Data drawing

Group 5

Karmen Gostiša
Anastasiia Klimashevskaia
Jaka Konda
Günther Moser

Institute of Interactive Systems and Data Science (ISDS),
Graz University of Technology
A-8010 Graz, Austria

27 May 2019

Abstract

This survey paper is a review and comparison of the Data-Driven Drawing instruments. Feature descriptions, functionality and usability are considered for each, with various test cases determining the capabilities of the tools. The advantages and disadvantages of different approaches are discussed and conclusions about use cases of the Data-Driven Tools are drawn.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The amount of data circulating around every day is tremendous and people have formed a need for proper visualisations of it for better grasping and understanding. In the modern world everything is done way easier digitally using various technologies.

Over the past years a lot of different programming languages and libraries emerged, allowing the users to create numerous sorts of different visualisations, using the expressive power of the tools to the full extent. For example, D3.js [Bostock 2011] with the help of JavaScript knowledge gives a wide range of opportunities for creating different static and interactive visualisations. Another example is Processing [Casey Reas 2001], a whole new programming language aimed on visualising what the user wants. However, there is a keyword here that can be observed in any of such cases - language. The user has to know the programming language to be able to use it fully and confidently. It is not enough to just grasp the basics of the language to create exactly what you want without wasting time on searching for tutorials and examples, that might not even exist. Learning programming languages can take a long time, and the learning curve might be pretty steep. Another opportunity would be using tools like MS Excel [Microsoft 1987] or Libre Office Spreadsheet [Division 2011] with the built-in plotting functions for visualising the data loaded directly in the program. It is easier to manipulate the data and change parameters indeed, and no programming language knowledge is needed, only experience in using the tool itself, what is definitely not as complex as a programming language. Nevertheless, these tools are always lacking one important quality - expressiveness. Mainly the user is supplied with only standard templates of scatter plots, line plots and diagrams. In case the user needs something more sophisticated, personalised and unique, then this is not the right choice.

If expressiveness is the main required feature, then another option are vector graphics editors like Adobe Illustrator [Adobe 2016] or InkScape [Inkscape 2016]. Here the user has as much freedom as needed, however, these tools also require deep studying and a lot of experience of using it, what can take months or sometimes even years. If the user is not familiar with such graphic instruments it would be time-consuming and cumbersome to get to know the tool enough to start creating something they need. Moreover, all the data-binding would have to be done manually by modifying each of the objects in the chart to represent the corresponding piece of data, what could be a tremendous work.

Thus here comes the demand for an instrument combining the positive features of the approaches described above and avoiding the difficulties and drawbacks of each. The technique of Data-Driven Drawing was proposed - vector drawing tools with a possibility to bind certain object properties to the data supplied automatically, however, still being able to customise it and modify as the user wants. In this paper several tool adopting this approach are reviewed and tested, with their functionality being compared afterwards. The original list of the tools selected for the review is:

- Lyra

- Data Illustrator
- Charticulator
- Hyecoo
- iVisDesigner
- DataInk
- Power BI

1.2 Overview of survey structure

The paper is organised as follows: each tool is reviewed in detail first separately, discussing the interface, options and functionality of each tool with a general opinion and conclusion about the tool. Afterwards, the test cases used for trying out the tools are described and explained, with the further results of the attempts to recreate the visualisations of the test data sets. Lastly, the conclusion about the discussed tools is drawn at the end of the paper.

Chapter 2

Tools

The main focus of this survey is to provide an overview of tools for Data Drawing. In this Chapter, each tool is described to the extent of understanding its main features and capabilities. The following points are presented for each tool:

- a general description;
- the user interface description;
- the main features;
- import and export options;
- a subjective assessment.

2.1 Charticulator

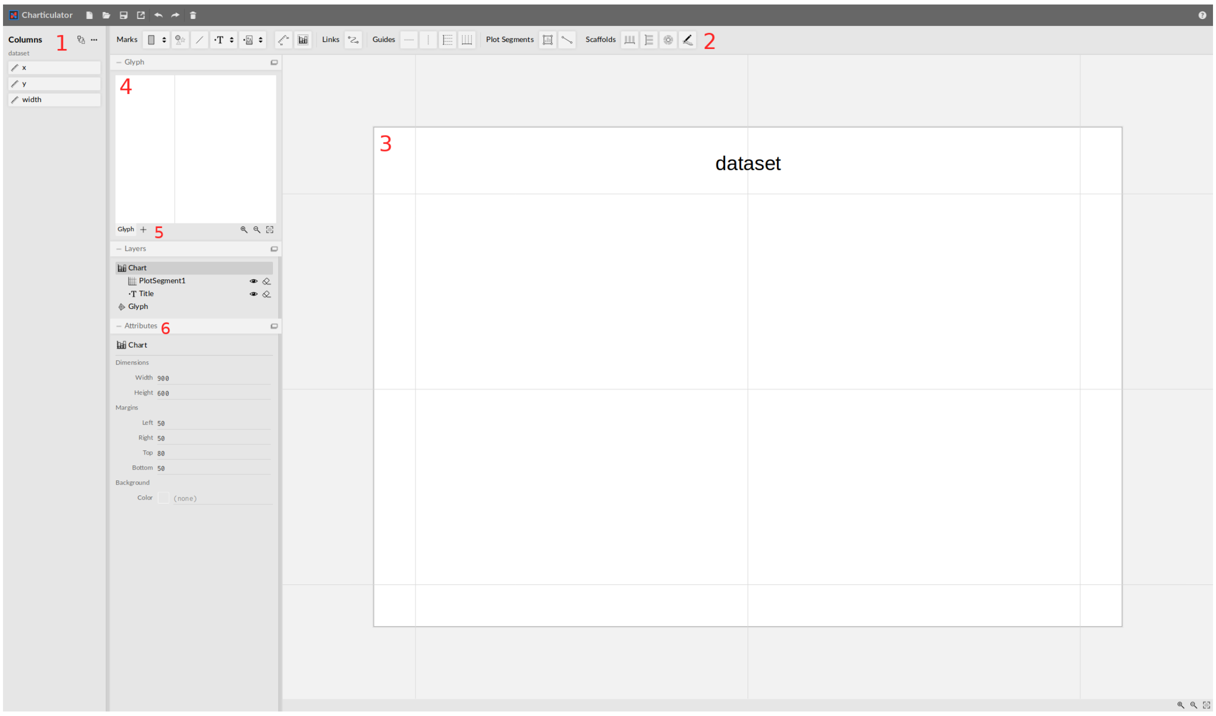
Charticulator [Ren et al. 2019] is a web-based Data Drawing tool that enables the user to draw custom data visualisations using “lazy data binding". It is developed and maintained by Microsoft Research. Hosted version is available as well as source code for self-hosted option using Node.js and Yarn. It is licensed under MIT license .

2.1.1 User interface

Charticulator’s user interface consists of six parts (see Figure 2.1. On the left side of the tool there is a Dataset Panel (1), listing the columns from loaded dataset. The top toolbar (2) contains widgets for specifying Marks, Links, Guides, Plot Segments, and Scaffolds. The chart canvas is in the white centre section (3). Glyph Editor (4), Layers Panel (5) and Attributes Panel (6) are next to the import section. Those can also be shown as a separate window using pop-ups. The data can be bound to x and y axis using drag and drop or the “bind" symbol from the Attributes Panel.

2.1.2 Features

The drawings are built with Glyphs. These are composed out of marks (rectangles, line, etc.) and layout constraints in the smaller top left canvas. Each plot segments has one glyph bound to the data points. The User can "overlay" several plots and draw Glyphs with different data bindings to stack several drawings together to the create the desired output.



Dimensions

Width 900

Height 600

Margins

Left 50

Right 50

Top 80

Bottom 50

Background

Color (none)

Figure 2.1: Screenshot of the Charticulator interface.

Layout composition

Charitucultor supports several different different layout compositions. By default it uses rectangular plot with Cartesian coordinate system. The user can choose between different layout compositions such as stacking by clicking on the plot and select the desired orientation. The stack operation is performed by automatically discretizing numerical or categorical value of an attribute value and afterwards placing Glyphs on top of each other inside the created groups. This way it is possible to use a combination like:

- **Y/X**: Vertical Scaffold - Horizontal Scaffold / H. Categorical Axis / H. Numerical Axis.
- **Y/X**: V. Categorical Axis - Horizontal Scaffold / H. Categorical Axis / H. Numerical Axis.
- **Y/X**: V. Numerical Axis - Horizontal Scaffold / H. Categorical Axis / H. Numerical Axis.

Text formatting

Text formatting is very simple. The format for the values follows the d3's syntax. E.g. ${avg(Width)}.{.1f} means use the average value of the numerical Value Width. By removing or editing the second part the user can change the number past the decimal point.

2.1.3 Import and export options

Charticulator has the option to import datasets in CSV format with the prerequisite of first row containing the column names. After the tool is launched, user can import data in the *New chart* panel under “Data" or “Links" section by clicking the button or dragging and dropping a file into the area to open the data file.

Exporting charts can be done in one of the following ways:

- **Export as Image**: PNG, JPEG or SVG. For the PNG and JPEG, desired DPI (dots per inch) can also be defined.
- **Export as HTML**: SVG together with several libraries is embedded in HTML and exported.

- **Export as Charticulator Template**: User can define axes and scales or change the exposed properties before saving the whole project as a template. The template can then be used as a chart component to create a nested chart.
- **Export as Power BI Custom Visual**: this option is similar to the Charticulator Template export option only to be used with Power BI.

2.1.4 Subjective assessment

An option to load multiple data sets would be a great addition or fix the link option they provide. Overall Charticulator is a very powerful and stable tool. The GUI is user-friendly and easy to understand, giving the user a lot of flexibility. The possibility to install and run the tool locally is also a very convenient feature.

2.2 Data Illustrator

Data Illustrator [Illustrator 2019a] is a web tool, implementing interactive techniques in modern vector design tool for direct manipulation of visualisation configurations and parameters. It was launched in 2018 by Adobe in a collaboration with the Georgia Institute of Technology [Liu et al. 2018]. As of April 2019 it is a research prototype and not an Adobe product. It is proprietary tool so the user has no right to copy it without a license from the owner. The current version 1.1 was released on May 30, 2018 and tested with Google Chrome.

2.2.1 User interface

Data Illustrator's interface is composed of eight components (see Figure 2.2):

1. **Left Toolbar**: selection and drawing tools;
2. **Upper Toolbar**: menu bar with command buttons, appearing in the drop-down list;
3. **Data Variables Panel**: data import and summary;
4. **Layers Panel**: list of shapes and their organisation in the canvas;
5. **Canvas**: place to select, draw and manipulate shapes;
6. **Data Table Panel**: data shown as a table;
7. **Right Toolbar**: align tools and data operators;
8. **Property Inspector**: properties of selected items and controls to bind data to items.

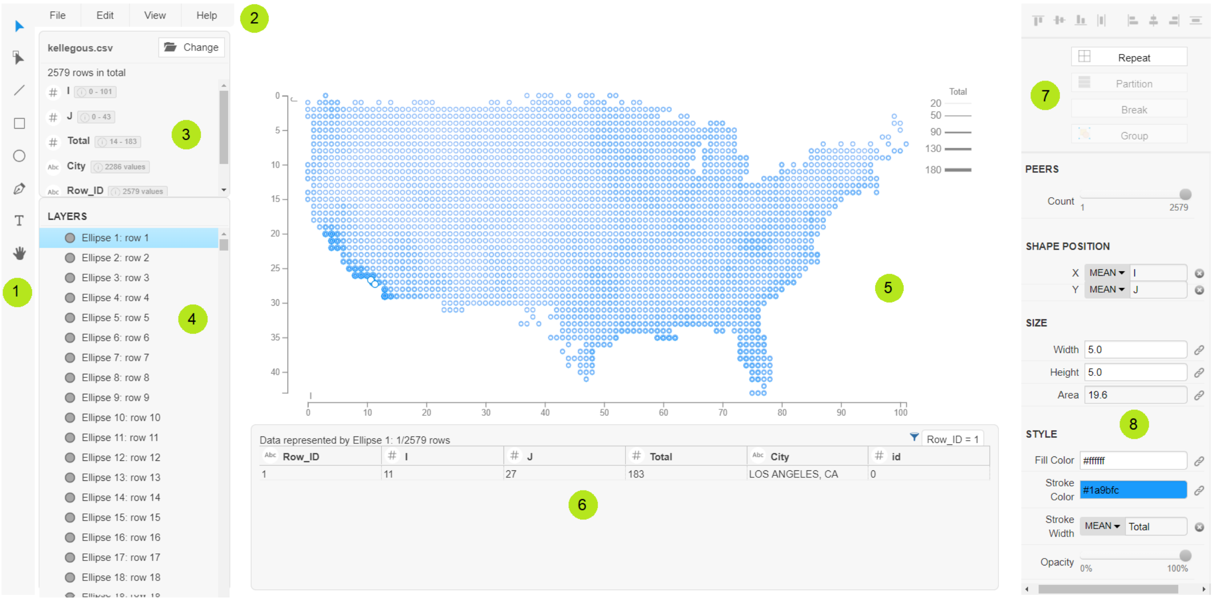
2.2.2 Features

Drawing and manipulating shapes

There are three shapes available: *rectangle*, *ellipse* and *line*. User can draw them to desired size or resize them later by dragging one of the anchor handles. It is also possible to add a *text* and *freeform path* on the canvas.

Shape is represented as a path that consists of a series of anchor points connected by line or curve segments. Shape can be selected with the Selection Tool and an anchor point or a segment with the Direct Selection Tool. When the user makes a selection, the associated UI with the properties of selected item appears.

Shapes can be grouped and aligned in all directions. The tool also provides zooming and panning functionalities.



kellegous.csv

2579 rows in total

- # I 0 - 101
- # J 0 - 43
- # Total 14 - 183
- Abc City 2286 values
- Abc Row_ID 2579 values

- Ellipse 1: row 1
- Ellipse 2: row 2
- Ellipse 3: row 3
- Ellipse 4: row 4
- Ellipse 5: row 5
- Ellipse 6: row 6
- Ellipse 7: row 7
- Ellipse 8: row 8
- Ellipse 9: row 9
- Ellipse 10: row 10
- Ellipse 11: row 11
- Ellipse 12: row 12
- Ellipse 13: row 13
- Ellipse 14: row 14
- Ellipse 15: row 15
- Ellipse 16: row 16
- Ellipse 17: row 17
- Ellipse 18: row 18

Data represented by Ellipse 1: 1/2579 rows

Row_ID = 1

Row_ID	I	J	Total	City	id
1	11	27	183	LOS ANGELES, CA	0

Count 1 2579

X MEAN I

Y MEAN J

Width 5.0

Height 5.0

Area 19.6

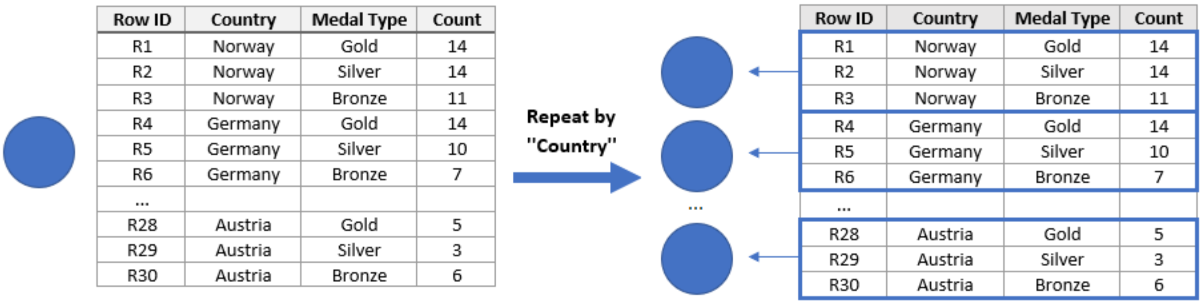
Fill Color #ffffff

Stroke Color #1a9bfc

Stroke Width MEAN Total

Opacity 0% 100%

Figure 2.2: Screenshot of the Data Illustrator's interface consisting of eight components taken on 27 May 2019.



Row ID	Country	Medal Type	Count
R1	Norway	Gold	14
R2	Norway	Silver	14
R3	Norway	Bronze	11
R4	Germany	Gold	14
R5	Germany	Silver	10
R6	Germany	Bronze	7
...			
R28	Austria	Gold	5
R29	Austria	Silver	3
R30	Austria	Bronze	6

Row ID	Country	Medal Type	Count
R1	Norway	Gold	14
R2	Norway	Silver	14
R3	Norway	Bronze	11
R4	Germany	Gold	14
R5	Germany	Silver	10
R6	Germany	Bronze	7
...			
R28	Austria	Gold	5
R29	Austria	Silver	3
R30	Austria	Bronze	6

Figure 2.3: Data Illustrator's repeat action on "Country" from 2018 Winter Olympic medal dataset.

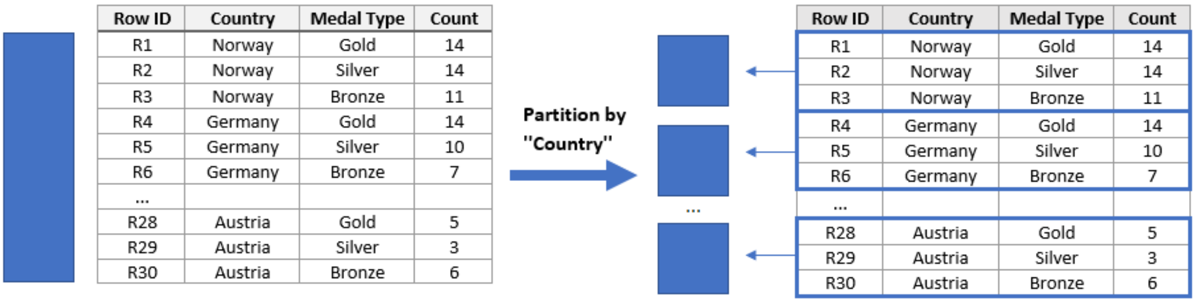
Creating shapes by data

Data Illustrator provides two controls to create shapes by data: repeat and partition. Repeat creates multiple copies of any shape, placing them in a grid layout by default. Partition divides a shape into parts, putting them in a layout that depends on shape type.

For the repeat control user needs to specify a categorical data variable, e.g. "Country" in 2018 Winter Olympic medal dataset (see Figure 2.3). By default, shapes will be put in a grid layout with two handles, allowing to control how many shapes are generated and how they are placed in the layout.

The partition control also requires a categorical data variable to divide the shape and associate each part of the shape with a distinct value and corresponding data rows. For example, using the 2018 Winter Olympic medal dataset, rectangle partition action on the data variable "Country" will divide the rectangle into 20 smaller rectangles as there are 20 distinct "Country" values (see Figure 2.4). By default, rectangles are placed in a stack layout. It is also possible to partition data using the line shape. In this case, the partition action divides the line into smaller line segments, associating each anchor point with a distinct value and corresponding data rows.

Sometimes it is necessary to break the layout as its rigidity prevents the user from create visualisation design. In such cases, the "Break Grid" control is used. Once the layout is removed, the user can freely move the shapes around and for example, bind data to the positions of the shapes. However, once the



Row ID	Country	Medal Type	Count
R1	Norway	Gold	14
R2	Norway	Silver	14
R3	Norway	Bronze	11
R4	Germany	Gold	14
R5	Germany	Silver	10
R6	Germany	Bronze	7
...			
R28	Austria	Gold	5
R29	Austria	Silver	3
R30	Austria	Bronze	6

Row ID	Country	Medal Type	Count
R1	Norway	Gold	14
R2	Norway	Silver	14
R3	Norway	Bronze	11
R4	Germany	Gold	14
R5	Germany	Silver	10
R6	Germany	Bronze	7
...			
R28	Austria	Gold	5
R29	Austria	Silver	3
R30	Austria	Bronze	6

Figure 2.4: Data Illustrator's rectangle partition action on "Country" from 2018 Winter Olympic medal dataset.

layout is broken, it can not be re-applied anymore.

Collections, peers and groups

All shapes generated by a repeat or partition control form a *collection*. It is possible to select the collection as well as a shape inside a collection. Manipulating a shape will result in manipulating all shapes inside the collection, as they are *peers* of each other.

Although the terms "group" and "collection" sound similar, they are two distinct concepts. *A group* may consist of shapes that are different types and they may or may not be data bound, whereas each shape inside *a collection* always represents a unique value of the chosen categorical variable.

Binding data

Data Illustrator supports drawing dynamic visualisations, meaning data can be automatically bound to visual properties of shapes, anchor points, segments and collections. If a property can be bound to data, the list of all applicable variables (column names) is shown under the drop-down menu of the corresponding property. Binding is a constraint as once it is created, the property can not be manipulated anymore.

Data binding can be applied to the following shape properties:

- **Size**: width, height and area;
- **Position**: x and y position;
- **Colour**: fill colour, stroke colour and stroke width.

When the user binds a numerical data to a shape's size or position, one or more numerical axes are created. The number of axes depends on whether the shape is inside a collection. The range of scale can be adjusted by dragging the range control handle. Similarly, categorical axes are created when binding a categorical variable to a shape's size or position.

Data Illustrator provides automated colour mappings for shapes, anchor points and text that can be used to map a set of colours to a categorical attribute (e.g. Figure 2.5) or to map numerical values to a continuous colour gradient. For choosing the colours, the Colour Picker widget is provided.

2.2.3 Import and export options

User can import tabular data formatted as CSV files with the assumption that the first row represents the column names. The tool then parses the data, detects the data type for each column and displays data summary in the data variables panel. Besides importing, user can also choose to open one of the sample datasets.

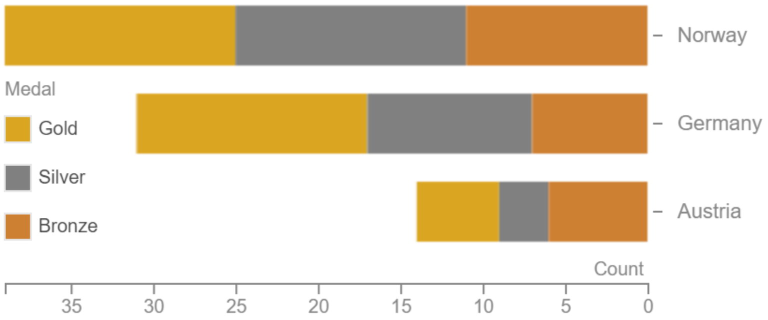


Figure 2.5: Data Illustrator's stacked bar chart using categorical colour mapping on attribute "Medal" from 2018 Winter Olympic medal dataset.

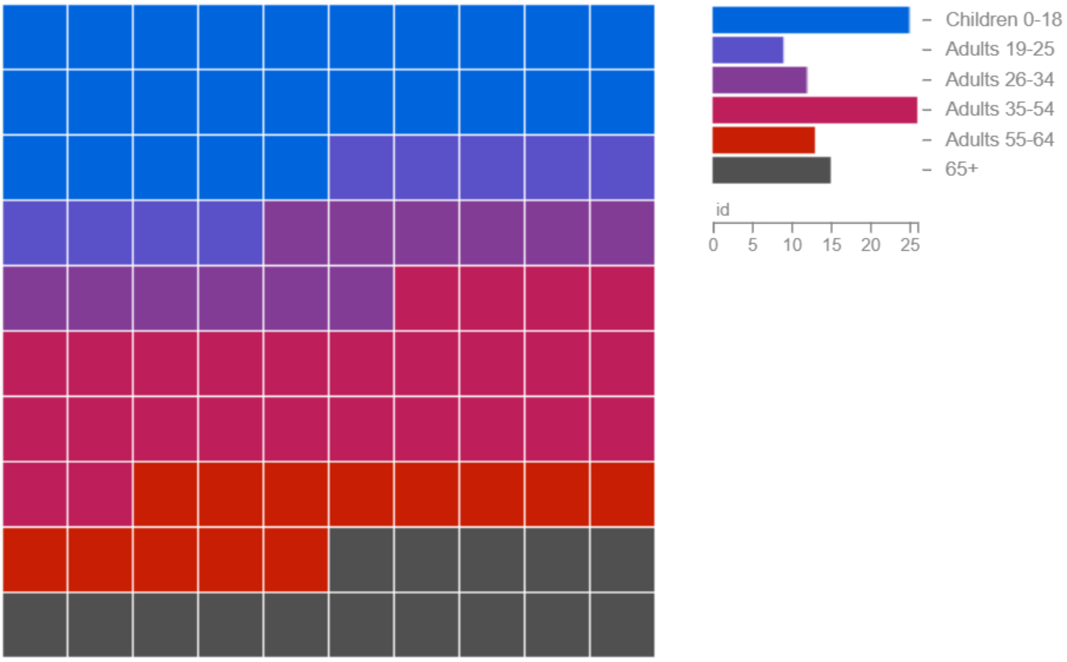


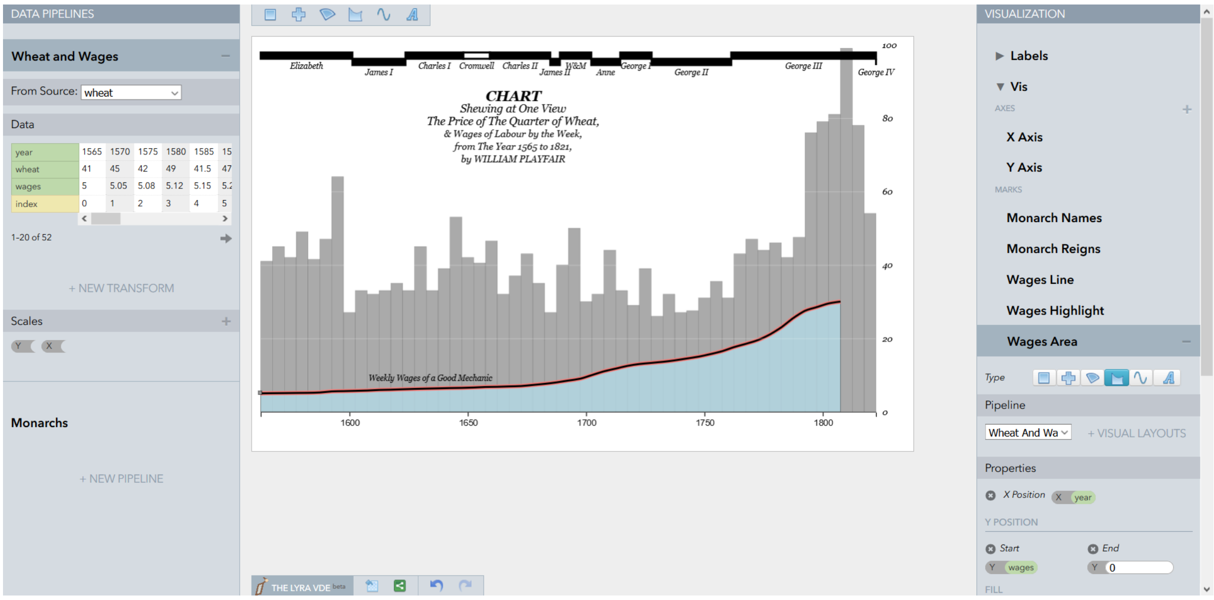
Figure 2.6: Data Illustrator's waffle chart using numerical colour mapping and showing the distribution of population by age group in the U.S. in 2016. Screenshot taken from the tool's official Gallery [Illustrator 2019b].

The tool supports exporting as an SVG file, so the visualisation can be opened in a vector editor tool or embedded in a webpage.

User can also save work on a computer in a "diproj" file and open it later.

2.2.4 Subjective assessment

Data Illustrator is a powerful tool. Once the user learns basic concepts such as repeat and partition action, it is easy to use. The tool has well-written tutorial and provides many sample visualisations along with demo videos showing how to re-create them.



From Source: wheat

year	1565	1570	1575	1580	1585	15
wheat	41	45	42	49	41.5	47
wages	5	5.05	5.08	5.12	5.15	5.2
index	0	1	2	3	4	5

1-20 of 52

Pipeline: Wheat And Wa

X Position: X year

Start: Y wages

End: Y 0

Figure 2.7: Screenshot of the Lyra tool. On the left there is data panel, centre visualisation view and on the right is the layer panel.

2.3 Lyra

Lyra [Satyanarayan and Heer 2014] is an interactive Visualisation Design Environment (VDE) tool that enables users to draw custom visualisations without the need of any programming knowledge. It was developed by University of Washington Interactive Data Lab as a part of the Vega project [Satyanarayan 2019]. The current version of Lyra is no longer supported as Lyra 2 is in the works, but marked as early development and unstable. Developers are currently in the process of re-implementing the feature set. The tool is licensed under BSD-3-Clause. In this section Lyra is reviewed and its functionality described.

2.3.1 User interface

Lyra's user interface is split into three columns, as it can be observed from Figure 2.7. The first column represents all of the data import and manipulation options. The middle column contains the main drawing. To the far right there are options to manipulate properties of the individual objects in the visualisation. To bind data to property, user can drag and drop a variable to a property field.

2.3.2 Features

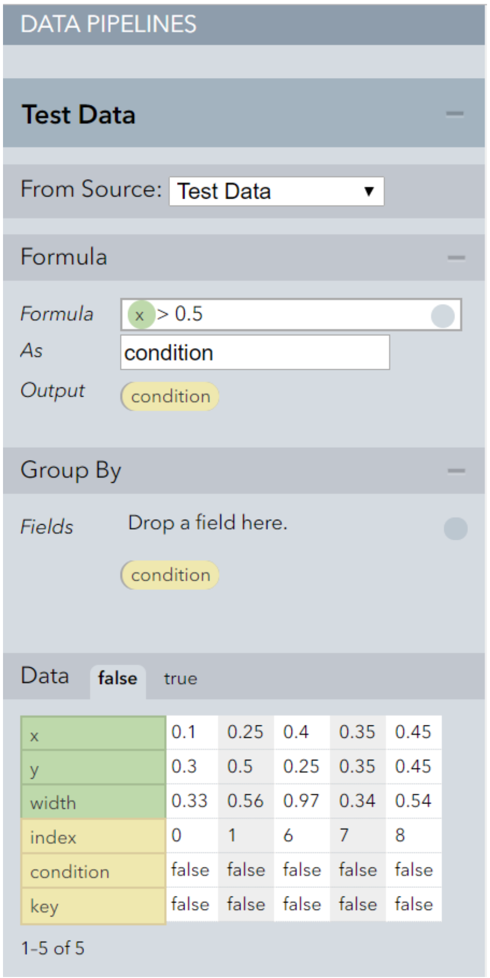
Manipulation

Lyra authors organised the tool to contain all of the data manipulation options on the left side of the view named Data Pipelines. A single pipeline consists of the actual data, transformation options and scales.

Original data can not be changed in any way, but the user is given an option to create new attributes based on the existing ones with several functions. Functions can be changed, but in current review version cannot be re-ordered, requiring the user to know the exact steps or redo the previous work in case of a mistake. The available functions are:

- **Filter**: adds a new textbox to the filter area, giving users an option to drag attributes into the textbox and write a filtering condition. Each textbox enables user to provide a condition written in JavaScript language, making a condition such as $Math.pow(x, 2) > 0.5 \; \&\& \; (y < 0.1 || y > 0.9)$ valid.

- **Sort**: as the name implies, user can drag and drop multiple attributes in order in which he or she likes the data to be sorted on. Generated attribute Index is changed, which affects the order in which data is iterated when doing visualisations.



DATA PIPELINES

Test Data

From Source: Test Data ▾

Formula

Formula: x > 0.5

As: condition

Output: condition

Group By

Fields: Drop a field here.

condition

Data **false** true

x	0.1	0.25	0.4	0.35	0.45
y	0.3	0.5	0.25	0.35	0.45
width	0.33	0.56	0.97	0.34	0.54
index	0	1	6	7	8
condition	false	false	false	false	false
key	false	false	false	false	false

1-5 of 5

Figure 2.8: Screenshot of the data pane. Data has been imported and named manually. An attribute *condition* has been calculated with formula transformation with condition $x > 0.5$ and at the end data was grouped by newly created attribute. Tab *false* can be seen as selected at the bottom part of the image.

- **Formula**: a new attribute is generated based on the given formula. The user experience and flexibility are identical to the *Filter* functionality. Generated attributes are coloured yellow.
- **Group by**: the data is split by the given attributes. All given attributes are treated as categorical data even if the values are decimal numbers, unable to specify grouping ranges. All rows with the same value are split into two separate datasets, which can be observed as tabs. In case of a boolean split, created tabs are true and false. For easier understanding, a Figure 2.8 is provided.
- **Stats**: it calculates the number of occurrences (count), max, mean, median, min, std. dev., sum and variance for the selected variable.

It is worth mentioning, that the user needs to explicitly drag and drop variable into field, simply typing x does not work. There is also a sixth function named *Window*, but the function is not mentioned in the documentation. During the exploration it appears it perform convolution on the selected attribute between the two consecutive index values to smooth the transition.

Scales

To represent different dimensions, Lyra uses the term *Scales*. Upon binding a variable to the axis, a new scale is automatically created. For each scale user has an option to select the type, either uniform, ordinal, logarithm and several others. They are normally split to a domain and a range which can be of multiple types such as colour and shape. First specifies the number range and the second the value we map to. An example of such case would be mapping a temperature attribute, which can be negative, to colour or size of the object. In the first case number domain derived from temperature attribute is mapped between blue and red colour range to indicate cold and hot areas with linear type. For the object size example, the values are mapped between two manually specified positive numbers, since the area can not be negative, resulting in a shifted and scaled domain.

Scales can be reused to keep the same scale range in visualisations such as Nightingale's Flower [Nightingale et al. 1859].

Visualisation canvas

In the centre of the tool is the main visualisation canvas. At the top, six main building blocks are provided, which can be dragged and dropped to the canvas. Once positioned, properties can be edited in the *Layers* pane or resized on the canvas.

Layers

On the left side, Lyra tool features a pane for layers as a way to organising visualisations. Primarily it serves the purpose of customisation of object properties. To create a clearer separation between types, developers split each layer into *axes* and *marks*. First are used, as the name suggests, to create axes and adjust corresponding configuration such as styling, ticks or tick frequency, grid lines and labels. The second similarly but with an intent to manipulate objects that build the chart itself. Marks can be reordered and have also an additional option named visual layouts. They provide automatic calculation of different values as bindable variables for visualisations. Such an examples are an angle for the pie chart and geographic coordinates or force-directed layout coordinates after the simulation, which normaly are not provided with the dataset.

2.3.3 Import and export options

Lyra supports all of the project related actions such as new, load and save, located at the bottom of the visualisation layout.

User can use one of the many example datasets predefined in the tool or import its own. Lyra enables users to import the data by providing a URL address to the data or by manually pasting the data in the text box. An option how to parse the data needs to be manually specified. The tool supports JSON, CSV and TSV formats, without an option to change the delimiter. Once the data is parsed, a user can see detected data types for each attribute and change it if necessary.

It supports both saving the project and loading, but the project is stored in the web browser's IndexedDB storage as JSON object. An experienced user is able to export the data from one machine to another or between the browsers, but due to the required knowledge, in this paper it is assumed that Lyra does not support project saving. For export, it supports PNG and SVG as well as export to standard Vega JSON format.

2.3.4 Subjective assessment

Lyra is quite a versatile tool. The tool offers many data manipulation methods and but provides users with only six different marks to draw. What was missed was a feature such as free drawing tool to create custom shapes and options for different anchor positions. For example, rectangle can only be positioned based on its top left corner, preventing us to position it based on its centre point.

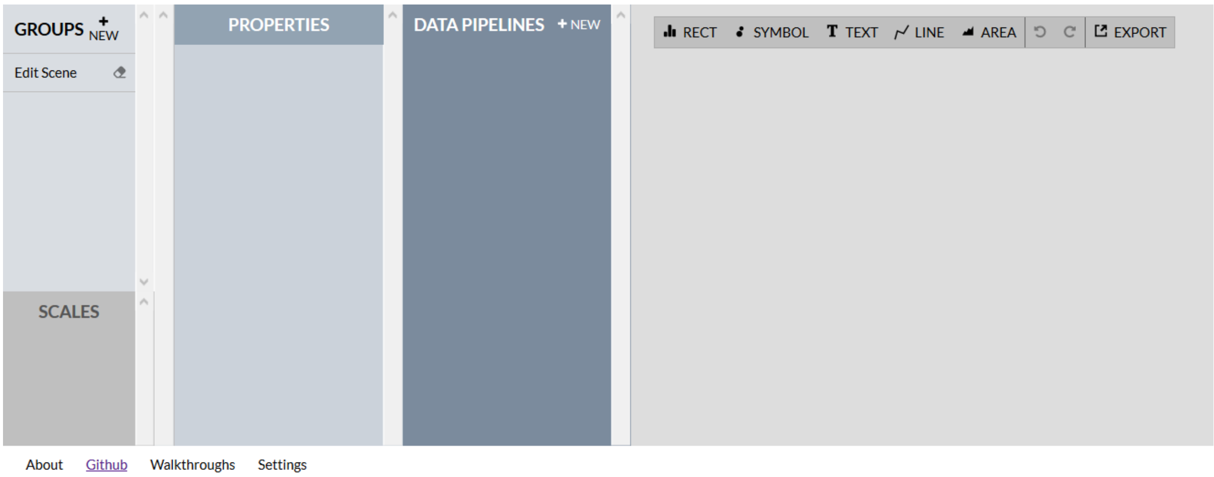


Figure 2.9: Screenshot of the development version of Lyra 2 taken on 27 May 2019.

The tool showed its weaknesses when some of the radial layouts were tried. The tool supports pie chart with an option to manipulate inner and outer radius, but it forces the use of pie slices. Labels are also not supported in such scenarios, requiring manual placement of text objects next to every slice.

Several of bugs were also noticed, which makes the entire experience quite difficult. The tool sometimes loses track of the object positions, unable to re-position them. Project loading does not correctly restore internal variable of the project. Adding a new data source or a mark will override the existing ones, forcing the user to remake the chart if any changes besides styling are necessary.

Lyra 2 was also installed locally, but was not usable due to very early stages of development. From the Figure 2.9 can be seen that while the user interface was changed, its main building blocks and principles remain the same.

2.4 Tools not included in the assessment

While selecting tools for testing apart from the tools described above a few others were also present in the list. However, during the process of testing and trying around, it was decided that some of those tools are too specific to be compared with the others - they are aimed on visualisations of very specific sort of data and cannot be used widely for different data sets like the others. Other ones among them were discarded by the creators and no longer maintained, thus the versions available were not functioning properly and not having all the functionality presented in official tutorial videos.

Nevertheless, short description of them would still be listed below, giving a brief overview of their special features and peculiarities.

iVisDesigner

iVisDesigner [Ren 2015a] is a visualisation tool created firstly as a thesis project by Donghao Ren of University of California, Santa Barbara and Peking University. It was launched first in March 2013 under the BSD 3-clause license, with an online version hosted for public usage. It is claimed to be an expressive tool for working with multidimensional data and creating such charts as scatterplots or parallel coordinates. The demo-video presented on the website is showing different opportunities that the tool is supporting. Moreover, there is a link to the source code available too, leading to the repository with the system.

Unfortunately, by the moment of this survey being written, the creator of iVisDesigner have stopped maintaining the project, switching to a newer one, Charticulator. Due to this reason, the online version

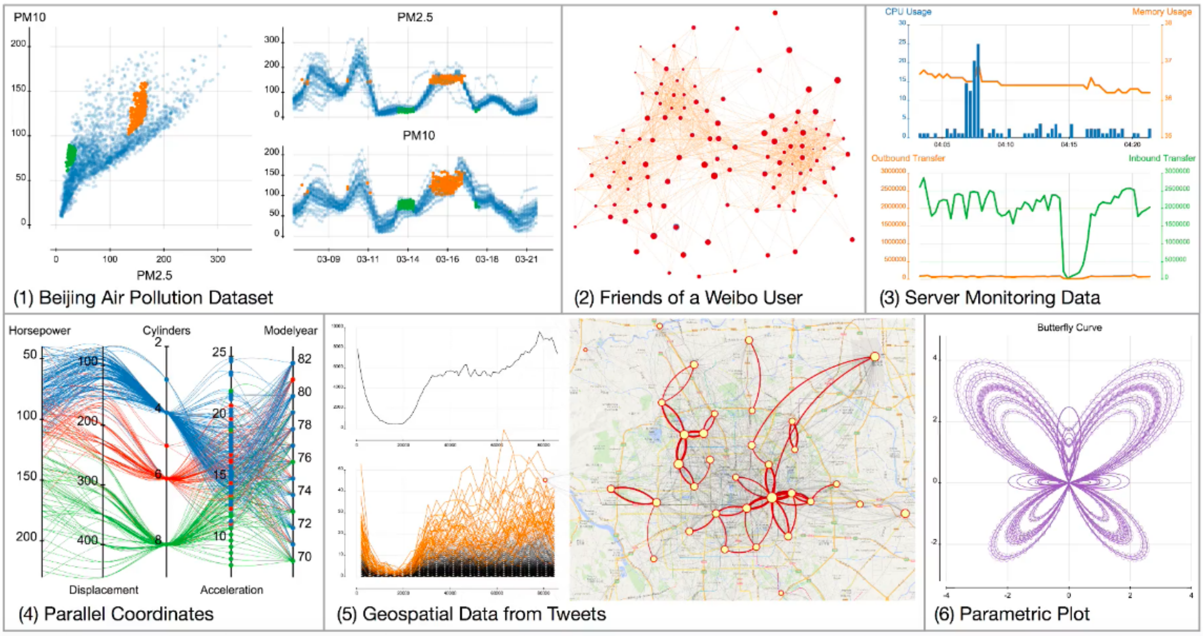


Figure 2.10: Examples of the charts and plots created with iVisDesigner. Screenshot is taken from the demo-video at [Ren 2015a].

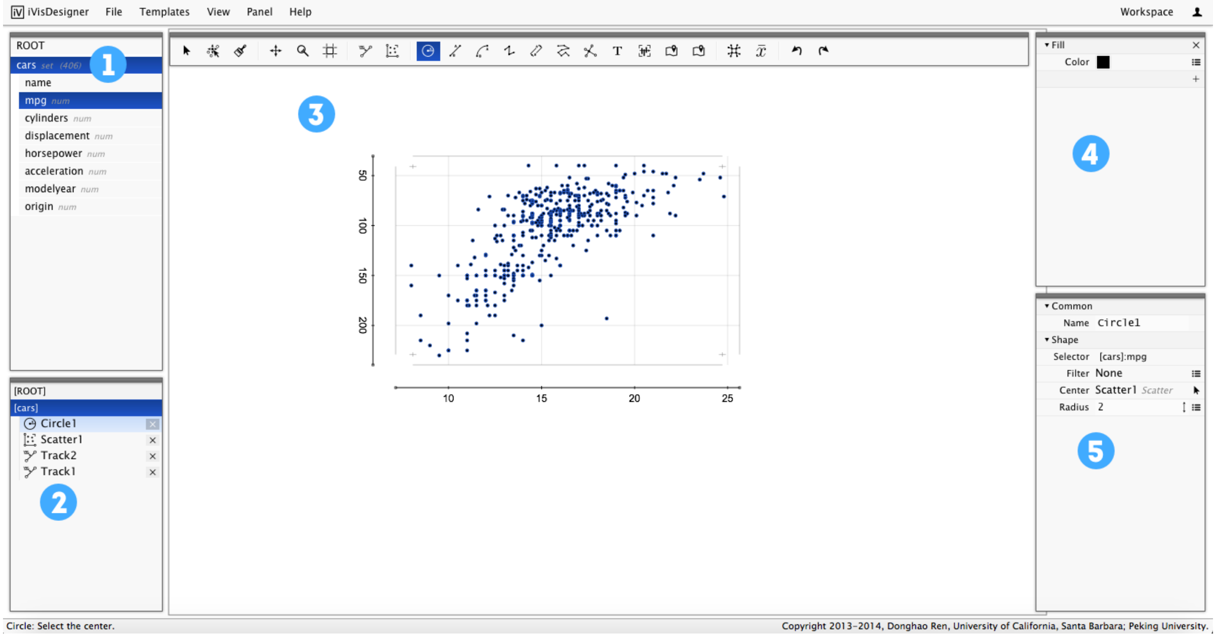
of the tool presented is not functioning properly - many of the announced functions are not working at all, making the tool almost impossible to use. This was the main reason why the tool was rejected in the process of evaluation.

However, the potential of iVisDesigner was rather promising. The tool is template based, claimed to be able to create scatter plots, parallel coordinates, timelines, graphs and even map-based plots, as can be seen in Figure 2.10. The user can import data as a CSV table or YAML Schema and Data, being able to modify the data a bit before the actual import if needed. The user interface seems to be rather simple and intuitive, as can be seen in Figure 2.11. Number 1 in the screenshot represents the fields of the dataset, while Number 2 - the list of the objects in the plot: all the axes, lines, shapes, etc. Number 3 is the actual working space, where the objects and plots can be placed, with a toolbar above it. The toolbar has buttons for creating and editing the objects, and also options to undo/redo changes. Settings fields on the right are connected to appearance (Number 4) of the selected objects and data binding properties (Number 5).

The workspace is allowing the user to create more than two axes and even several plots in the same area. A plot can be constructed manually by drawing the axes separately and then connecting them with dots or lines representing data correlations, or the user is free to choose from one of the plot templates, picking the required data fields for it.

Among the most interesting features mentioned in the demo-video is a tool to highlight certain data in the plots (e.g. Figure 2.12), what can be extremely useful in heavy parallel coordinates plots or several plots with the same data - the selection is applied to the data in all the plots present at the same time, meaning, for example, highlighting several nodes in a graph would also highlight the same objects in the co-occurrence matrix for those nodes. Another peculiar option is to tie the data points to specific coordinates on the map, that can be attached on the background. This could have been used for recreating one of the test-case datasets described later, where the information about Napoleon's army is connected to its position and movement. Unfortunately, the described tools are not present in the online version available on the website, and after reporting the issue, a reply from the author was received, that the toolkit is no longer maintained.

In the version available online there are less functions that were presented in the tutorial video, and the top menu is now working correctly - the documentation to the tool and the SVG exporting are unavailable.



Name Circle1

Selector [cars]:mpg

Filter None

Center Scatter1 Scatter

Radius 2

Circle: Select the center.

Copyright 2013–2014, Donghao Ren, University of California, Santa Barbara; Peking University.

Figure 2.11: Screenshot of the user interface of iVisDesigner, version available online at [Ren 2015b].

The tool in general was quite promising and proved to be a good foundation for Charticulator, the tool based on iVisDesigner. It does not have as much freedom as other data drawing tools described in this paper, however, it is still rather flexible with the opportunities it has included. It is easy to understand and learn how to use the tool, so the user would be able to grasp the concepts quickly and start using iVisDesigner to the full extent.

DataInk

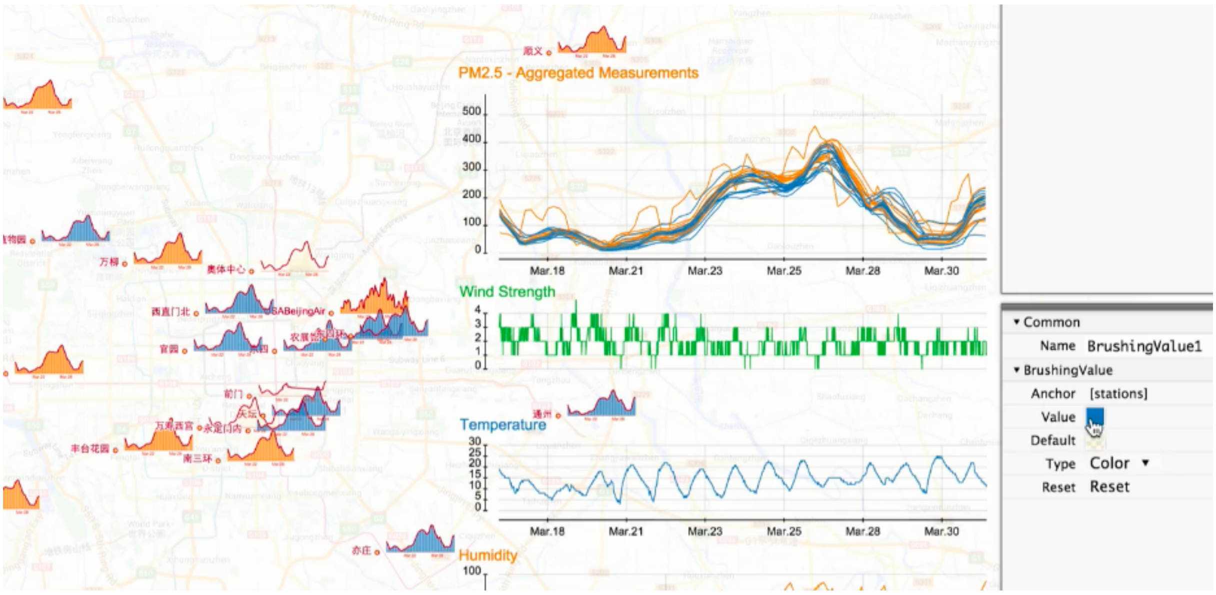
DataInk [Xia et al. 2018] is a glyph-based visualisation tool with a direct support of pen and touch input. It was initially put into the list of tools interesting for the research but later discarded as a very specific tool with a very narrow range of application opportunities. Mostly it can be used for classifying the data by different properties, being able to express those properties visually, so the user can easily see which data object possess which characteristics, also being able to group the objects by characteristics and place them in specific user-defined layouts.

The tool is perfect for on-the-go visualisations on presentations, for example, when the presenter needs to show the data in different layouts dynamically, or change the properties representations on the fly, observing the patterns in the data. DataInk requires absolutely no programming skills, so it can be used by anyone willing to have an expressive way to show the characterised data. Also, no special drawing skills are necessary - the glyphs can be as simple and minimalistic as the user wants.

DataInk was presented in 2018 by University of Toronto in cooperation with Microsoft Research and INRIA. The tool was tested by some participants, that have designer skills and experience, as well as by standard users without experience in art and design. All the participants coped with the tasks set before them, being able to use the tool properly without major issues.

The tool allows to hand-sketch glyphs as well as data layouts, manipulate binding of the glyphs to the data by adding, deleting and editing, modifying glyphs by color selection, scaling and moving.

The simplicity of the tool has both drawbacks and benefits - on the one hand, it is really easy to use and the learning curve is as small as possible. However, on the other hand, DataInk is very field specific and the expressiveness of the tool regarding different sorts of data is rather low.



▾ Common

Name BrushingValue1

▾ BrushingValue

Anchor [stations]

Value 🟦

Default

Type Color ▾

Reset Reset

Figure 2.12: "Brushing" tool for the data - coloring specific lines in the timeline plot on the right would automatically highlight corresponding plots on the map to the left. Screenshot is taken from the demo-video at [Ren 2015a].

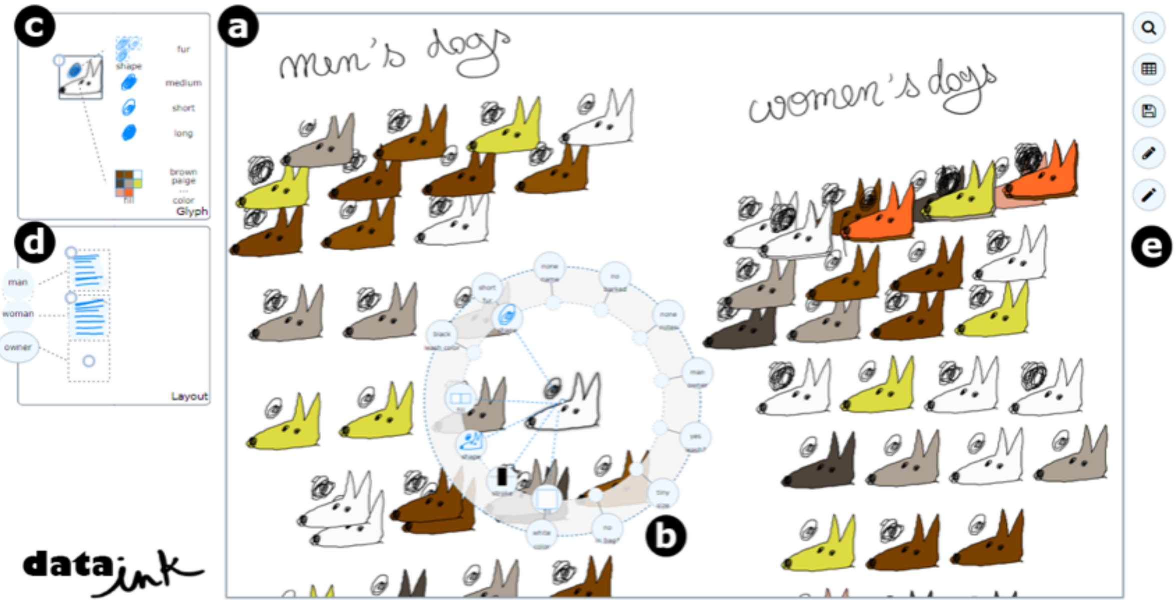


Figure 2.13: DataInk interface with (a) data canvas, (b) context-visual data palette with represtation tools, (c) glyph and (d) layout palettes, (e) side menu. Screenshot taken from [Xia et al. 2018].

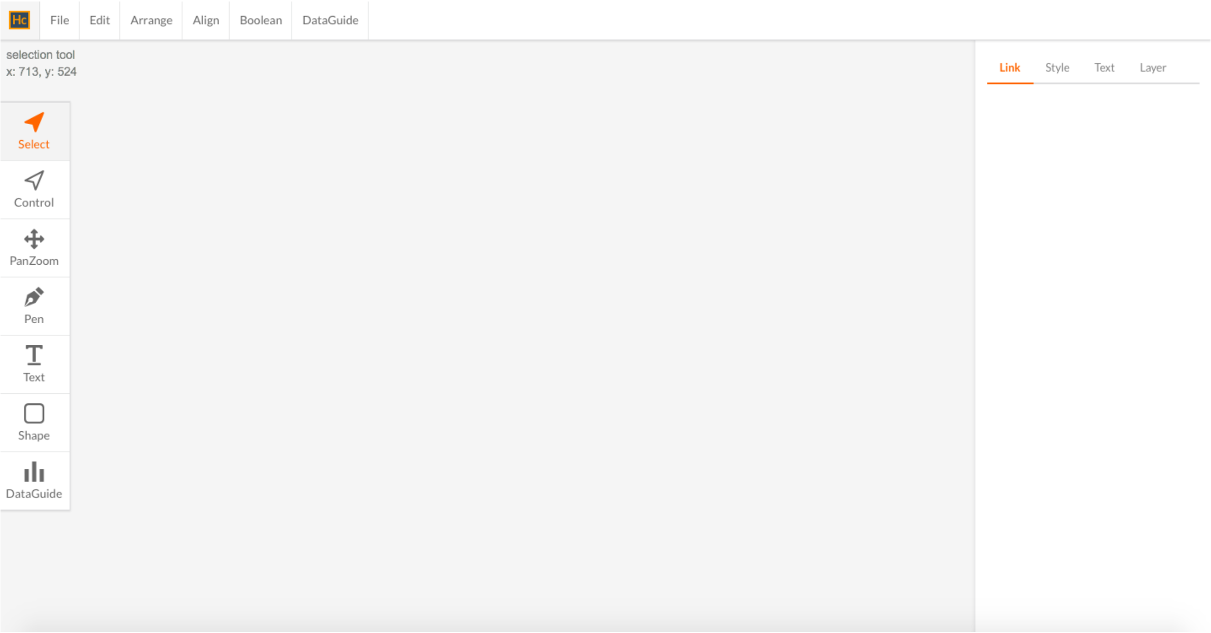


Figure 2.14: Graphic User Interface of Hyecoo Tool. Screenshot taken from [Hyecoo 2018].

Hyecoo

Hyecoo [Hyecoo 2018] was presented in 2017 [Kim et al. 2017] by Harvard University, Cornell University and Adobe Research. The tool was originally of interest, but was later also discarded from assessment because of being too specific. Hyecoo is claimed to be a drawing tool using Data-Driven Guides to represent numerical data. The downside of this is that it makes the tool useful only to create infographics, having only one-dimensional data. The main idea was to create a tool like Adobe Illustrator [Adobe 2016] or InkScape [Inkscape 2016], with less functionality to make it easier to learn and use, but also with data-binding properties.

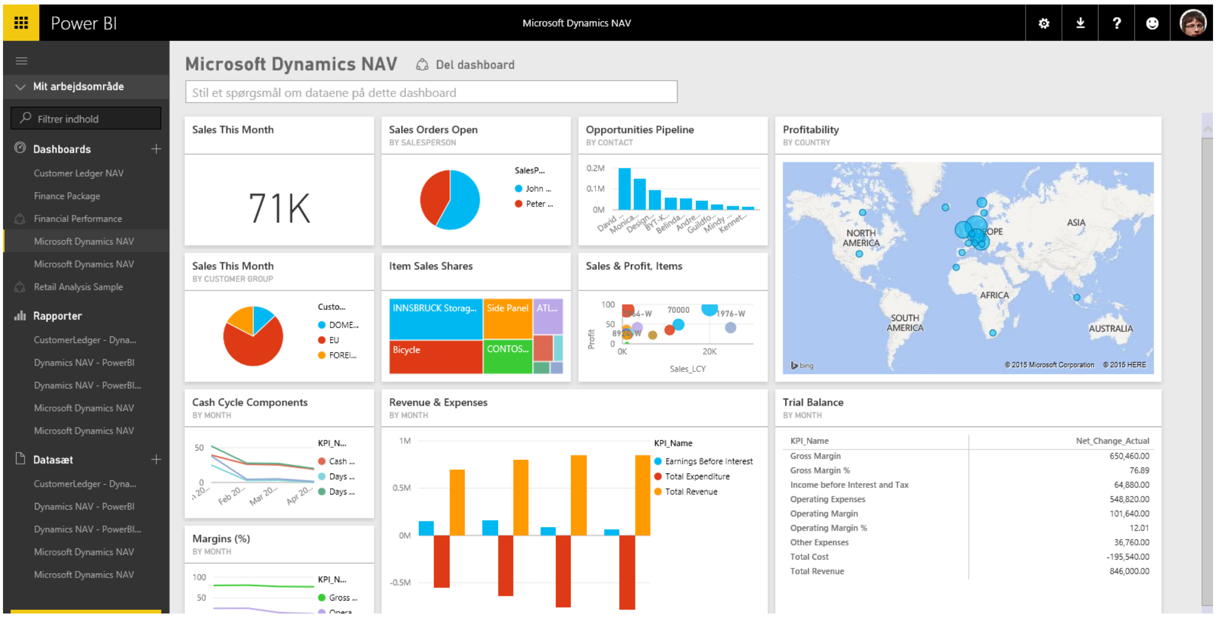
Hyecoo has rather inconvenient importing - only tab-separated data with labels in one line and values in another. After entering correct data by hand or pasting in, the user can create the guides corresponding the data. Afterwards the user is free to customise and draw shapes for the guides to represent the data in a better way. The tool allows to have several sets of data guides at the same time for more complex infographics.

The interface is very simple (Figure 2.14), but nevertheless is not user-friendly. The whole idea of using Data-Driven Guides must be documented better with more detailed tutorials and examples, than only the demo-video. It is hard to understand how to set up data linking for the objects, how to change their properties and so on. Although the examples shown in the videos look impressive and nice, the user would experience a lot of difficulties trying to get to that level of using Hyecoo.

PowerBI & Infographics Designer

Power BI [Microsoft 2019] is a Microsoft's business analytics service which aims to provide interactive visualisations on desktop or web for Office365 users. Its goal is to provide a tool simple enough to be used by any user to create their own dashboard for their business needs. The tool is heavily inspired by Excel, thus the user experience is also quite similar, visible on Figure 2.15. It was launched in 2015 and due to its enterprise target audience and good adoption it is well funded.

On the top a ribbon bar provides all of the operations for the project, such as adding a new chart, or importing the data. On the left there are options to switch between different views, namely the dashboard, the data table or entity relationships diagram of the data model. Finally, on the right there are properties



Trial Balance
BY MONTH

KPI_Name	Net_Change_Actual
Gross Margin	650,460.00
Gross Margin %	76.89
Income before Interest and Tax	64,880.00
Operating Expenses	548,820.00
Operating Margin	101,640.00
Operating Margin %	12.01
Other Expenses	36,760.00
Total Cost	-195,540.00
Total Revenue	846,000.00

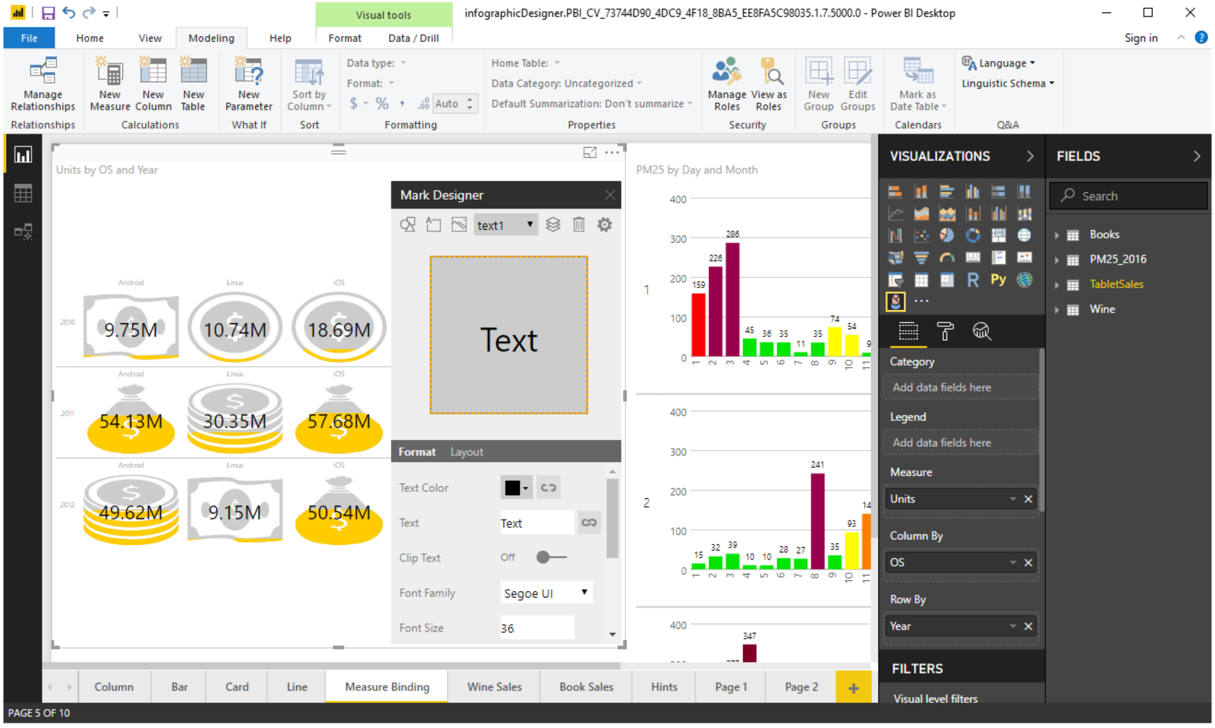
Figure 2.15: Graphic User Interface of Power BI Tool. Screenshot taken from [Microsoft 2019].

for the selected chart. The charts are predetermined, only parameters are exposed. Parameters represent attributes such as axis, mark colour and label and are set by dragging and dropping the desired field over the designated area. Due to a large user base adoption, a large number of community created templates can be acquired from the official Power BI store.

On of such extension in the store is called Infographics Designer [Microsoft 2016] (InfoNice short). Infographic visualisations are getting increasingly popular as they improve the effectiveness of the visualisations by making them quickly understood and easily remembered. They do this by introducing graphics from the same domain as the shapes on the charts. The InfoNice's goal is to extend manipulation options to the chart shapes, such as rectangles composing bar charts, points for scatter plots etc. to be replacable. Added pane is visible in Figure 2.16.

With the Mark Designer, shapes can be imported or picked from the built-in library. It also introduces some properties available for data binding and the concept of layers for marks. The entire approach is similar as in Charticulator but to a more limited extent.

The tool is perfect for everyone trying to do one of the most popular visualisations and scales well with the amount of data. It offers the most diverse data import options and manipulation to eliminate the need for multiple tools to do data processing. But similar as the other tools, limits from template based visualisations can quickly become too constraining for complex charts thus it was not classified as Data-Driven tool.



infographicDesigner.PBI_CV_73744D90_4DC9_4F18_8BA5_EE8FA5C98035.1.7.5000.0 - Power BI Desktop

Text Color ■

Text: Text

Clip Text: Off

Font Family: Segoe UI

Font Size: 36

Measure: Units

Column By: OS

Row By: Year

Figure 2.16: Graphic User Interface of Power BI Infographics Designer Tool. Screenshot taken from [Microsoft 2019].

Chapter 3

Test Scenarios

In this Section five test scenarios are described with the aim to cover as many functionalities as possible to provide a good overview of the tools.

3.1 Bret's dataset

Bret Victor presented the tool for creating data-driven visualisations in his talk at Stanford [Victor 2013]. The dynamic picture he drew during the talk is visible in Figure 3.1 with an example dataset in Table 3.1. The attributes x and y represent normalised points in 2D space, while width represents the standard deviation. The aim of using this visualisation as a test scenario is to examine various tool capabilities such as:

- **Drawing points with custom drawn shape**: a line of width $width$ with vertical marks at each side centred in the middle.
- **Lines connecting points**: points should be connected corresponding to the order in the dataset and each line should end with an arrow head.
- **Representing deviation**: full height transparent column.
- **Tick marks along an axis**: only at the values with a present point.
- **Multiple charts**: sharing the same coordinate area and axis.

3.2 Napoleon's march to Moscow

Napoleon's march to Moscow [Minard 1861], in Russia also known as *Patriotic War of 1812*, is a visualisation created by Charles Joseph Minard (see Figure 3.2). It displays the size of the Napoleon's army departing from Polish-Russian border marching towards Moscow. The wheat colour designates the men who enter Russia, the black those who leave it. Band thickness represents the army size at a rate of one millimetre for every ten thousand men. For spatial perception, names of the cities are placed alongside the march path. At the bottom of the visualisation the temperature and date are depicted.

#	x	y	width
1	0.09	0.17	0.09
2	0.2	0.53	0.07
		...	
8	0.46	0.63	0.1

Table 3.1: Example data recreated by Bret Victor's visualisation.

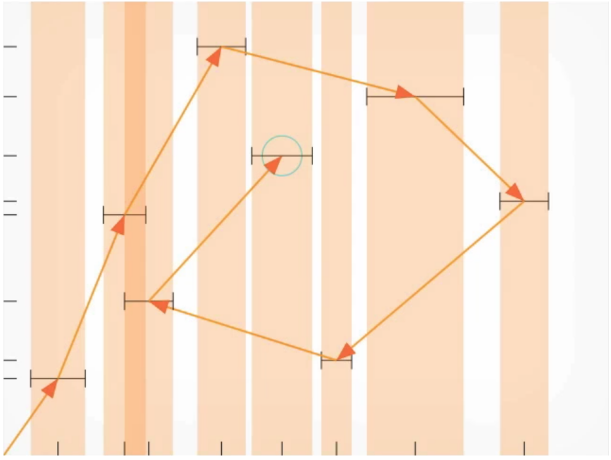


Figure 3.1: Bret’s demo visualisation result.

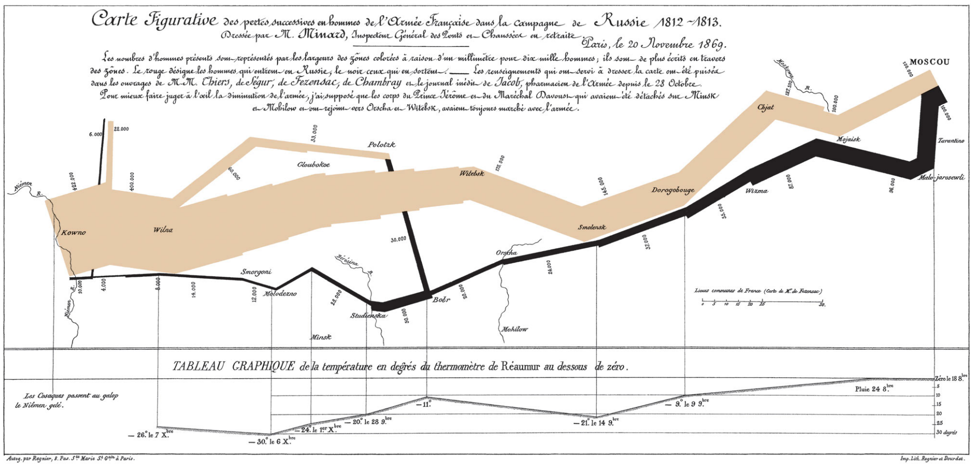

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Niémen R.
Kowno
Wilna
Gloubokoe
Polotzk
Witebsk
Smolensk
Dorogobouge
Wizma
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Moskowa R.
Smorgoni
Molodezno
Bérézina R.
Studienska
Minsk
Bobr
Orscha
Mohilow
422.000
400.000
6.000
22.000
60.000
33.000
175.000
145.000
127.100
100.000
100.000
100.000
96.000
87.000
55.000
37.000
24.000
20.000
50.000
30.000
28.000
12.000
14.000
8.000
4.000
10.000
Lieues communes de France (Carte de M.r de Fezensac)
0 5 10 15 20 25 50.
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
Zéro le 18 8.bre
Pluie 24 8.bre
— 9.° le 9 9.bre
— 21.° le 14 9.bre
— 11.°
— 20.° le 28 9.bre
— 24.° le 1.er X.bre
— 30.° le 6 X.bre
— 26.° le 7 X.bre
5
10
15
20
25
30 degrés
Autog. par Regnier, 8. Pas. S.te Marie S.t G.ain à Paris.
Imp. Lith. Regnier et Dourdet.

Figure 3.2: Minard’s visualisation of Napoleon’s Russian campaign. Image source [Wikipedia.org 1861].

The dataset [datavis.ca 2013] consists of three parts. First represents the number of Napoleon’s troops, the latitude and longitude, direction of travel and the army group (see Table 3.2). Second and third part represent city locations and temperature and their example can be seen in Table 3.3 and 3.4 respectively. This test scenario examines following features:

- Loading multiple datasets.
- Drawing multiple charts on a single canvas.
- Mapping longitude and latitude to x and y coordinates in the canvas space.
- Setting line thickens.
- Aligning two charts to represent the same value, longitude in this case.

The visualisation is complex as it introduces different coordinate space and line bifurcations and several different points as well as conditional colouring.

#	lon	lat	size	dir	group
1	24	54.9	340000	1	1
2	24.5	55	340000	1	1
			...		
49	24.1	54.7	6000	-1	3

Table 3.2: Example of Napoleon's march to Moscow dataset: location, size, direction and group of the army.

#	lon	lat	city
1	24	55	Kowno
2	25.3	54.7	Wilna
		...	
20	37.3	54.8	Malo-jarosewli

Table 3.3: Example of Napoleon's march to Moscow dataset: location and names of the cities.

3.3 Weather Radial visualisation

This test scenario is a recreation of Weather Radials infographic [Raureif 2013]. It features a radial illustrating the climatical characteristics of a place for a single year. Each day in a year is depicted as a coloured line, spanning from that day's minimum temperature to maximum, with colour representing the daily mean temperature. On the centre of every day's temperature line there is a blue circle, indicating the amount of precipitation for that day. Data was acquired from National Oceanic and Atmospheric Administration (NOAA) [NOAA 1807] for Boston area. The example of the dataset is shown in Table 3.5. The test scenario examines following tool capabilities:

- Radial layout.
- Grid lines and ticks within radial layout.
- Colour gradient with multiple colour steps.
- Combining several attributes within the same radial layout.
- Each item is a combination of several attributes.

3.4 Nightingale's rose diagram

Nightingale's rose diagram seen on Figure 3.4 visualises deaths during the Crimean War between April 1854 and March 1856. The visualisation has two polar area charts, each split into twelve slices representing months. Each slice has three coloured areas each measured from the centre of the circle representing different causes of deaths. The blue area represents deaths from diseases, the red deaths from wounds

#	lon	temp	date
1	37.6	0	18-Oct
2	36	0	24-Oct
		...	
9	25.3	-26	07-Dec

Table 3.4: Example of Napoleon's march to Moscow dataset: temperature at specific geographic point at specific date.

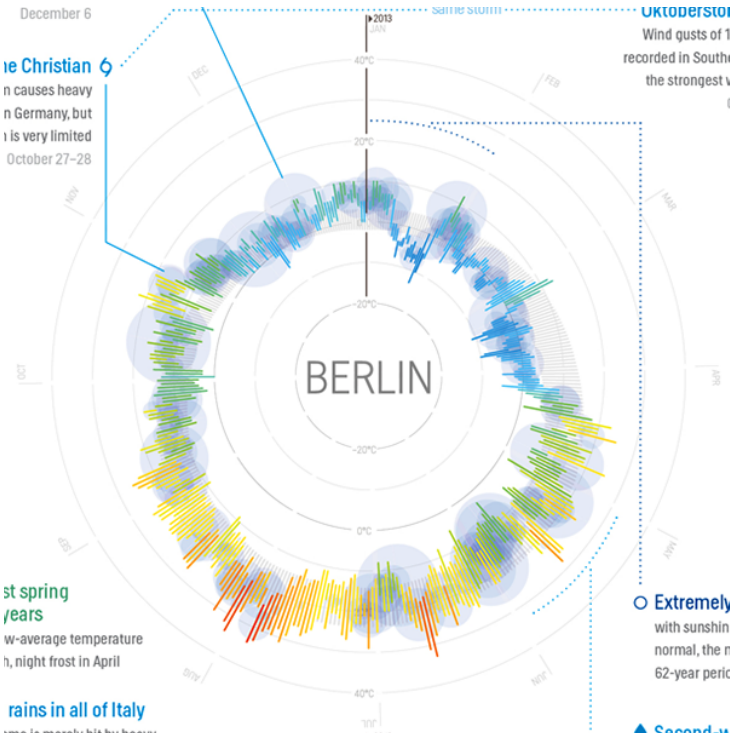


Figure 3.3: Open weather radials example for Berlin. Image source [Kekeritz 2013].

#	date	location	meanTemp (F)	maxTemp (F)	minTemp (F)	precip	avgPrecip
1	1/1/15	Boston, MA	28	33	22	0	0.11
2	1/2/15	Boston, MA	36	41	31	0	0.11
				...			
365	12/31/15	Boston, MA	40	47	33	0.08	0.1

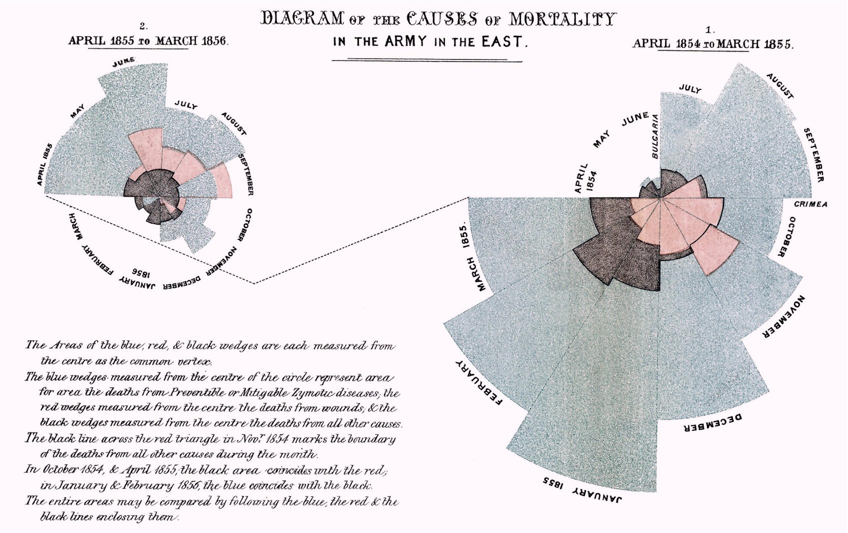
Table 3.5: Example of Boston weather measurements dataset for year 2015: date, location, mean temperature, max temperature, min temperature, precipitation, average precipitation.

and the black deaths from other causes. The example of the dataset taken from [Nightingale 2015] is displayed in Table 3.6. Using this visualisation as a test scenario examines following capabilities:

- Drawing radial chart.
- Mapping value to area.
- Labelling radial chart.
- Manipulating order of the elements so that slices do not occlude each other.

3.5 Box and Whisker Salaries chart

The box and whisker chart in Figure 3.5 shows the yearly statistics for the top ten jobs salaries grades in the state of Calisota. Each box has whiskers - lines that mark the endings of the lines coming out of the box, representing statistic information for one job. The minimum and maximum values are denoted with the vertical lines while the first, the second (median) and the third quartile are represented with a box. The box and whiskers chart may also include outliers - values that do not belong to their category's range between the minimum and the maximum value and are unique for their category, for example, only one person to have a salary like this. The example of the dataset used for this visualisation is shown in



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.
The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic diseases; the red wedges measured from the centre the deaths from wounds; & the black wedges measured from the centre the deaths from all other causes.
The black line across the red triangle in Novr. 1854 marks the boundary of the deaths from all other causes during the month.
In October 1854, & April 1855, the black area coincides with the red; in January & February 1856, the blue coincides with the black.
The entire areas may be compared by following the blue, the red & the black lines enclosing them.

Figure 3.4: Nightingale's rose diagram. Image source [Nightingale 1856].

#	date	army size	disease	wounds	other
1	1854-04-01T07:00:00.000Z	8571	1	0	5
2	1854-05-01T07:00:00.000Z	23333	12	0	9
		. . .			
24	1856-03-01T08:00:00.000Z	46140	15	0	35

Table 3.6: The example of Nightingale's rose diagram dataset: date, army size, deaths from disease, wounds or other causes.

Table 3.7. This test scenario covers following capabilities:

- Creating a box and whisker plot.
- Using custom attribute bound shapes.
- Handling empty values.
- Using categorical x axis.
- Handling long text labels.

#	x	low	q1	median	q3	high	outlier_1	outlier_2	outlier_3
1	Nurse	20000	26000	27000	32000	38000	50000	52000	
2	Hygienist	24000	28000	32000	38000	42000	48000		
					. . .				
9	Dentist	75000	99000	123000	160000	210000	220000	70000	
10	Physician	58000	96000	130000	170000	200000	42000	210000	215000

Table 3.7: The example of the Box and Whisker Salaries chart dataset: the minimum and maximum values, all the quartiles and up to three outliers.

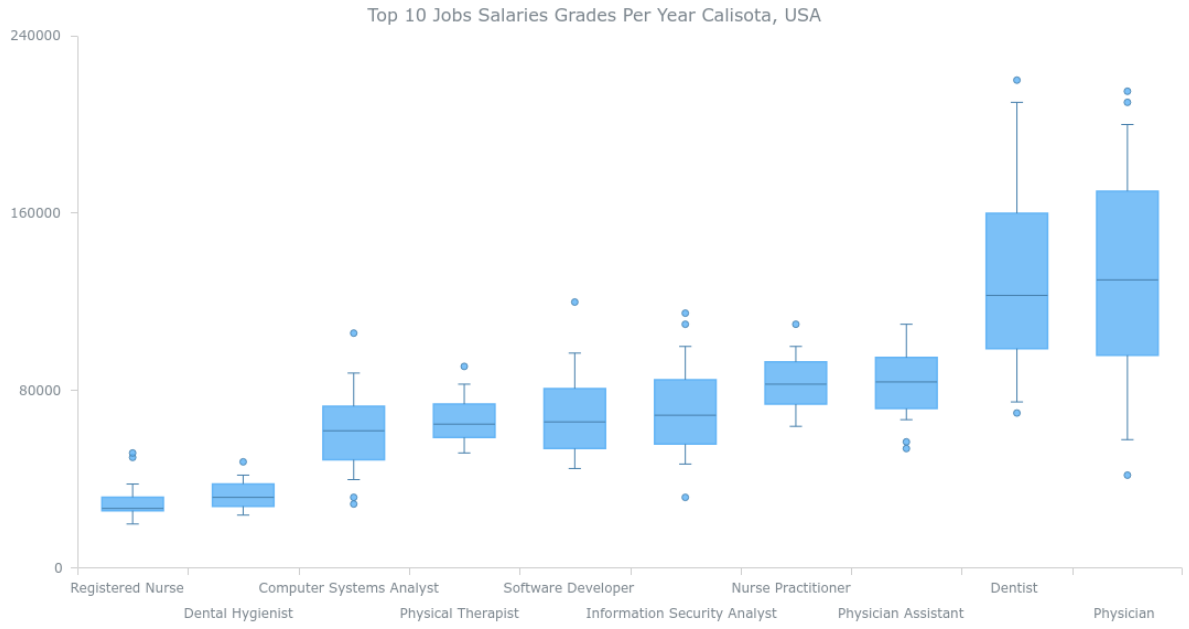


Figure 3.5: Box and Whisker Salaries chart. Image source [AnyChart.com 2019].

Chapter 4

Results

This Chapter describes results of test scenarios performed in tools chosen for the assessment: Charticulator, Data Illustrator and Lyra.

4.1 Bret's dataset

Charticulator

The recreation of Bret's visualisation can be seen in Figure 4.1. It is very similar to the original, although there were some features that could not be recreated, namely a horizontal lines with vertical marks for each bar and adjusted angle of arrows so they would point in the direction of the line.

Data Illustrator

As shown in Figure 4.2, one can immediately notice multiple coordinate axes. Although there is an option to reuse the axis, all the attempts that were made resulted in a broken layout (the whole graph disappearing). However, it is also possible to reuse the *scale* of existing axis, resulting in a new axis drawn next to the existing one. In this case, an aligning of the shapes has to be done manually.

Lines connecting the points were drawn fast and easily by partitioning the line shape and binding data to *x* and *y* position.

Taking a closer look at horizontal lines representing width, it can be noticed that the shape is not actually a line but a very flat ellipse. This was a workaround to recreating lines because the tool only allows the user to repeat the line shape in a grid layout and not a horizontal one. The former is not suitable for this visualisation. Vertical tick marks could be drawn in the same way although a lot of manual recalculations of corresponding *x* and *y* coordinates had to be done first.

Although the tool has Layers Panel (see Figure 2.2, the green dot with number 4 in it), it does not provide a real concept of layers as the user is unable to do any layer related operations such as copying, hiding or moving. The lack of the latter functionality demands planning ahead so the shapes are drawn in the right order and they do not overlap. This problem can be partially saved by changing opacity but this is not always optimal.

Lyra

The recreation of Bret's visualisation is shown on Figure 4.3. The visualisation was partially recreated. Lyra showed some big limitations in regards to drawing in positioning. A point must be one of the predefined shapes, unable to import or draw a custom one. Another limitation showed up with background positioning. Rectangles can only be positioned based on the top left anchor point, by *x*, *y* or both coordinates, thus unable to position them based on the centre of the object.

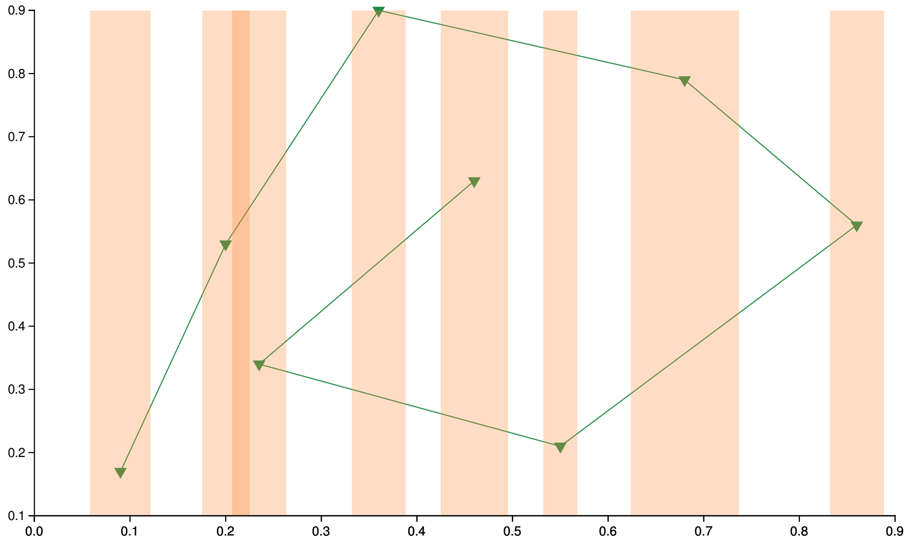


Figure 4.1: Recreation of Bret's visualisation in Charticulator.

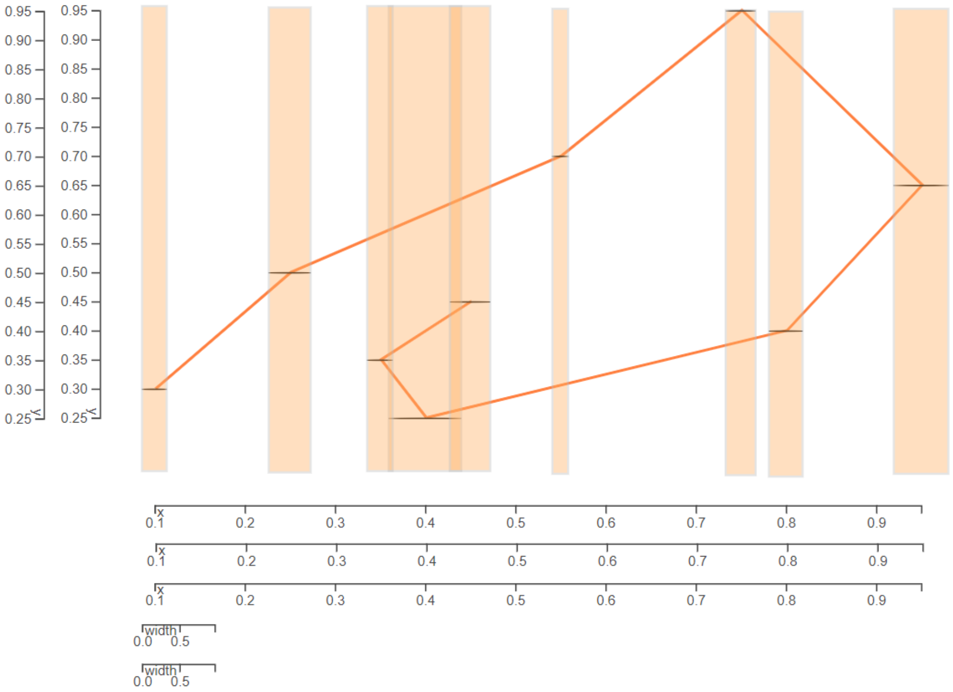


Figure 4.2: Recreation of Bret's visualisation in Data Illustrator.

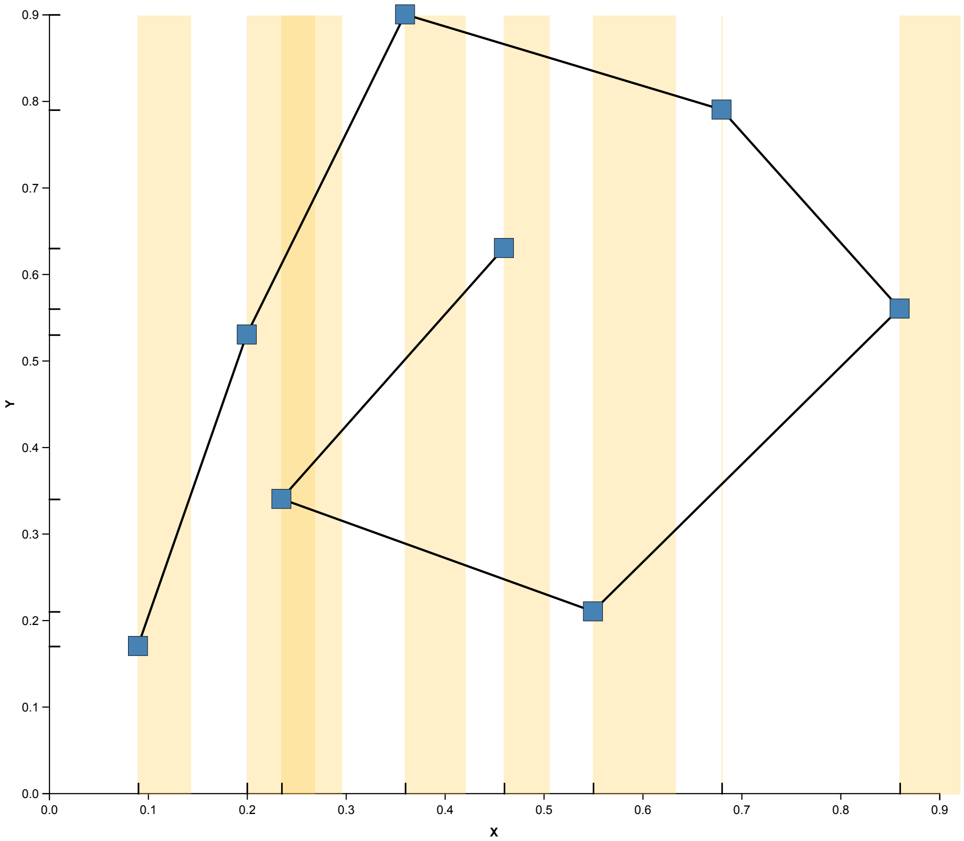


Figure 4.3: Recreation of Bret's visualisation in Lyra.

4.2 Napoleon's march to Moscow

Charticulator

While trying to recreate Napoleon's march to Moscow visualisation in Charticulator, a limit of the import feature was shown. There is no option to import multiple datasets simultaneously. We were able to colour the direction to Moscow with green and the retreat with red. The connections of different paths taken by the army are done by links. Due to the missing import of multiple datasets, it was not possible to add cities and temperatures. Other options were explored to try to show them but all of them failed. Nevertheless, there is a possibility to create multiple charts and then combine them in one chart as shown in Figure 4.4.

Data Illustrator

Figure 4.5 shows the recreation of Napoleon's march visualisation in Data Illustrator. During the drawing some limitations showed up. The army size could not be depicted as binding data to line width is not supported. The tool also does not allow conditional use of data, leading to manual preprocessing of the dataset shown in Table 3.2, namely splitting it into six files based on direction and group of the army. In addition, the user can not load multiple datasets, however datasets can only be loaded consecutively, having only one loaded. In this case, the undo action results in breaking the layout of previously drawn graphs. The main disadvantage that showed up while using multiple datasets was inability to reuse the axis. Therefore twelve different lines had to be manually scaled and aligned to fit the existing axes. To keep visualisation clear, only four axes were kept in Figure 4.5. There were several other bugs, namely failing to load the project file, blinking drop-down menu when choosing the axis and the whole visualisation becoming unresponsive, demanding to start over. During the recreation of this visualisation, the tool showed good promise as it is quiet intuitive, however, fixing all bugs mentioned would make it

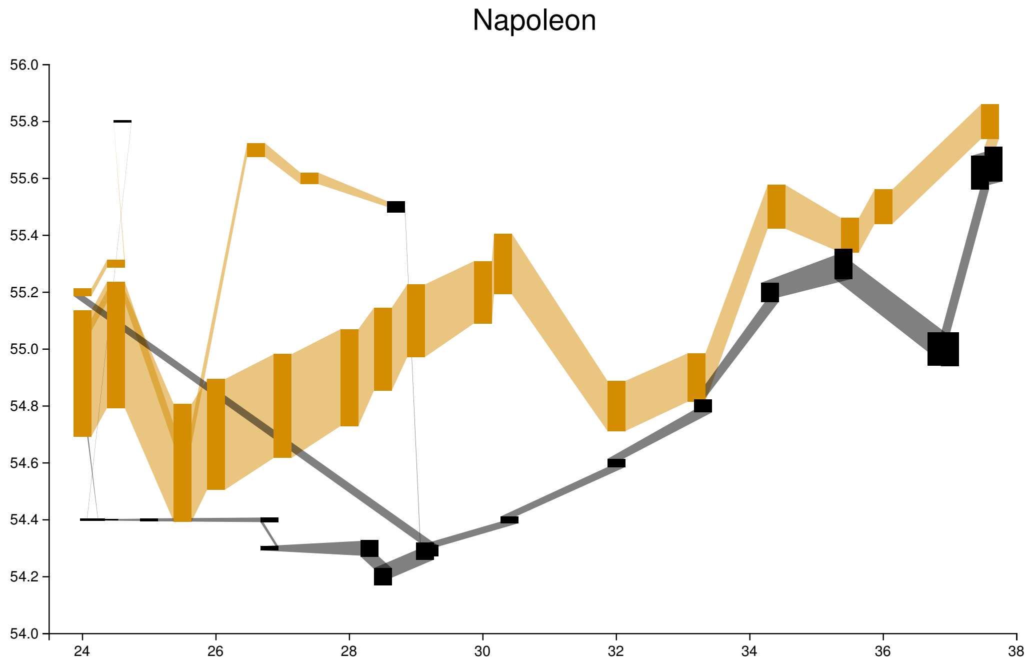


Figure 4.4: Recreation of Napoleon's march visualisation in Charticulator.

even better.

Lyra

Comparing the original image and the visualisation recreated by Lyra in Figure 4.6, recreated visualisation is very similar. However, due to the complex nature of the visualisation, the tool showed several bugs requiring to restart making the visualisation from beginning. Despite visual manipulation, a large amount of knowledge is required to create such chart. Two filters had to be applied, group by to split the army into groups and direction and window, which is an undocumented function that interpolates the line between two points to show gradual thinning. Mapping from geographic space to canvas space also required the knowledge of mean of our points and manually set the translation and scale factor. The tool could perform this task automatically based on the layer size.

4.3 Weather Radial visualisation

Charticulator

This scenario can be fully reproduced. A custom glyph with a scale was created. Within it is possible to set minimum, maximum temperature and perception for the selected day. A line was added between this points and coloured based on the predefined spectral gradient. This glyph was repeated in the radial layout for every dataset entry.

Data Illustrator

The result of selected test scenario is visible on Figure 4.8. Temperature lines could not be arranged in a circle but only horizontally. Still, the tool provides nice and easy way depicting temperature lines and colouring them based on the mean temperature value. There was only one issue, namely showing the precipitation legend, as the circles and values overlapped with no possibility to separate them.

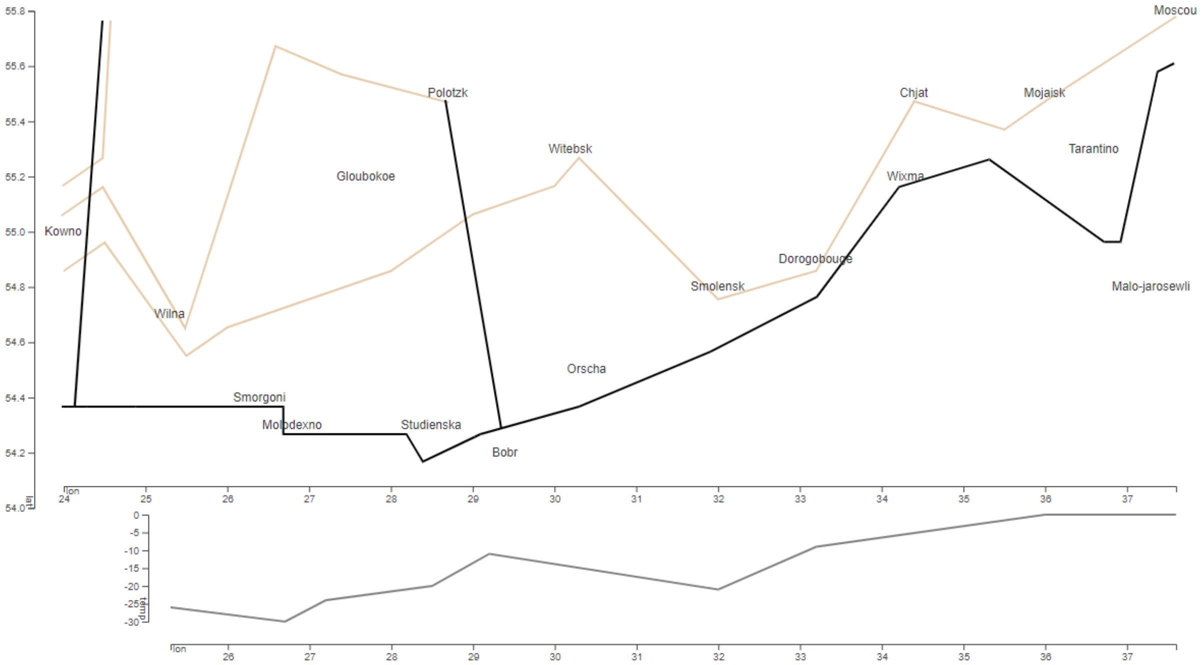


Figure 4.5: Recreation of Napoleon's march visualisation in Data Illustrator.

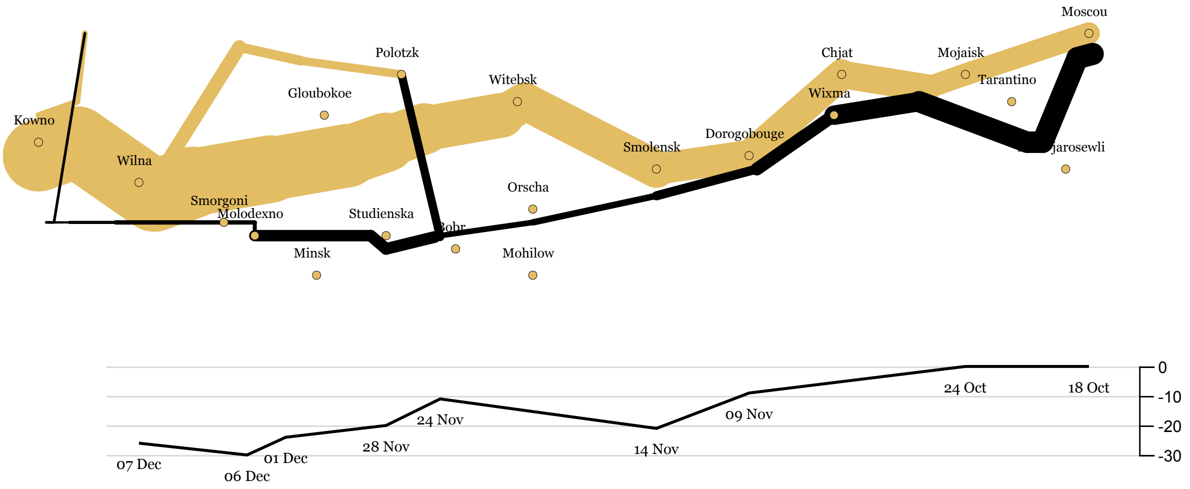


Figure 4.6: Recreation of Napoleon's march visualisation in Lyra.

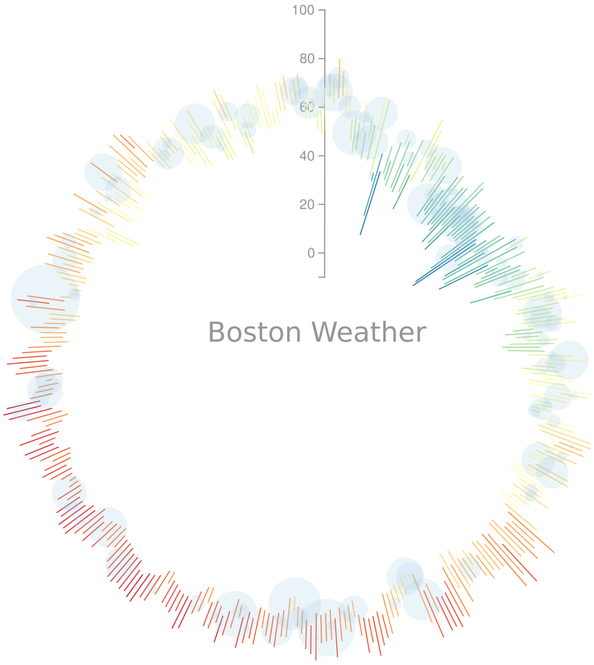


Figure 4.7: Recreation of Weather Radials visualisation in Charticulator.

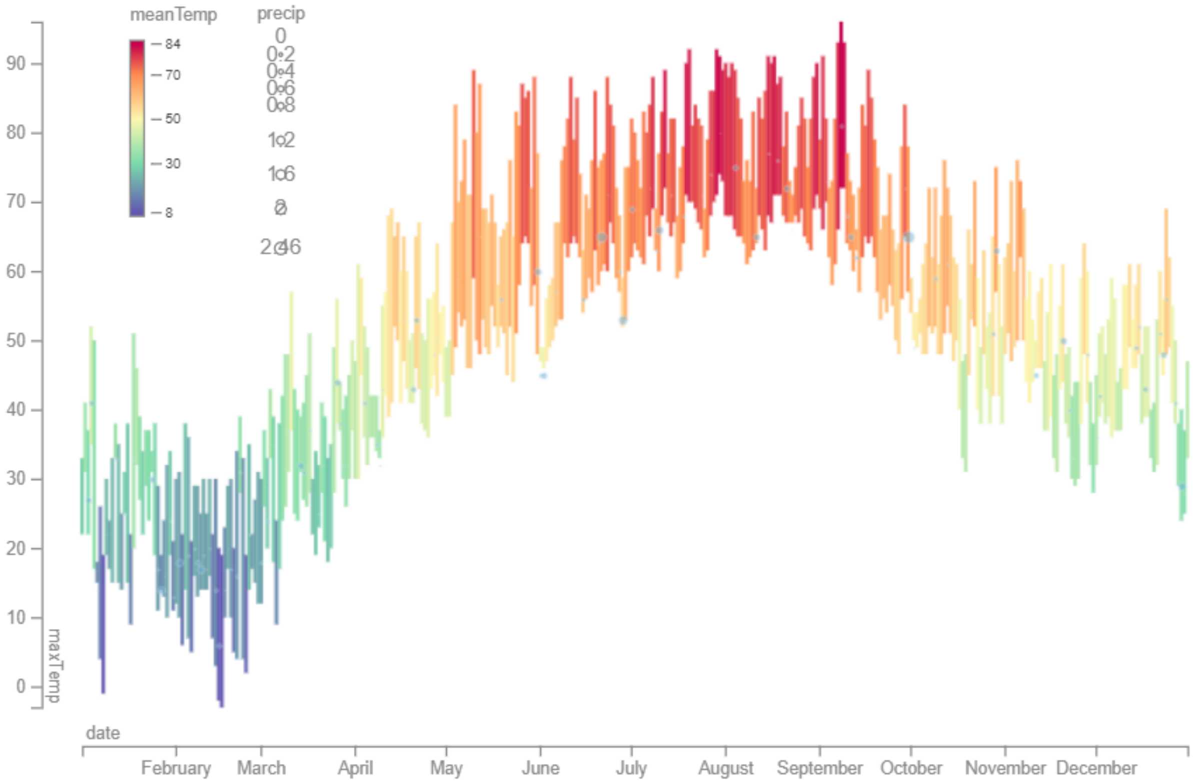


Figure 4.8: Recreation of Weather Radials visualisation in Data Illustrator.

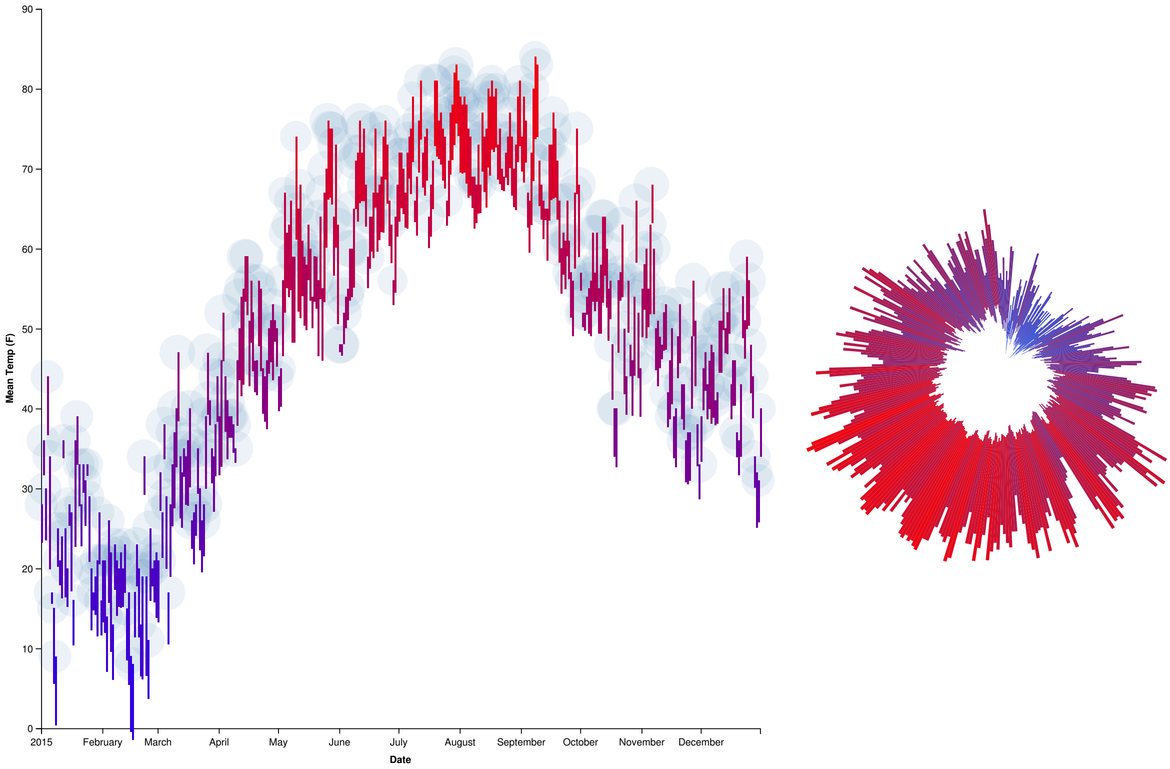


Figure 4.9: Recreation of Boston weather visualisation in Lyra. Line plot as bars for day indicators and circles for precipitation amount. Right the closest recreation in radial space.

Lyra

The selected test showed several issues and can be observed on Figure 4.9. As already noted, ticks in the radial plot are not supported as well as not any other object shapes. Minimum and maximum temperature were bound to inner and outer circle radius, resulting in a somewhat visually similar diagram on the left. But the colour scale only supports two values for interpolation, showing only blue and red. In order to indicate height on the line chart, a difference between minimum and maximum temperature had to be calculated using a formula data transformer and bound to height. Precipitation was successfully recreated on the line chart.

4.4 Nightingale’s rose diagram

Charticulator

With originally defined dataset visualisation is not recreatable. Changing the data to have multiple entries for each month and a type (disease, wounds, other) enables such an option. It creates a glyph for every entry, which is afterwards showed one over another by setting month for axis value, changing the chart to polar plot and sorting the values by value to avoid coverage. Result can be seen on Figure 4.10.

Data Illustrator

The radial layout is currently not supported in Data Illustrator. The alternative visualisation of Nightingale’s rose diagram is visible in Figure 4.11. Instead of using radial chart, the deaths are represented in a line chart. The functionality that proved to be very useful in this visualisation is the transparency of coloured area as it enables seeing overlapping parts.

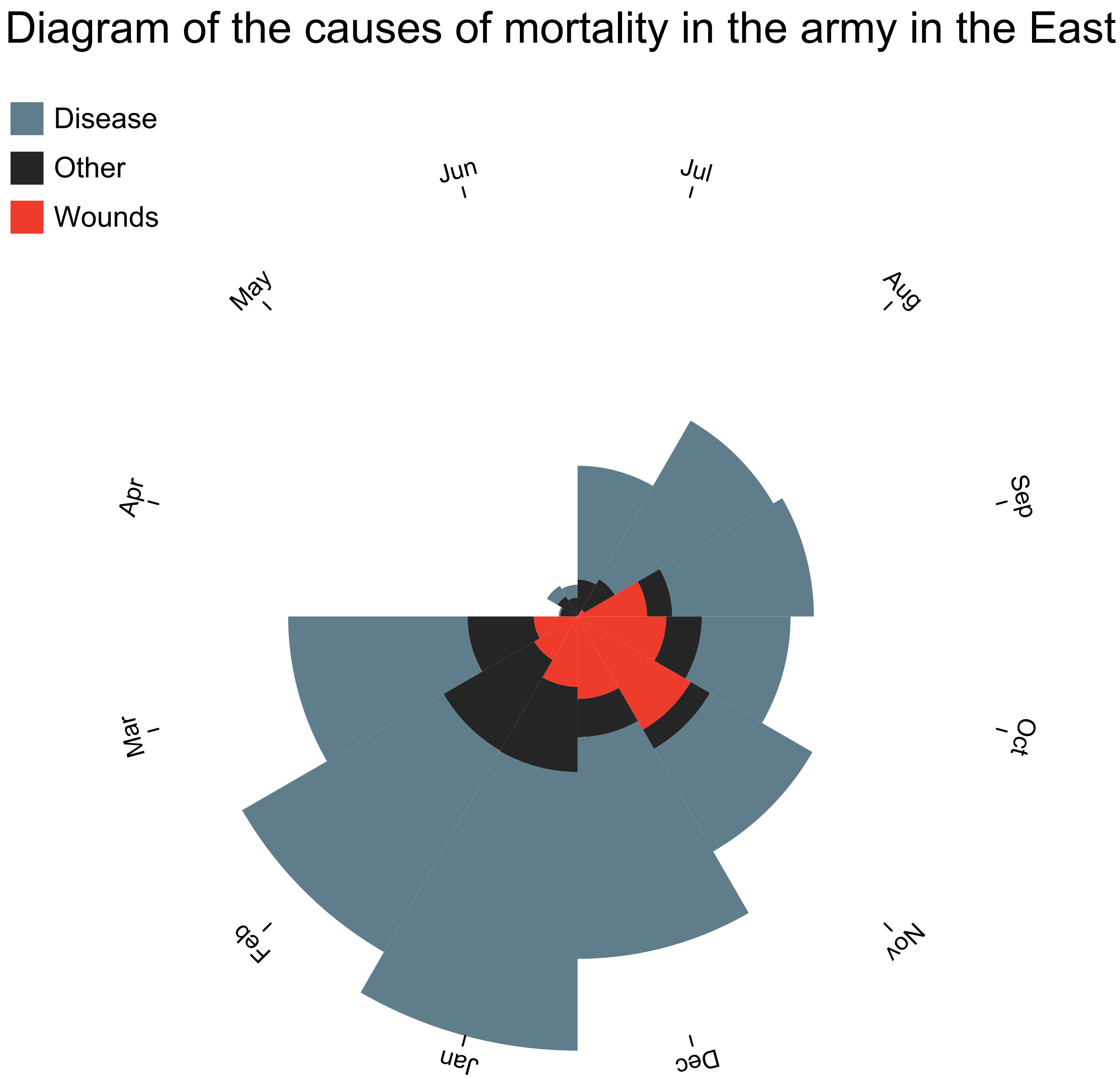


Figure 4.10: The recreation of Nightingale's rose diagram in Charticulator.

Lyra

The Nightingale's rose diagram was successfully recreated, Figure 4.12, by using multiple diagram plots and positioning them in the same centre. As there are three different charts one over another, the same scale was reused so that the distance from the centre represents the same distance unit. Due to overlaying, ordering for specific slices could not be changed. The red slices therefore completely cover brown for certain months hiding that information.

Several parts of the chart were unable to be done using Lyra. Values can not be mapped to area. Distance is therefore square root of the value to achieve an approximation of the ground truth. Ticks also support only Cartesian space, so all of the labels would need to be placed manually. Starting point for the polar diagram is always at the top. With April as a first month in the dataset, where normally January is expected to be.

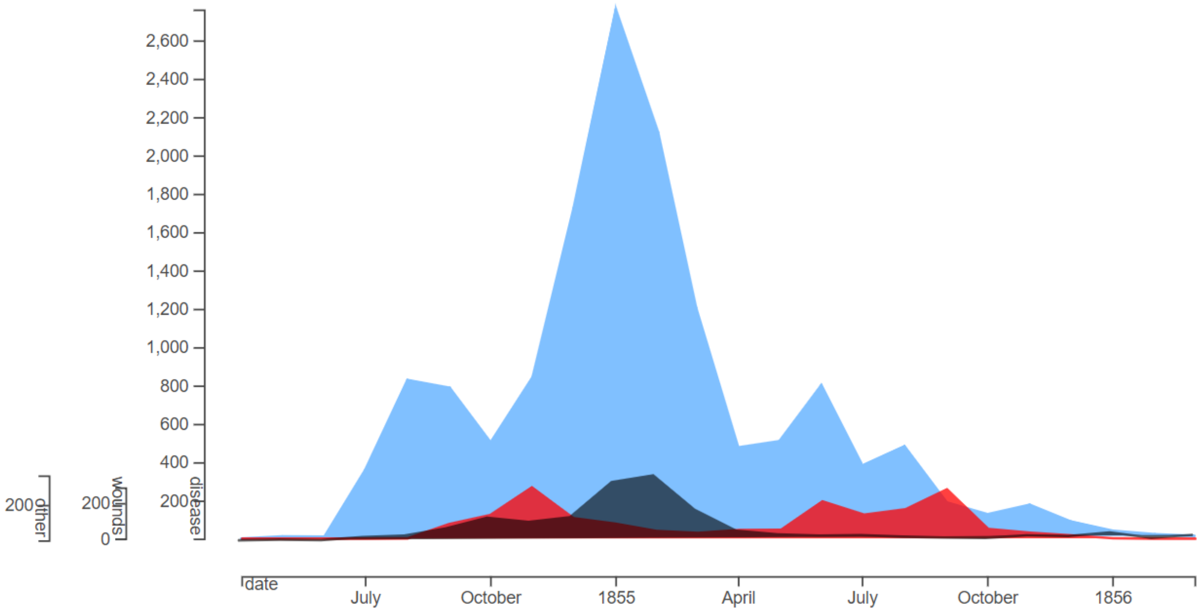


Figure 4.11: Alternative recreation of Nightingale’s rose diagram visualisation in Data Illustrator.

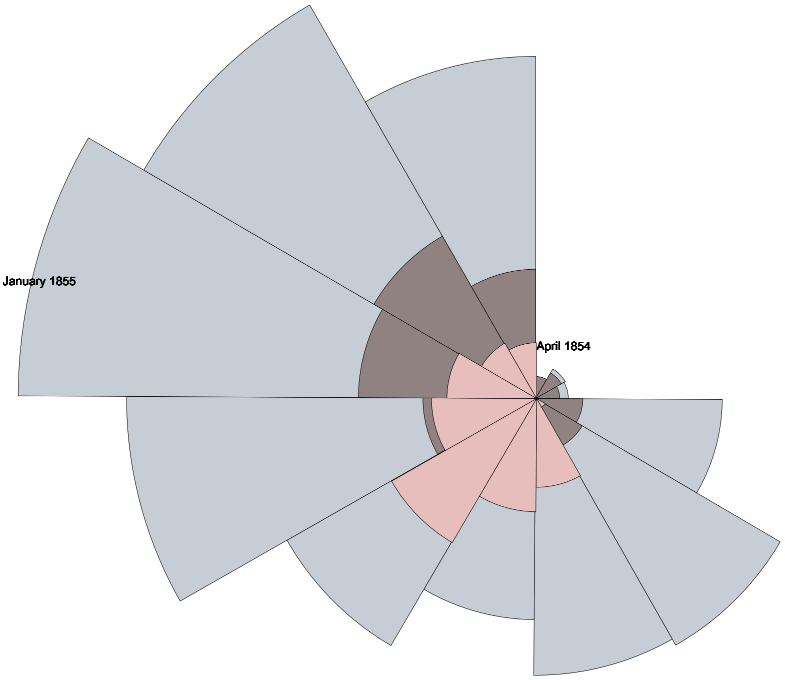


Figure 4.12: Recreation of Nightingale’s rose diagram in Lyra.

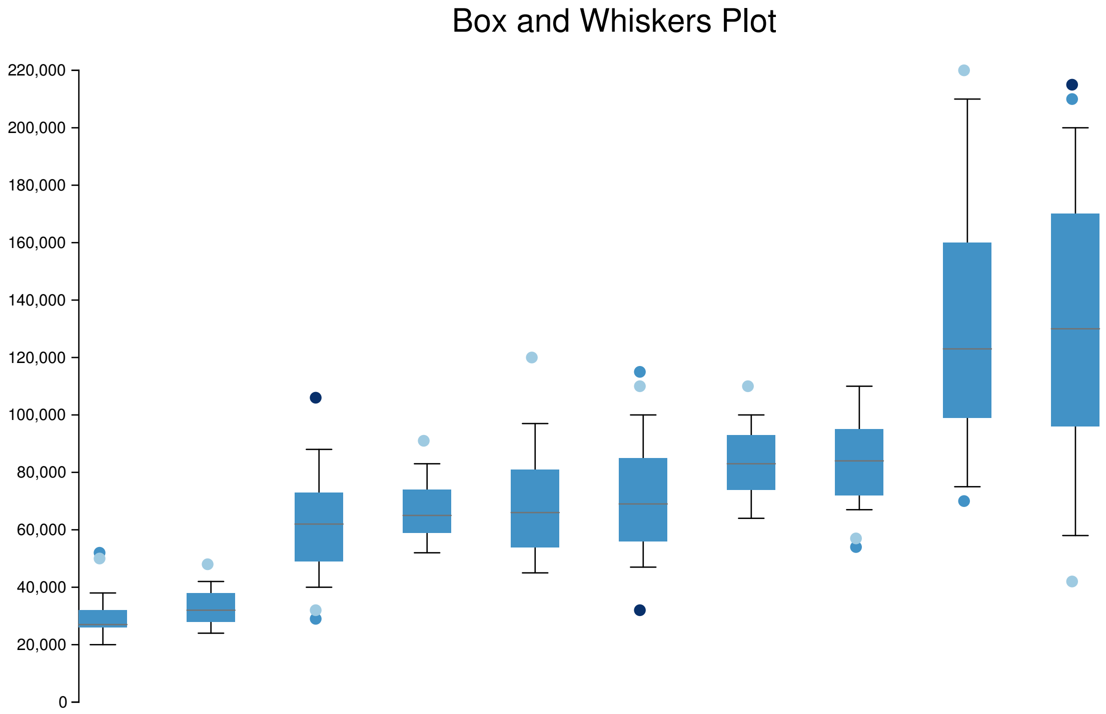


Figure 4.13: The recreation of box and whisker salaries visualisation in Charticulator.

4.5 Box and whisker salaries chart

Charticulator

This example is fully recreateable. As we can see, the bottom of the box is bound to the first quatile, and the top to the third quartile. The mean salary is shown with a dark grey line in the box. To show the low and high salary values we also used a grey line. For the outliers we simply used circles.

Data Illustrator

The box and whisker salaries visualisation recreated in Data Illustrator is shown in Figure 4.14. The tool provided everything needed to recreate this visualisation. The only downside that showed up during the creation was the inability to better handle long label names, for example splitting them in multiple rows or portraying them at an angle. The result is therefore the greater gap space between boxes than needed.

Lyra

This scenario can be recreated in Lyra (Figure 4.15), but some issues occurred similar to previous tests. The tool is not able to group marks into more complex symbols, but each part, box, line, each dot must be created separately on the same chart. Domain range for y scale must be manually defined so that the marks are placed at the correct positions. Central positioning again showed as an issue, unable to align the vertical line to the centre of the axis. Mapping categorical values to x axis worked without any issues. However, the tool treated all empty outlier values as zeros, which can be seen on the axis.

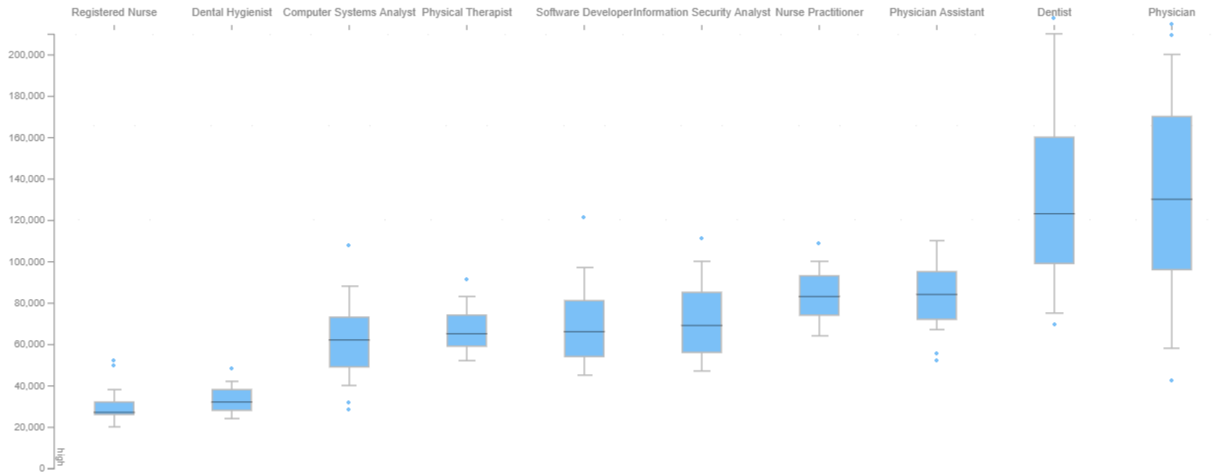


Figure 4.14: The recreation of box and whisker salaries visualisation in Data Illustrator.

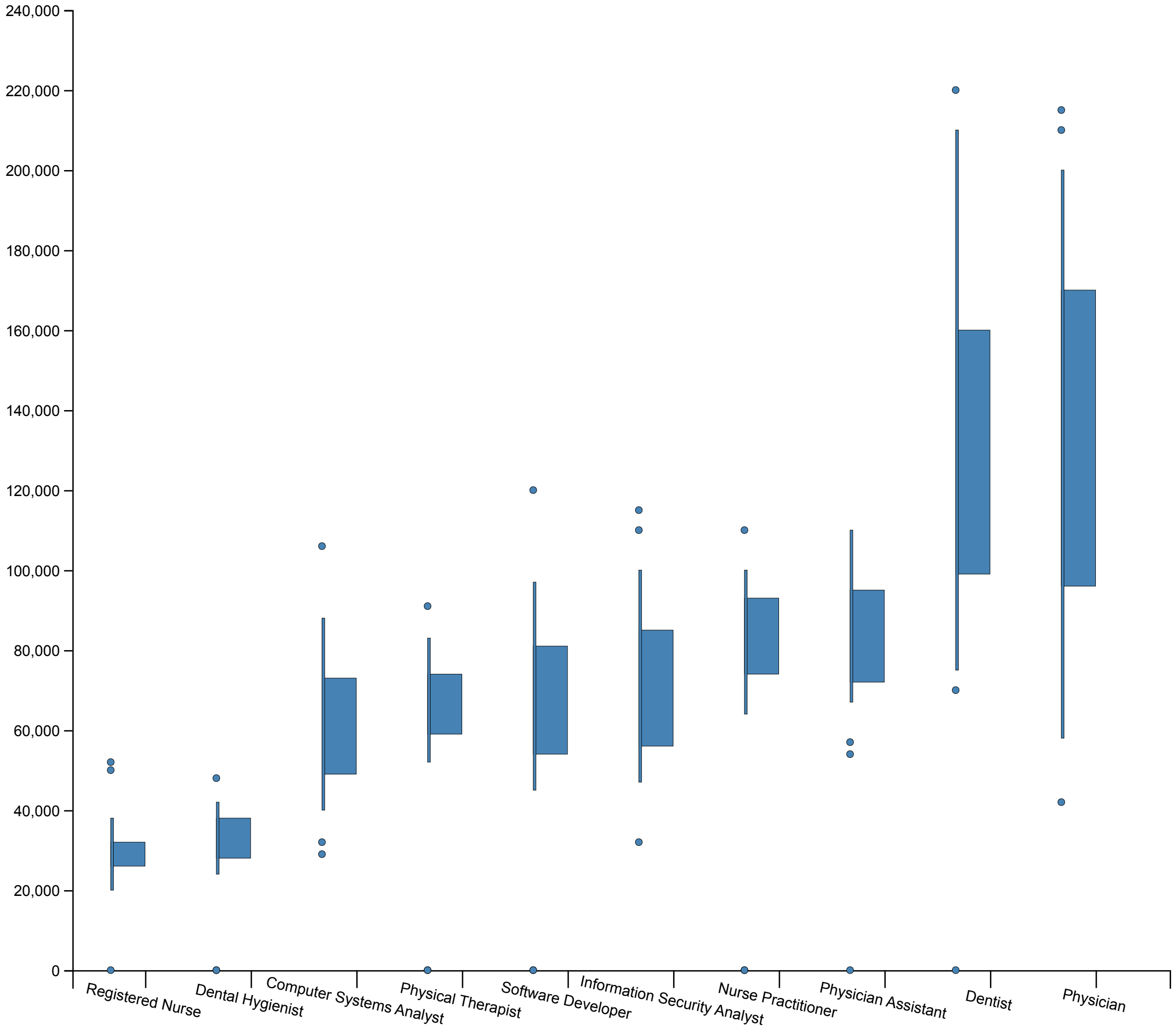


Figure 4.15: Recreation of box and whiskers plot in Lyra.

Chapter 5

Conclusion

Several Data-Driven tools were tested, discussed and evaluated using different test scenarios. We described several other popular tools but not included them in an evaluation as they are too specific to be compared with the others or are no longer maintained by the authors.

The test scenarios were combined out of five test scenarios that we normally can not easily create using everyday tools with the built-in plotting functions for visualising data. The test scenarios examine capabilities to draw different coordinate spaces, shapes and they test the tools for colouring and annotation options. While working on the survey, a more complete feature table was created (see Table 5.1).

From the Table 5.1 and the authors impressions, Charticulator is the clear winner. The tool is the most stable and reliable offering a rather large amount of features, which is visible in the Charts section of the table. Data-Illustrator and Lyra are ranked equally in our table. From the subjective opinions of the authors, Lyra seems to be a better choice especially with its stability, data manipulation options and ability to share axis which is mandatory. Data-Illustrator on the other side offers greater flexibility in the euclidean space and takes shape first approach same as Charticulator which is more intuitive and flexible compared to Lyra's approach.

All of the reviewed tools showed weaknesses with custom radial charts and high dependency on the format of the data, which must be carefully tailored for specific visualisations.

From the other tools, Power BI together with Infographics Designer showed most promise. With a large pallete of different visualisations and more charts available from the store, powerful data processing and import tools as well as the ability to manipulate the shapes, Power BI offers a great alternative to Charticulator and other Data-Driven tools.

All of the tools serve their purpose to offer a great amount of freedom in creating visualisations that the everyday tools can not provide. They eliminate the need to have coding skills making them accessible for a wider audience however still have some learning curve and require a different way of thinking. Unfortunately all of the tools have their limits and in such cases a programming knowledge is still needed.

Feature	**Charticulator**	**Data Illustrator**	**Lyra 1.0**
Importing			
External source (via URL)	✗	✗	✓
CSV	✓	✓	✓
Excel Spreadsheet	✗	✗	✗
Project file	✓	✓	✗
Data manipulation	✗	✓	✓
Export			
HTML	✓	✗	✗
Vector Image	✓	✓	✓
Raster image	✓	✗	✓
Project file	✓	✓	✗
Visual Tools			
Draw arbitrary shape	✓	✓	✗
Import arbitrary shape	✗	✗	✗
Object grouping	✓	✓	✗
Object Manipulation			
Colouring specific objects manually	✗	✓	✗
Colouring conditions	✓	✓	✓
Group to create a new shape	✓	✓	✗
Duplicate objects	✗	✗	✗
Scale	✓	✓	✓
Rotation	✗	✗	✗
Availability			
Source code available	✓	✗	✓
Self-hosted	✓	✗	✓
Scriptable	✗	✗	✗
Hosted	✓	✓	✓
License	MIT	Proprietary	BSD-3-Clause
Charts			
Several charts on one sheet	✓	✓	✓
Bar	✓	✓	✓
- Grouped/Stacked	✓	✓	✗
- Vertical/Horizontal	✓	✓	✓
Radial chart	✓	✗	✓
Line	✓	✓	✓
Scatter plot	✓	✓	✓
Parallel coordinates	✓	✗	✗
Network	✓	✗	✗
Sankey Chart	✓	✗	✓
Legend (automatic)	✓	✓	✗
User Experience			
Touch friendly	✓	✗	✓
Basic keyboard bindings	✓	✓	✗
Cross browser (if applicable)	✓	✓	✓
Stability (1 to 10)	9	6	7
Is under active development?	✓	N/A	No (Lyra 2.0)
Extensibility	✗	✗	✗
Total	**28**	**21**	**21**

Table 5.1: Feature evaluation sheet comparisons results.

Bibliography

Adobe [2016]. *Illustrator*. 21st Nov 2016. http://adobe.com/products/illustrator.html (cited on pages 1, 16).

AnyChart.com [2019]. *Box and Whisker Chart*. 2019. https://www.anychart.com/products/anychart/gallery/Box_Charts/Box_and_Whisker_Chart.php (cited on page 24).

Bostock, Mike [2011]. *D3.js*. 2011. https://d3js.org (cited on page 1).

Casey Reas, Ben Fry [2001]. *Processing*. 2001. https://processing.org (cited on page 1).

datavis.ca [2013]. *Re-Visions of Minard*. 2013. http://www.datavis.ca/gallery/re-minard.php (cited on page 20).

Division, Star [2011]. *Libre Office*. 2011. https://www.libreoffice.org (cited on page 1).

Hyecoo [2018]. *Hyecoo Online Tool Page*. 30th Apr 2018. https://hyecoo.namwkim.org (cited on page 16).

Illustrator, Data [2019a]. *Data Illustrator*. 19th May 2019. http://data-illustrator.com/ (cited on page 5).

Illustrator, Data [2019b]. *Population Distribution by Age*. 19th May 2019. http://www.data-illustrator.com/example.php?v=waffle (cited on page 8).

Inkscape [2016]. *Inkscape*. 21st Nov 2016. https://inkscape.org/ (cited on pages 1, 16).

Kekeritz, Timm [2013]. *Infographic: Weather Radials 2013 on Behance*. 2013. https://www.behance.net/gallery/14416245/Infographic-Weather-Radials-2013 (cited on page 22).

Kim, Nam Wook, Eston Schweickart, Zhicheng Liu, Mira Dontcheva, Wilmot Li, Jovan Popovic and Hanspeter Pfister [2017]. *Data-Driven Guides: Supporting Expressive Design for Information Graphics*. IEEE Transactions on Visualization and Computer Graphics PP.99 (Jan 2017), pages 1–1. doi:10.1109/TVCG.2016.2598620 (cited on page 16).

Liu, Zhicheng, John Thompson, Alan Wilson, Mira Dontcheva, James Delorey, Sam Grigg, Bernard Kerr and John Stasko [2018]. *Data Illustrator: Augmenting Vector Design Tools with Lazy Data Binding for Expressive Visualization Authoring*. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI '18. Montreal QC, Canada: ACM, 2018, 123:1–123:13 (cited on page 5).

Microsoft [1987]. *Excel*. 1987. https://products.office.com/en-us/excel (cited on page 1).

Microsoft [2016]. *Infographics Designer for Power BI*. 2016. https://www.microsoft.com/en-us/research/project/infonice/ (cited on page 17).

Microsoft [2019]. *Power BI*. 19th May 2019. https://powerbi.microsoft.com/ (cited on pages 16–18).

Minard, Charles Joseph [1861]. *Mapping Napoleon's March*. 1861 (cited on page 19).

Nightingale, Florence [1856]. *Florence Nightingale*. 1856. https://en.wikipedia.org/wiki/Florence_Nightingale (cited on page 23).

Nightingale, Florence [2015]. *Nightingale's Rose diagram dataset*. 2015. https://github.com/datasets-io/nightingales-rose/ (cited on page 22).

Nightingale, Florence, William Farr and Andrew Smith [1859]. *A contribution to the sanitary history of the British army during the late war with Russia*. John W. Parker and Son, 1859 (cited on page 11).

NOAA [1807]. *National Oceanic and Atmospheric Administration (NOAA)*. 1807. https://www.ncdc.noaa.gov/cdo-web/search?datasetid=GHCND (cited on page 21).

Raureif [2013]. *Weather Radials Poster by Raureif*. 2013. https://www.weather-radials.com/ (cited on page 21).

Ren, Donghao [2015a]. *iVisDesigner*. 2015. https://donghaoren.org/ivisdesigner/ (cited on pages 12–13, 15).

Ren, Donghao [2015b]. *iVisDesigner Toolkit*. 2015. https://donghaoren.org/ivisdesigner/toolkit (cited on page 14).

Ren, Donghao, Bongshin Lee and Matthew Brehmer [2019]. *Charticulator: Interactive construction of bespoke chart layouts*. IEEE transactions on visualization and computer graphics 25.1 (2019), pages 789–799 (cited on page 3).

Satyanarayan, Arvind [2019]. *Vega*. 19th May 2019 (cited on page 9).

Satyanarayan, Arvind and Jeffrey Heer [2014]. *Lyra: An interactive visualization design environment*. Computer Graphics Forum. Volume 33. 3. Wiley Online Library. 2014, pages 351–360 (cited on page 9).

Victor, Bret [2013]. *Bret Victor*. 2013. https://vimeo.com/66085662 (cited on page 19).

Wikipedia.org [1861]. *Charles Joseph Minard*. 1861. https://en.wikipedia.org/wiki/Charles_Joseph_Minard (cited on page 20).

Xia, Haijun, Nathalie Henry Riche, Fanny Chevalier, Bruno De Araujo and Daniel Wigdor [2018]. *DataInk: Direct and creative data-oriented drawing*. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM. 2018, page 223 (cited on pages 14–15).