.1. This chart was implemented by a previous group [Aumüller et al. 2024] and is fully functional. One problem that could not be resolved is the automatic colour change of the righthand bars from red to blue, which happens for some as yet unknown reason when the sample population dataset is loaded and visualised.

4.1.1 Datasets

Two sample datasets are provided for the paired bar chart. The first, `population-austria.csv`, contains information about the number of men and women in different age groups in Austria. The corresponding paired bar chart can be seen in Figure 4.1. The second, `co2-emissions.csv`, contains the total CO_2 emissions from 2022 and 2023 for six continents. The chart can be seen in Figure 4.2.

4.1.2 Mapping

The paired bar chart has three chart variables (Left Side, Y Axis, and Right Side) that need to be mapped to data dimensions in RAWGraphs. Figures 4.3 and 4.4 show the intended mappings for the provided datasets.

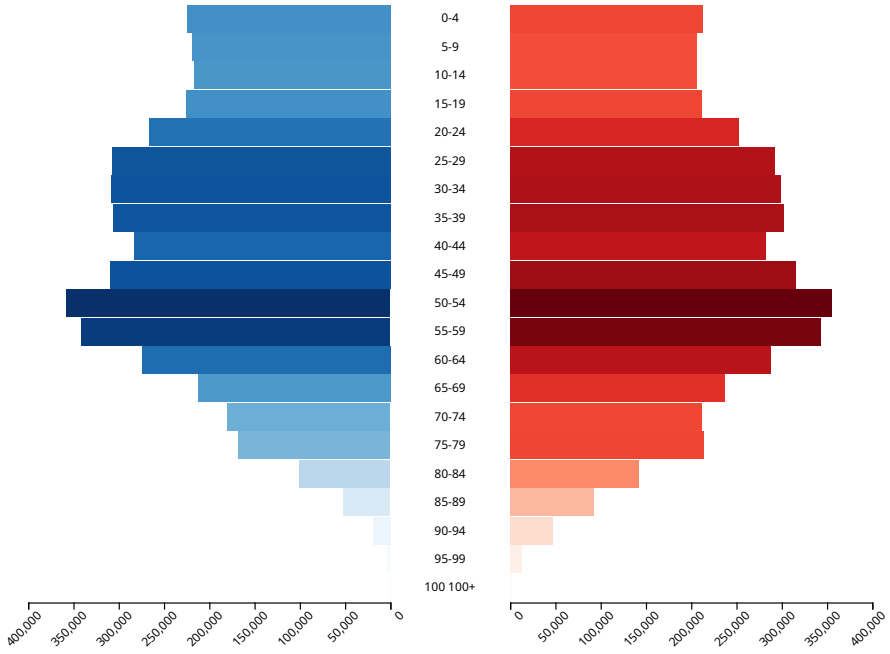

Figure 4.1: Paired Bar Chart: Age pyramid showing the number of men (left) and women (right) in different age bracktes in Austria. [Exported from RAWGraphs by Laura Pessl.]

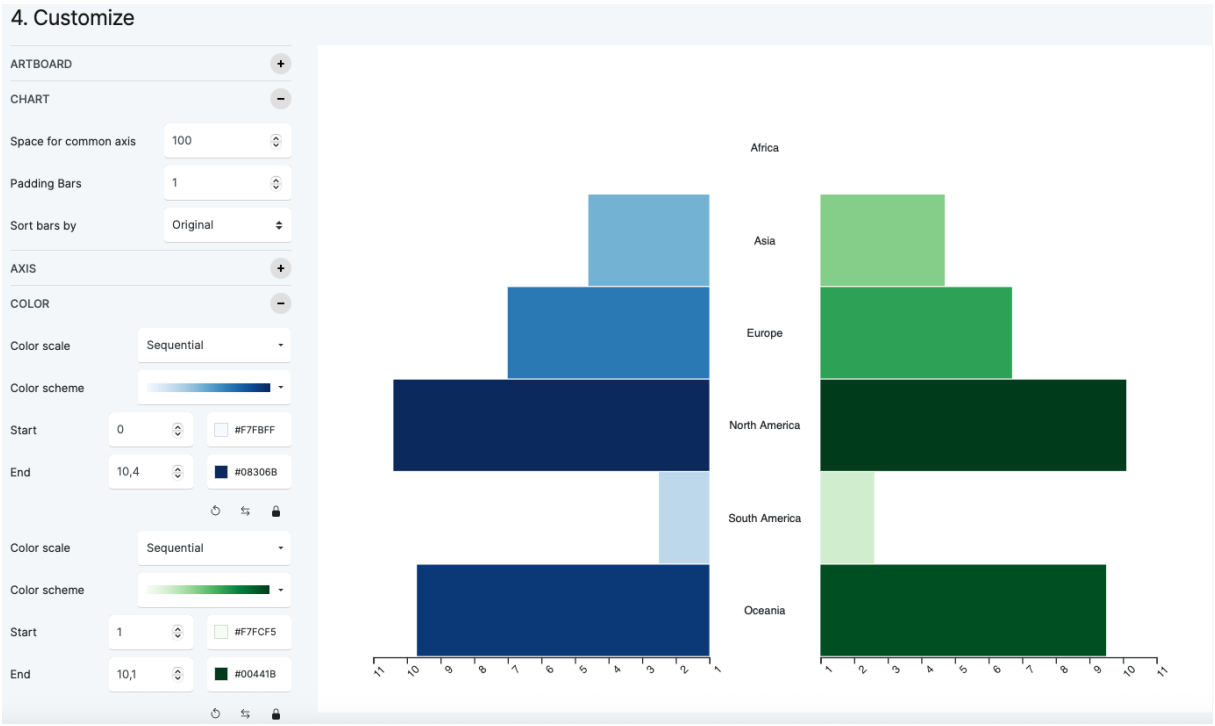

Figure 4.2: Paired Bar Chart: Customisation settings for a chart showing CO_2 emissions per continent for 2022 (left) and 2023 (right). [Screenshot taken by Laura Pessl.]

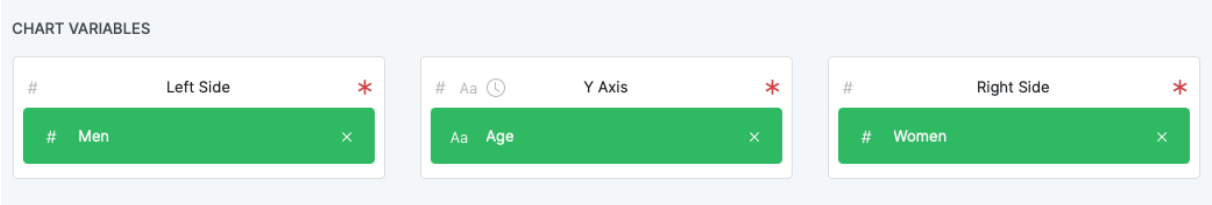

Figure 4.3: Paired Bar Chart: Mapping of dimensions to chart variables for `population-austria.csv`. [Screenshot taken by Laura Pessl.]

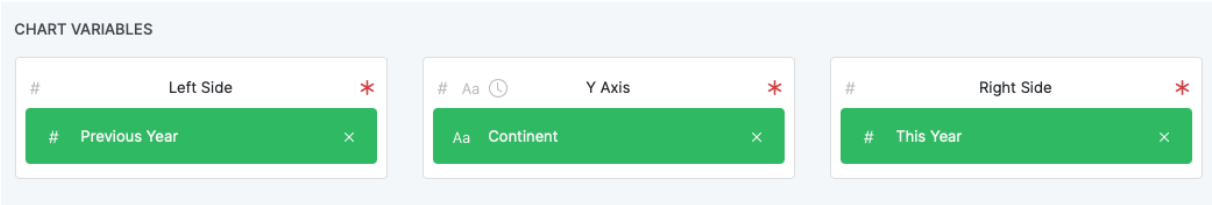

Figure 4.4: Paired Bar Chart: Mapping of dimensions to chart variables for `co2-emissions.csv`. [Screenshot taken by Laura Pessl.]

4.1.3 Customisation

The additional settings for the paired bar chart are:

- Chart: Space for y axis, padding for bars, sorting of bars.
- Axis: Axis labels, label rotation, axis visibility.
- Color: Colour scales and schemes.

Some of them can be seen on the left side of Figure 4.2.

4.2 Connected Scatterplot

A connected scatterplot is a chart consisting of data points, defined by an x and y axis, connected by lines. Arrows on the lines indicate the change over time. It is often used to observe data value changes over a longer period of time. The previously implemented chart [Aumüller et al. 2024] was improved by adding an option to set the size of the arrows. Figure 4.5 shows a connected scatterplot of population evolution in Germany.

4.2.1 Datasets

Two sample datasets are provided for the connected scatterplot: `germany-population-growth.csv` contains data about the population evolution of Germany from 1950 to 2022, and `unemployment-rate.csv` contains data about the unemployment rate in the US from 2015 to 2025. The respective connected scatterplots can be seen in Figure 4.5 and Figure 4.6

4.2.2 Mapping

The connected scatterplot has six chart variables that can be mapped to, whereby the variables X Axis and Y Axis are required. Optionally, dimensions can be assigned to variables Size, Color, Connection By and Label. Figures 4.7 and 4.8 show possible mapping configurations for the two provided datasets.

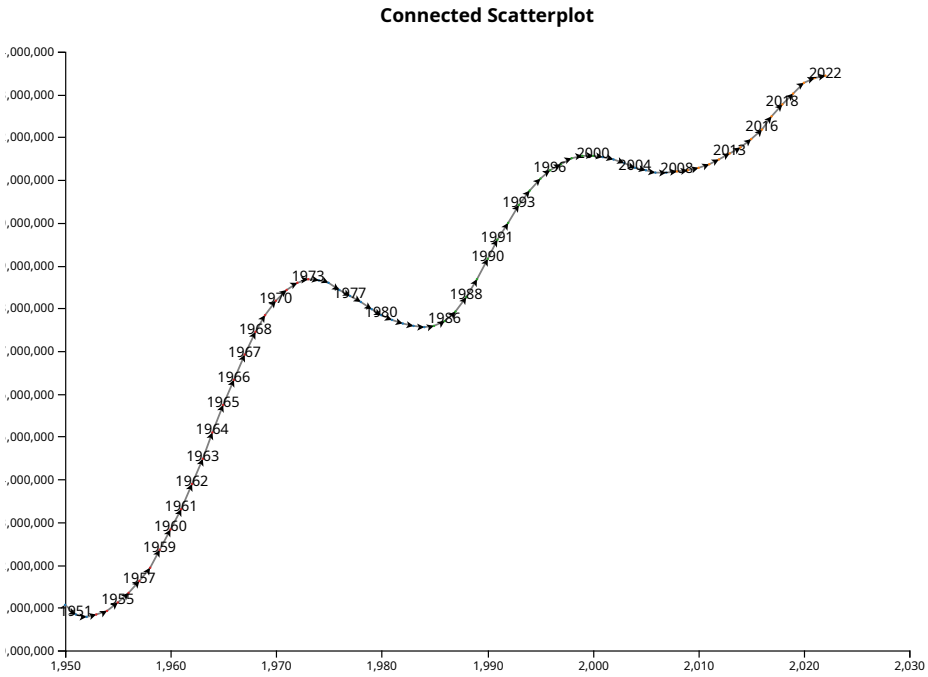

Figure 4.5: Connected Scatterplot: Population change in Germany from 1950 to 2022. [Exported from RAWGraphs by Laura Pessl.]

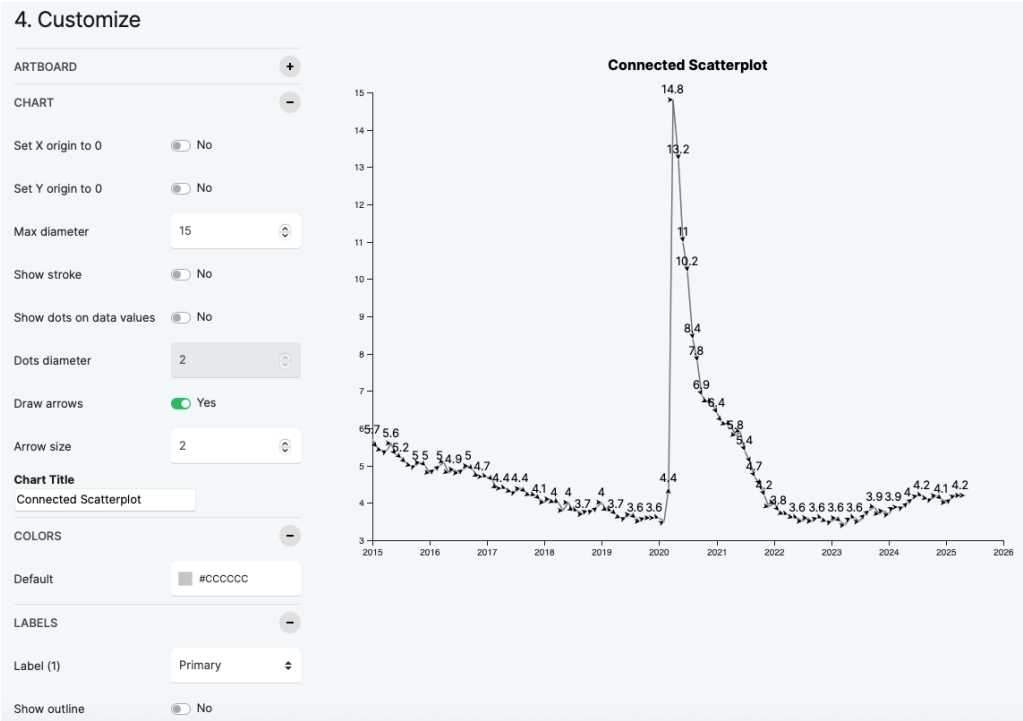

Figure 4.6: Connected Scatterplot: Customisation settings for chart showing the unemployment rate in the US from 2015 to 2025. [Screenshot of RAWGraphs by Laura Pessl.]

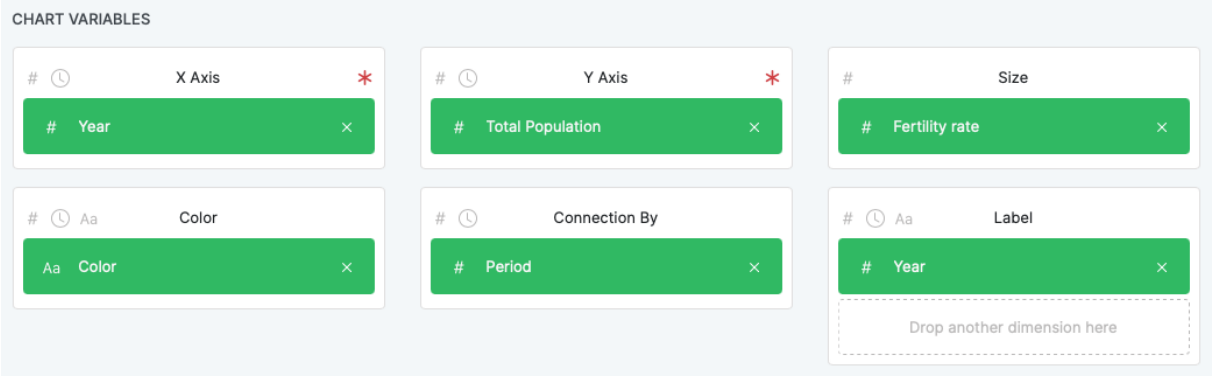

Figure 4.7: Connected Scatterplot: Mapping of dimensions to chart variables on RAWGraphs for `germany-population-growth.csv`. [Screenshot of RAWGraphs by Laura Pessl.]

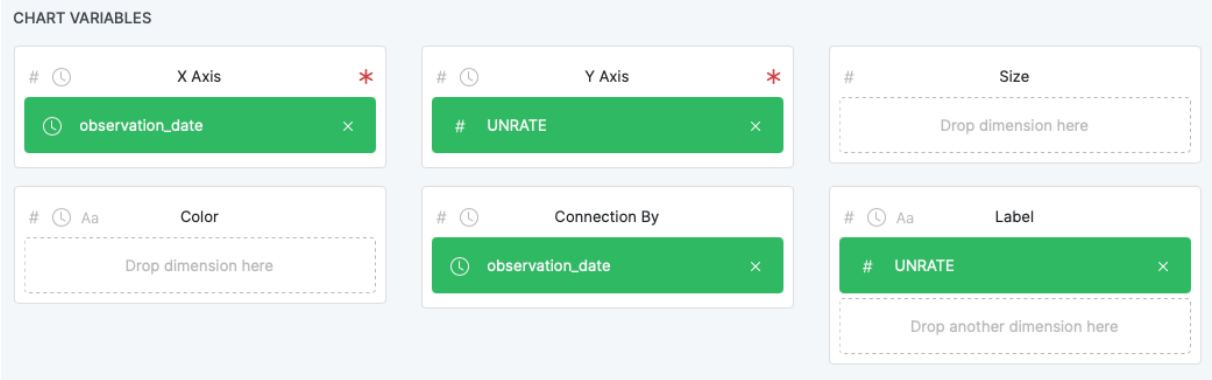

Figure 4.8: Connected Scatterplot: Mapping of dimensions to chart variables on RAWGraphs for `unemployment-rate.csv`. [Screenshot of RAWGraphs by Laura Pessl.]

4.2.3 Customisation

The customisation settings for the connected scatterplot are:

- Chart: Axis, stroke, arrow settings, title.
- Colors: Colour settings.
- Labels: Label and outline settings.

Some of them can be seen in Figure 4.6.

4.3 Polar Area Chart

A polar area chart (or polar area diagram, or rose chart) displays data similar to a pie chart where each wedge usually represents a time period and the segments inside the wedges visualise different categories. The radius of each category segment corresponds to its value. This implementation is based on the drawing of Nightingale [2025] from 1856. The order of the category segments is determined by computing the total value for each category. The category with the highest total is placed at the bottom, while the one with the lowest total forms the top layer.

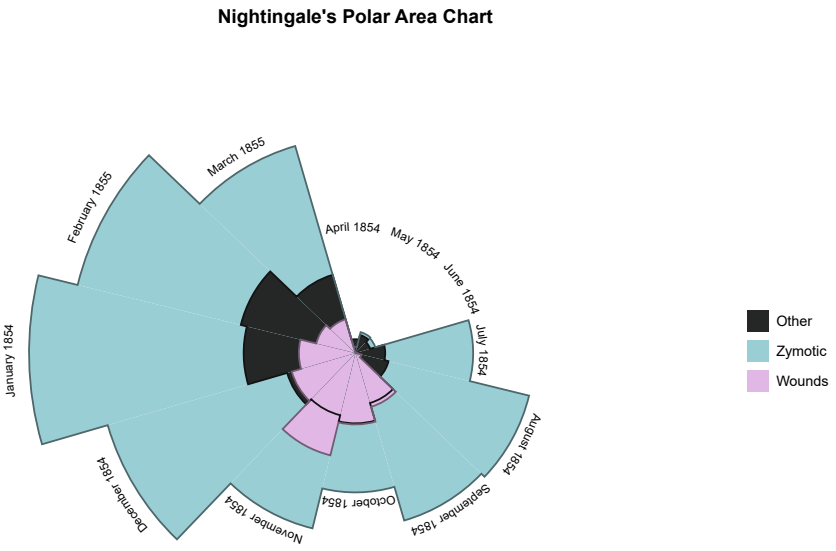

Figure 4.9: Polar Area Chart: The original Nightingale dataset of causes of mortality from April 1854 to May 1855. [Exported from RAWGraphs by Laura Pessl.]

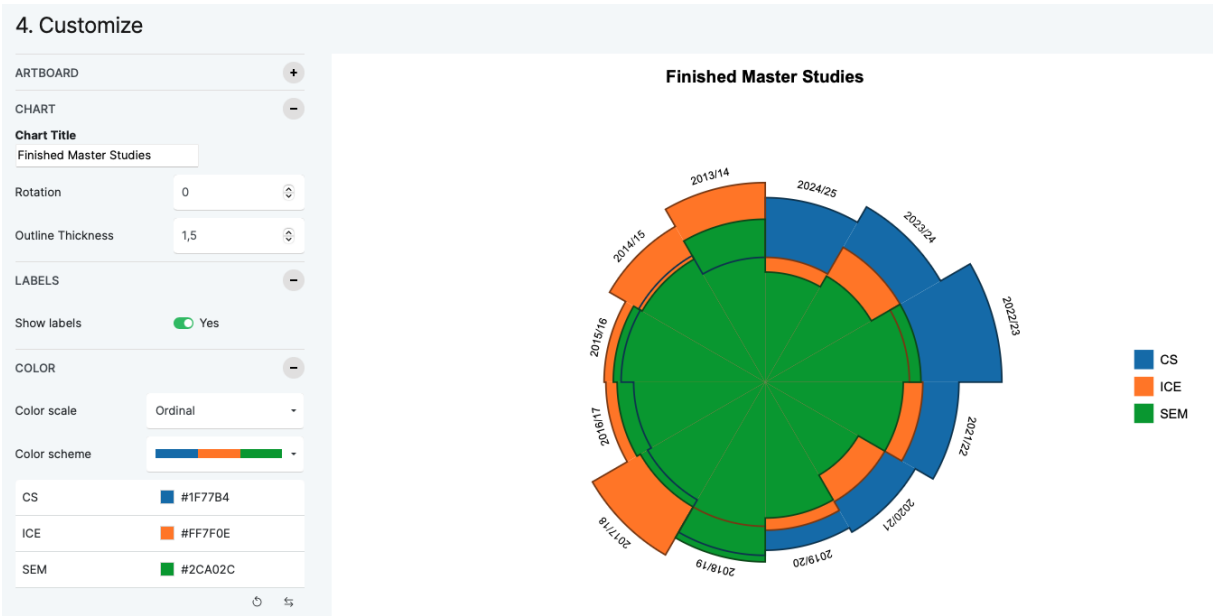

Figure 4.10: Polar Area Chart: Customisation settings for a chart showing finished master studies from 2013 to 2025. [Exported from RAWGraphs by Laura Pessl.]

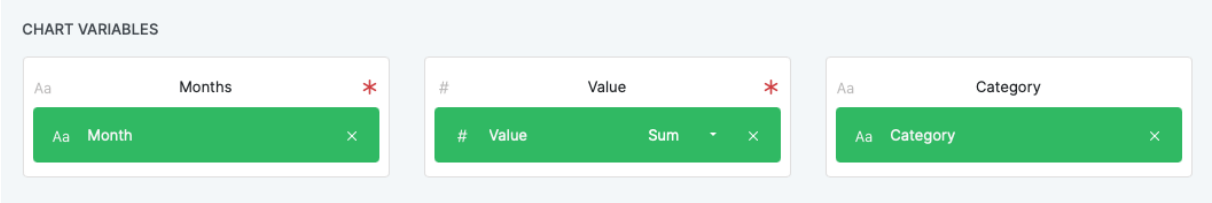

Figure 4.11: Polar Area Chart: Mapping of chart variables for dataset *nightingale.csv*. [Screenshot of RAWGraphs by Laura Pessl.]

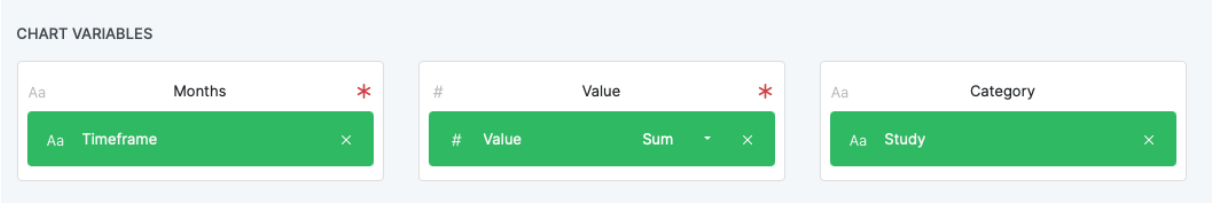

Figure 4.12: Polar Area Chart: Mapping of chart variables for dataset *master-studies.csv*. [Screenshot of RAWGraphs by Laura Pessl.]

4.3.1 Datasets

Besides the original Nightingale dataset `nightingale.csv`, a second dataset `master-studies.csv` contains the number of finished master studies in three degree programmes at Graz University of Technology from 2013 to 2025. Figure 4.9 shows the original Nightingale dataset, whereas Figure 4.10 shows the finished master studies dataset.

4.3.2 Mapping

The polar area chart requires the two mandatory chart variables Months and Value. For colouring the wedges, the optional Category variable can be mapped. Figures 4.11 and 4.12 show how the dimensions need to be mapped in RAWGraphs for the provided datasets.

4.3.3 Customisation

The additional customisation options for the polar area chart are:

- Chart: Title, rotation, outline settings.
- Label: Label settings.
- Color: Colour settings.

Some of them can be seen in Figure 4.10.

4.4 Similarity Map

A similarity map is a 2d plot for multidimensional datasets, which places similar items closer together and dissimilar ones farther apart. It helps to reveal patterns, clusters, or relationships in complex datasets, often using projection methods like PCA or t-SNE. This chart was originally implemented by Aumüller et al. [2024]. Some software bugs concerning the correct parsing of data were fixed in this project. A start was made on implementing force directed placement as a further projection method, but it could not be finished in the scope of this project.

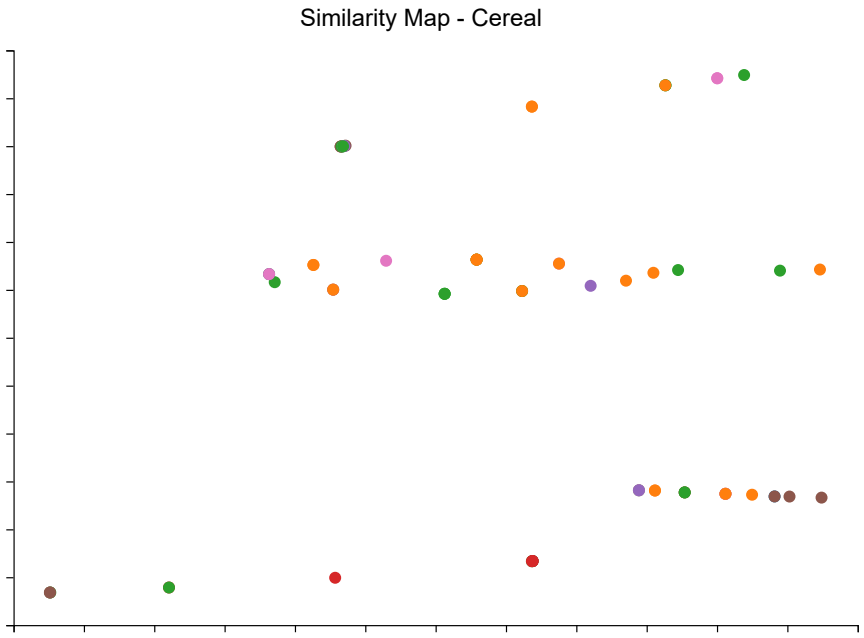

Figure 4.13: Similarity Map: Similarity of cereals using t-SNE projection. [Exported from RAWGraphs by Laura Pessl.]

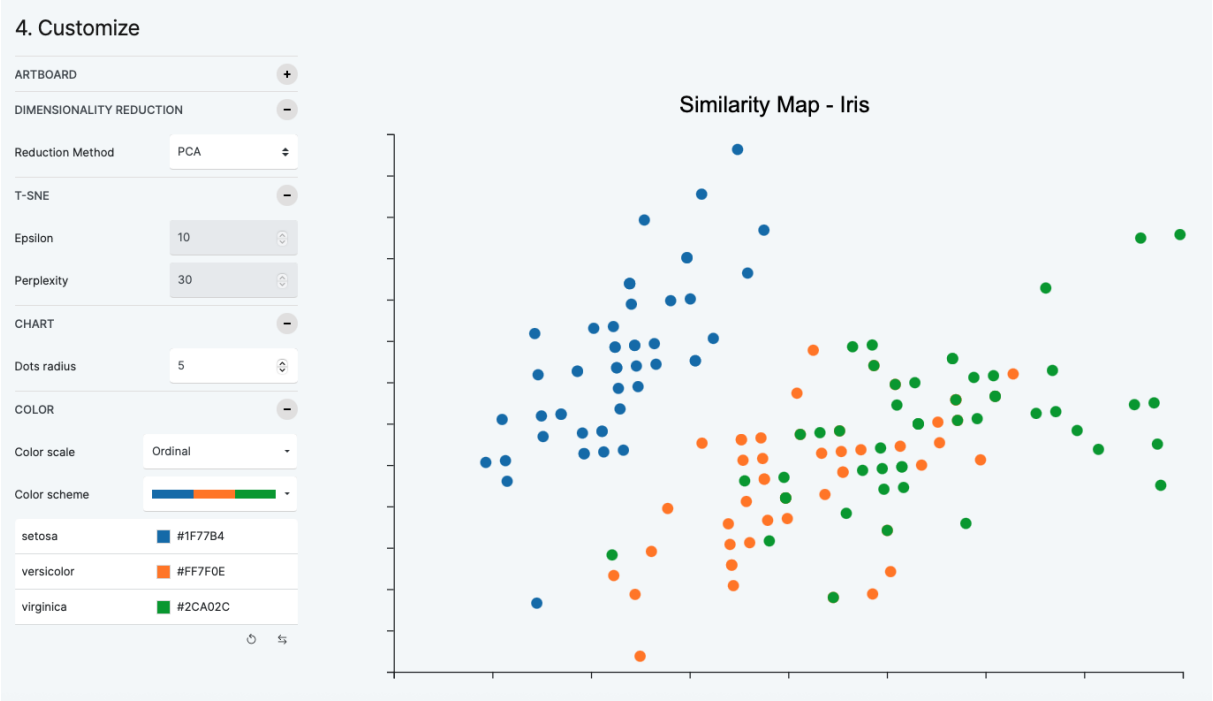

Figure 4.14: Similarity Map: Customisation settings for similarity map of the `iris.csv` dataset using PCA projection. [Screenshot of RAWGraphs by Laura Pessl.]

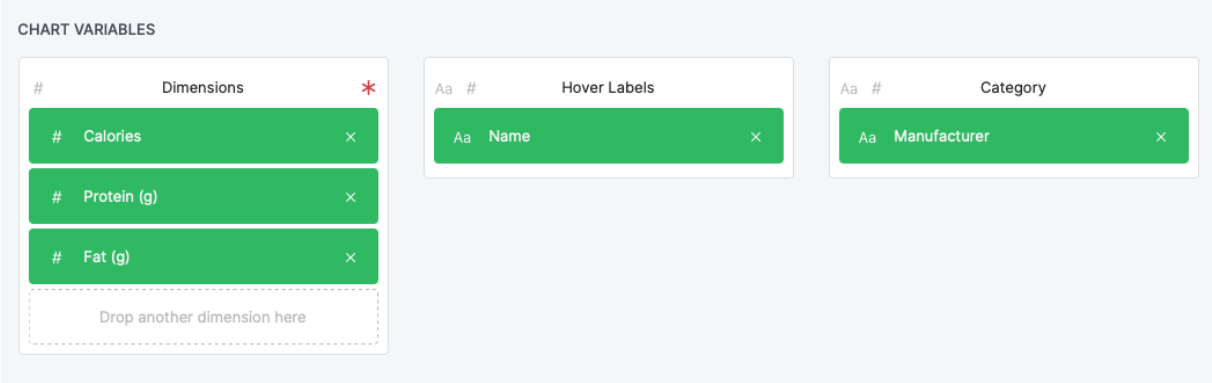

Figure 4.15: Similarity Map: Mapping of dimensions for `cereals.csv` dataset. [Screenshot of RAWGraphs by Laura Pessl.]

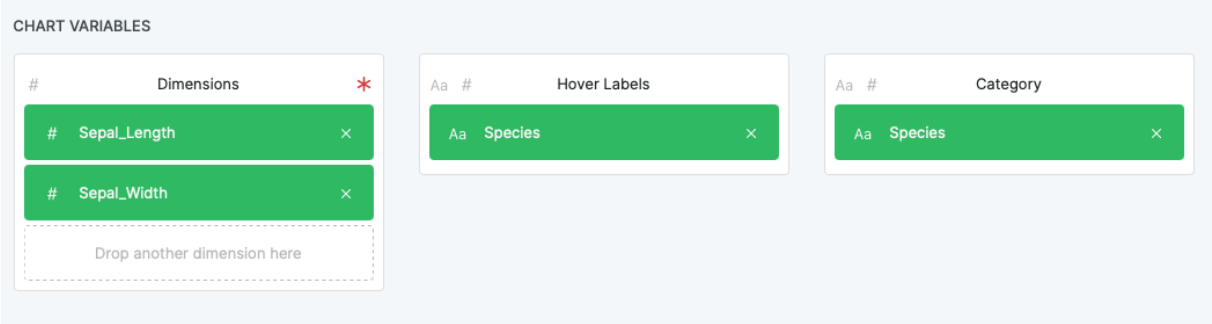

Figure 4.16: Similarity Map: Mapping of dimensions for `iris.csv` dataset. [Screenshot of RAWGraphs by Laura Pessl.]

4.4.1 Datasets

Two datasets are provided as examples to be used with the similarity map: `cereals.csv` contains nutrition data about cereals from different manufacturers, and `iris.csv` contains data about different species of iris flower, including their sepal and petal characteristics. They are shown in Figures 4.13 and 4.14, respectively.

4.4.2 Mapping

There are three chart variables for the similarity map that can be mapped. The Dimensions variable is mandatory and multiple dimensions can be assigned to it. Hover Labels and Category are optional variables. The recommended mappings can be seen in Figure 4.15 and Figure 4.16.

4.4.3 Customisation

The additional customisation settings for the similarity map are:

- Dimensionality Reduction: Selection of the projection (reduction) method: PCA, UMAP, t-SNE, or FDP.
- T-SNE: Further settings for t-SNE method.
- Chart: Dots settings.
- Color: Colour settings.

Some of them can be seen in Figure 4.14.

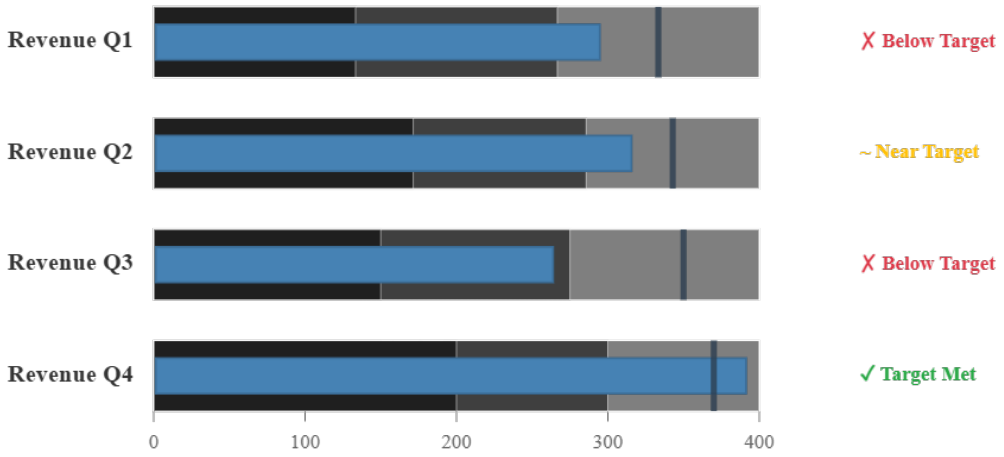

Figure 4.17: Bullet Chart: Quarterly revenue performance against set targets. [Exported from RAWGraphs by Vera Tysheva.]

4.5 Bullet Chart

A bullet chart is a variation of a traditional bar chart, developed by Stephen Few in 2006 [Few 2013], to serve as a more compact and informative visual representation of data. Designed for dashboards and performance metrics, bullet charts efficiently display a single, quantitative measure and provide contextual information in a compact space, without distracting decorations.

A bullet chart typically consists of the following elements:

- *Main Bar (Measure)*: A horizontal bar representing the current value or performance metric (e.g., revenue, sales, score).
- *Comparative Marker*: A vertical line or symbol indicating a target, goal, or benchmark value.
- *Qualitative Ranges*: Shaded background bands representing performance categories (e.g. poor, satisfactory, and good).

These elements allow viewers to easily compare the current performance to predefined targets within a qualitative context.

4.5.1 Dataset

Only one example dataset is provided for bullet charts: `revenues.csv` contains information about revenues and targets of a fictitious company. The corresponding bullet chart can be seen in Figure 4.17

4.5.2 Mapping

The bullet chart has four chart variables, to which data dimensions can be mapped: Current Value, Target Value, Qualitative Ranges, and Chart Titles. Figure 4.18 shows the intended mapping for the provided dataset.

4.5.3 Customisation

The additional customisation options for the bullet chart are:

- Chart: Colours, values, axis ticks, global scale, performance indicator.

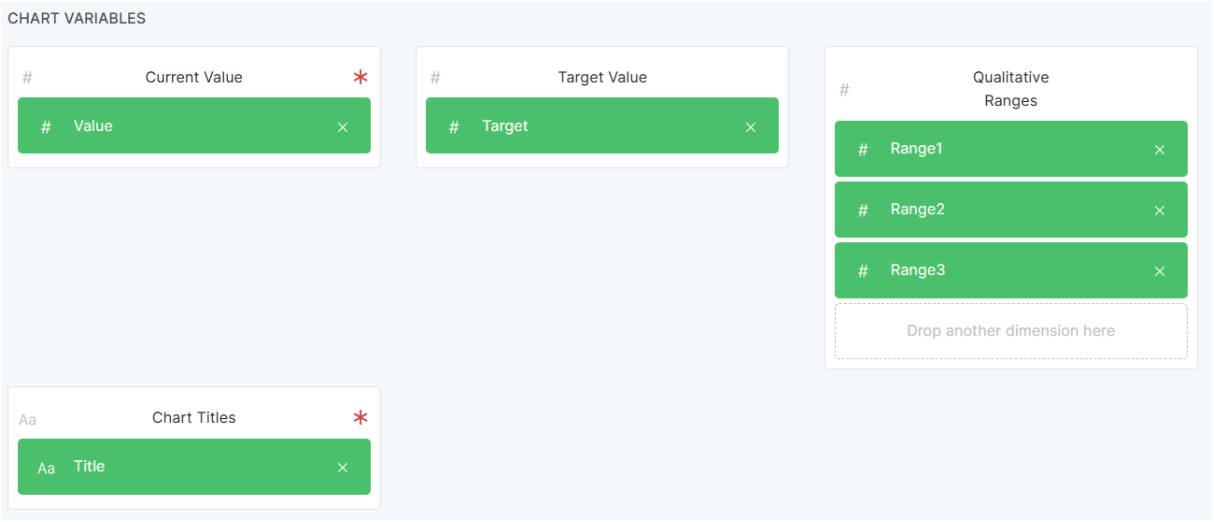

Figure 4.18: Bullet Chart: Mapping of dimensions to chart variables for `revenues.csv` dataset. [Screenshot of RAWGraphs by Vera Tysheva.]

4.6 Pareto Chart

A Pareto chart is a combination of a bar chart and a line chart. The bars represent individual values in descending order, while the line shows the cumulative percentage of the total. A Pareto chart is especially useful in quality control and decision-making processes. Figure 4.19 shows a Pareto chart visualising the number of complaints in various categories.

4.6.1 Datasets

Two example datasets are provided for Pareto charts: `complaints.csv` contains data about the reaons for complaints and how often they appear, and `km.csv` contains the average kilometres per hour for different types of transport such as walking, cycling, or driving. The first dataset is shown in both Figures 4.19 and 4.20.

4.6.2 Mapping

A Pareto chart has two chart variables to which dimensions can be mapped: Category and Values. Figures 4.21 and 4.22 show the intended mappings for the provided datasets.

4.6.3 Customisation

The additional customisation options for the Pareto chart are:

- Axis: Axis labels, label rotations, axis label visibility.
- Chart: Padding between bars, sorting order, cumulative line visibility.
- Color: Bar colour, cumulative line colour.

Some of them can be seen in Figure 4.20.

4.7 Other Potential Charts

Various other potential chart options were considered, including:

- *Waffle Chart*: There are typically 100 cells, each cell representing one percent of the whole [DVP 2025b].

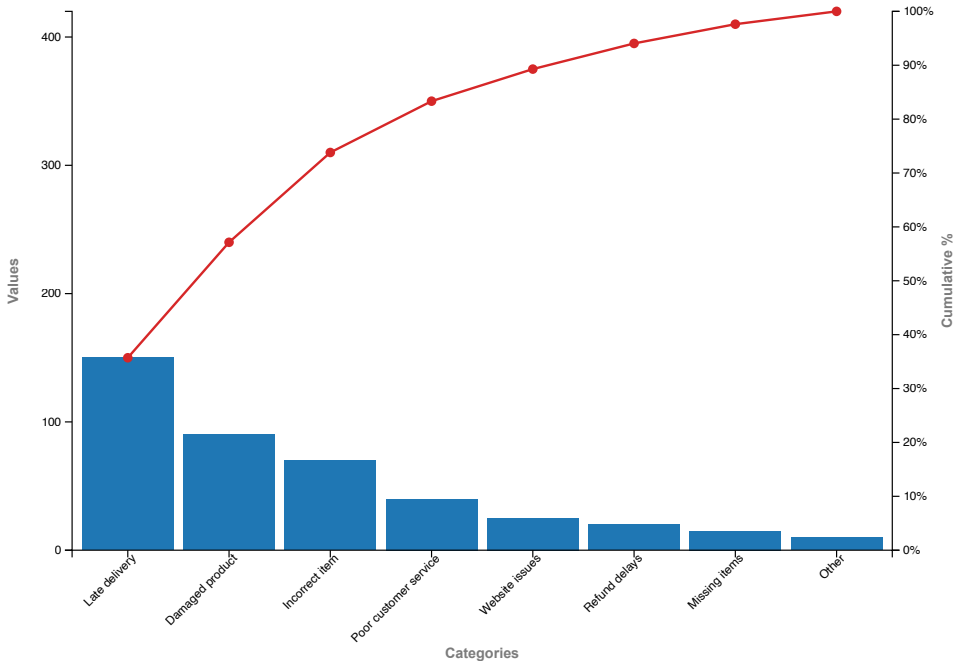

Figure 4.19: Pareto Chart: Reasons for complaints and their cumulative total. [Exported from RAWGraphs by Bastian Kandlbauer.]

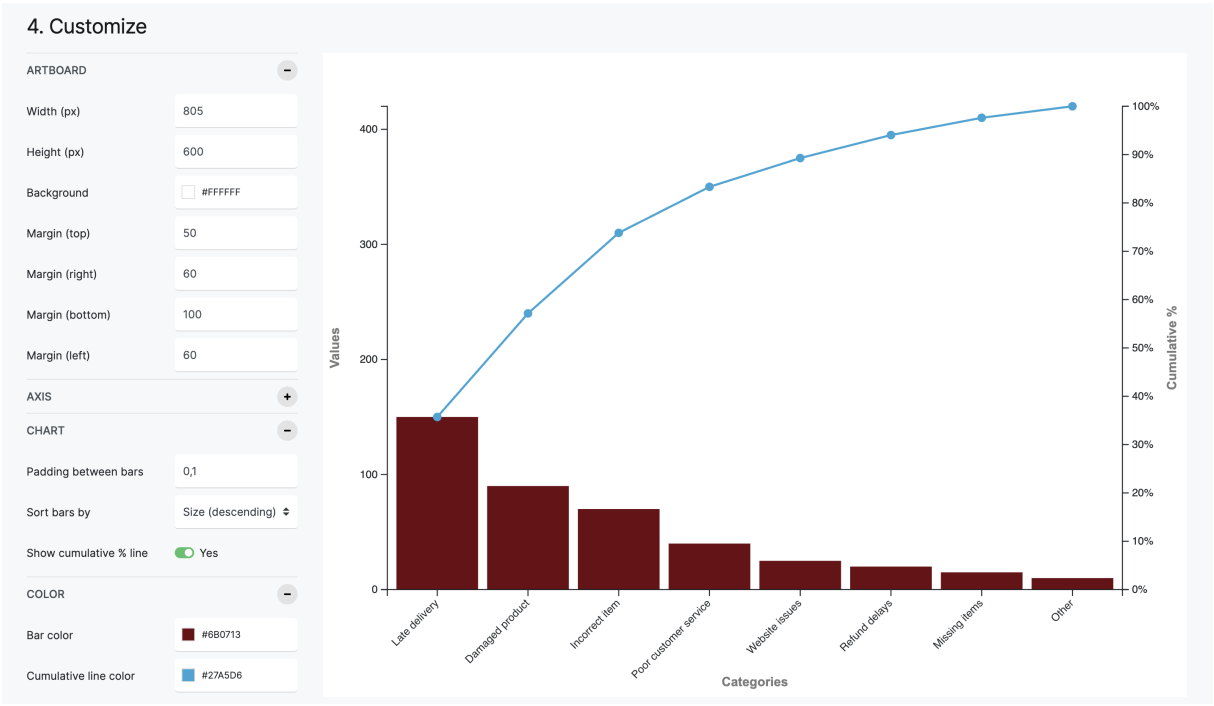

Figure 4.20: Pareto Chart: Customisation settings for Pareto chart of reasons for complaints. [Screenshot of RAWGraphs by Bastian Kandlbauer.]

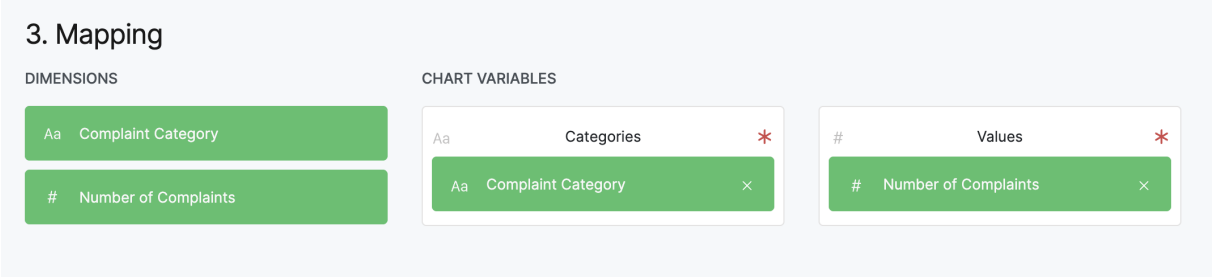

Figure 4.21: Pareto Chart: Mapping of dimensions for the `complaints.csv` dataset. [Screenshot of RAWGraphs by Bastian Kandlbauer.]

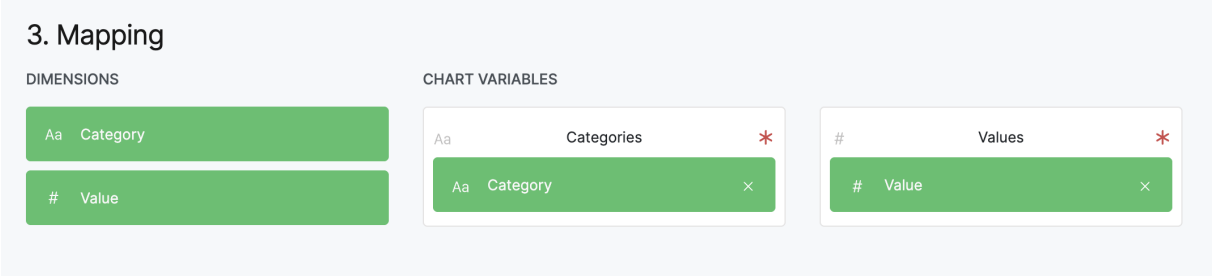

Figure 4.22: Pareto Chart: Mapping of dimensions for the `km.csv` dataset. [Screenshot of RAWGraphs by Bastian Kandlbauer.]

- *Jittered Strip Plot*: Displays distribution of individual one-dimensional values [DVP 2025a].
- *Force-Directed Tree*: Shows connections between objects in a hierarchy [amCharts 2025].

but these were not followed further.

Chapter 5

Concluding Remarks

This project collected six different custom charts for RAWGraphs into a suite called the HCC Custom Charts. Four of the charts built upon existing implementations from previous groups of students, which were updated and improved. Two of the charts were built from scratch.

The RAWGraphs tutorials provided clear guidance, making the creation of new charts straightforward. One limitation encountered was that certain customisation options in RAWGraphs cannot be controlled by the chart developer. This particularly affected the generation of the underlaying SVG, since charts are automatically placed inside a fixed-size SVG `<rect>` element. This restricts the flexibility and scalability of the SVG image and goes against best practices. Furthermore, uploading datasets to RAWGraphs, such as `cities.csv` dataset used by previous groups, can lead to issues, since RAWGraphs sometimes interprets numeric values as dates by default, meaning the end user has to spot the issue and manually override the type of the data column.

Bibliography

amCharts [2025]. *Force-Directed Tree*. 30 Jun 2025. https://amcharts.com/demos/force-directed-network/ (cited on page 21).

Aumüller, Thomas, Martin Heider and Abdelrahman Ramadan [2024]. *Custom Charts for RAWGraphs*. Project Report. 706.057 Information Visualisation SS 2024. Graz University of Technology, Austria, 07 Aug 2024. 36 pages. https://courses.isds.tugraz.at/ivis/projects/ss2024/ivis-ss2024-g1-project-raw-custom-charts.pdf (cited on pages 9, 11, 15).

DensityDesign, Calibro and Inmagik [2025]. *RAWGraphs*. 30 Jun 2025. https://rawgraphs.io/ (cited on pages 1, 4–5).

DVP [2025a]. *Jittered Strip Plot*. Data Viz Project, 30 Jun 2025. https://datavizproject.com/data-type/jitter-plot/ (cited on page 21).

DVP [2025b]. *Waffle Chart*. Data Viz Project, 30 Jun 2025. https://datavizproject.com/data-type/percentage-grid/ (cited on page 19).

Few, Stephen [2013]. *Bullet Graph Design Specification*. 10 Oct 2013. https://perceptualedge.com/articles/misc/Bullet_Graph_Design_Spec.pdf (cited on page 18).

Kandlbauer, Bastian, Vera Tysheva and Laura Pessl [2025]. *HCC Custom Charts Repository*. 30 Jun 2025. https://github.com/laurapessl/hcc-custom-charts (cited on page 9).

Mauri, Michele, Tommaso Elli, Giorgio Caviglia, Giorgio Uboldi and Matteo Azzi [2017]. *RAWGraphs: A Visualisation Platform to Create Open Outputs*. Proc. 12th Biannual Conference of Italian SIGCHI (CHItaly 2017) (Cagliari, Italy). 18 Sep 2017. doi:10.1145/3125571.3125585. https://researchgate.net/publication/319285617_RAWGraphs_A_Visualisation_Platform_to_Create_Open_Outputs (cited on page 1).

Nightingale, Florence [2025]. *Diagram of the causes of mortality*. 30 Jun 2025. https://commons.wikimedia.org/wiki/File:Nightingale-mortality.jpg (cited on page 13).

RAWGraphs [2025]. *RAWGraphs Custom Chart Template*. 30 Jun 2025. https://github.com/rawgraphs/custom-rawcharts-template/ (cited on page 7).