

Data Drawing Applications

Group 3 Jihad Itani, Piotr Kupiec, Emanuel Moser and Martin Sackl

706.057 Information Visualisation SS 2021
Graz University of Technology

10 May 2021

Abstract

The task to visualize data in forms everybody easily understands has become more important than ever. There exists a huge amount of data and there exist multiple tools which help users to create graphs, charts or other visualizations of their choice. However, most of them concentrate on visualizing the given data in a set of given, possible ways like bar charts or pie charts and do not provide functionality to generate non-standard charts.

This survey explores the field of data drawing applications, which make it possible to create custom, non-standard charts and is an update of the survey from a 2019 group of Information Visualization [Gostiša et al. 2019]. Special attention is paid on updates, new versions or any changes of the applications surveyed in 2019 as well as on finding application, which have not yet been surveyed in detail.

First, a definition of data drawing is stated, followed by listing several data drawing applications. Four of them will be surveyed in depth. At the end of this survey, some recommendations will be given.

© Copyright 2021 by the author(s), except as otherwise noted.

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

Contents

Contents	i
List of Figures	iii
List of Tables	v
1 Introduction	1
1.1 Definition of Data Drawing	1
1.2 Finding Data Drawing Applications	1
2 Datasets	3
2.1 Bret Victor	3
2.2 Napoleon	4
2.3 World War I	4
3 Charticulator	7
4 Lyra 2	9
4.1 Technical Aspects	9
4.1.1 User Interface	9
4.1.2 Data Processing	9
4.1.3 Exports	9
4.2 Problems	11
4.3 Opinion	11
5 Data Illustrator	13
5.1 Technical Aspects	13
5.1.1 User Interface	13
5.1.2 Data Import/Export	13
5.1.3 Usage	14
5.2 Opinion	16
6 DataToon	17
7 Conclusion	21
Bibliography	23

List of Figures

2.1	Dataset: Bret Victor	3
2.2	Dataset: Napoleon	4
2.3	Dataset: World War I	5
3.1	Charticator: User Interface	8
4.1	Lyra 2: User Interface	10
4.2	Lyra 2: Drag-and-Drop	10
4.3	Lyra 2: Data Manipulation Widget	11
5.1	Data Illustrator: User Interface	14
5.2	Data Illustrator: Repeat and Partition	15
5.3	Data Illustrator: Collection and Group.	15
5.4	Data Illustrator: Sample Visualization.	16
6.1	DataToon: User Interface	18
6.2	DataToon: Template Selection	18
6.3	DataToon: Tool Selection	18

List of Tables

1.1 Data Drawing Applications. 2

Chapter 1

Introduction

In this survey, we want to describe the field of data drawing and data drawing applications. We first start by describing, what data drawing actually is and what a possible definition of it could be. We then introduce the way of finding data drawing applications and list all of our findings. Four of these applications we will go into detail and show all important aspects. Before we introduce these four applications, we mention and describe the used datasets for generating our custom charts. At the end of the paper we will also state some honorable mentions and give some recommendations.

1.1 Definition of Data Drawing

Since there are hundreds of visualization applications as well as simple drawing applications out there, it is necessary to come up with a definition for data drawing (applications) in order to separate them accordingly. We wanted to state a definition, that matches data drawing applications, which are visualization tools as well as drawing tools. Therefore, we defined the following statements as necessary for data drawing applications:

- Data drawing applications allow users to create custom, non-standard charts.
- Data drawing applications let users specify the binding between data characteristics and visual properties manually (such properties can be width, length, colors, etc.).
- Data drawing applications respond to different datasets automatically.
- In general, data drawing applications allow users to fully customize their charts.

1.2 Finding Data Drawing Applications

Two years ago, a group already did a survey about data drawing in Information Visualization course [Gostiša et al. 2019] and covered some data drawing applications like Charticulator and Lyra. The focus of this paper is on finding out, if these tools have changed, if there are new branches, new versions or any updates or improvements. Of course, we also wanted to find new applications, not covered in the 2019 survey. Therefore, we searched through references and citations of papers from the already known tools and also used simple web search engines to find new tools. As mentioned in the previous section, it is not that easy to find applications which match the definition above. We created a list of applications we found, selected several important attributes we wanted to compare and sorted out all applications which did not match our definition of data drawing applications. Finally, we came up with the list of applications shown in Table 1.1.

Name	Open-Source	Self-Hostable	Locally-Installable	Input Data	Undo/Redo	Data Type	Surveyed	Price	SVG Export
Charticulator	Yes	Yes/No	No	CSV	Yes	Tabular	2019	free	Yes
Data Illustrator	Yes	No	No	CSV	Yes	Tabular	2019	free	Yes
Lyra2	Yes	No	Yes	JSON, CSV, TSV	Yes	Tabular	2021	free	Yes
DataToon	No	Yes	Yes	JSON, CSV, TSV	Yes	Graph	2021	free	Yes
VisComposer	No	Yes	Yes	-	-	Tabular	2021	-	Yes
DataSelfie	Yes	No	No	-	No	Tabular	2021	-	-
StructGraphics	No	Yes	Yes	JSON	Yes	Tabular	2019	free	Yes
Ivy	No	Yes	Yes	CSV	Yes	Tabular	2019	free	Yes
DataInk	No	Yes	Yes	-	-	Tabular	2021	free	-
DataQuilt	No	Yes	Yes	-	-	Tabular	2021	free	-

Table 1.1: Data drawing applications, ranked by research depth.

Chapter 2

Datasets

For our custom charts we created, our showcase videos and several other test purposes, we used three datasets. We want to describe them shortly, to get a better understanding when presenting the use of the applications in the upcoming section. The three datasets used have been given the short names: Bret Victor, Napoleon and World War I.

2.1 Bret Victor

This dataset was originally created by Bret Victor and recreated by group 5 of the class of 2019 which they kindly allowed us to use. Basically this dataset has 3 features (x, y, width) which Victor was binding in his presentation to random visual properties. The result of that binding can be seen in Figure 2.1.

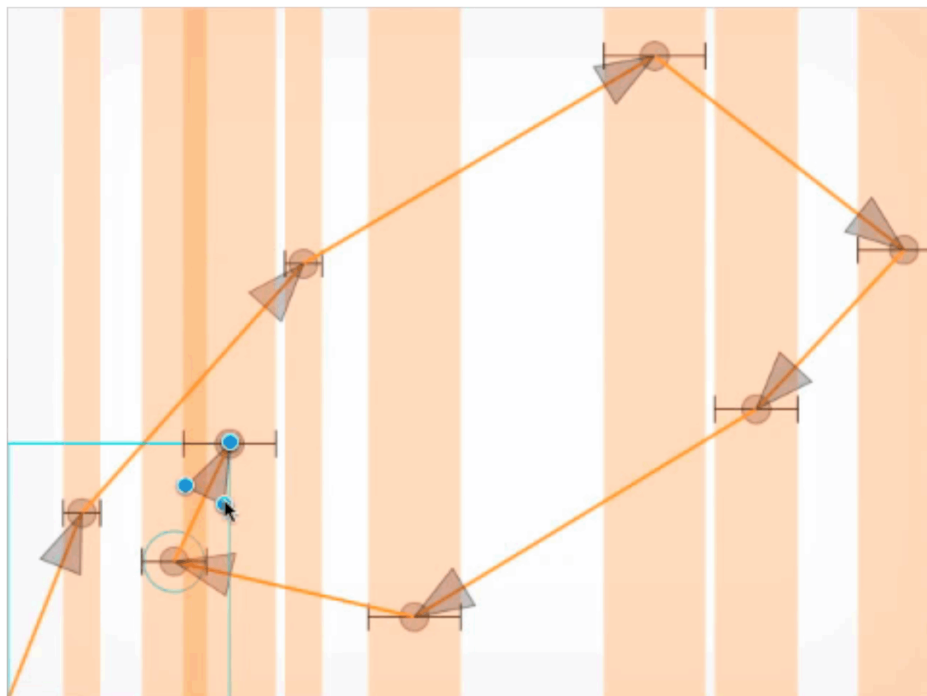


Figure 2.1: Bret Victor's dataset. [Image extracted from <https://vimeo.com/66085662>.]

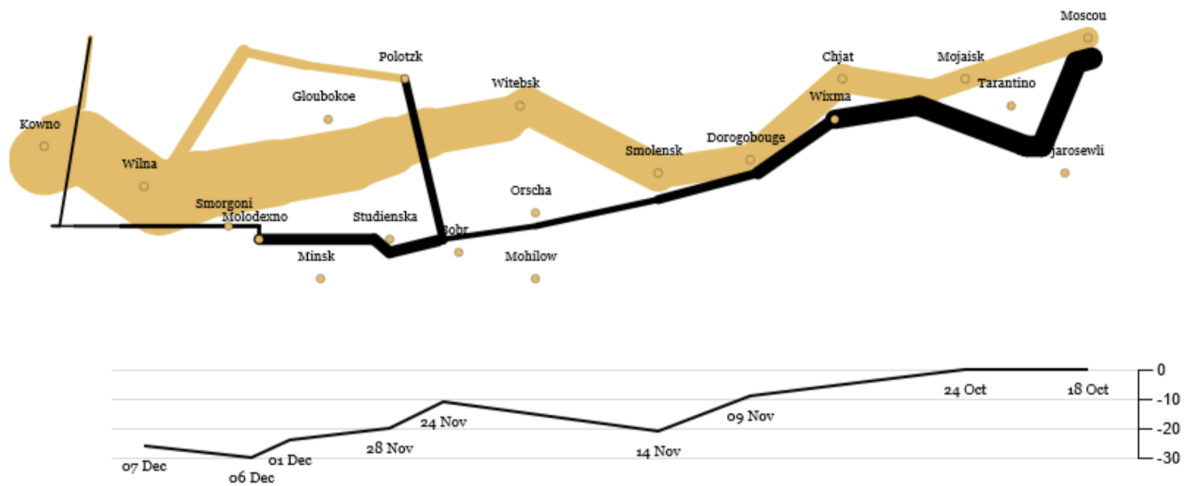


Figure 2.2: Napoleon's march. [Image created by the authors with Lyra1.]

2.2 Napoleon

Originally, this dataset and visualization was created by Charles Joseph Minard, depicting Napoleon's 1812 Russia campaign. The dataset consists of 3 CSV files:

1. `army.csv`

Describes the size of Napoleon's army at a certain longitude and latitude, if the army is advancing towards Russia or retreating. Since during this campaign the army split it is also separated into several groups.

2. `cities.csv`

Describes the longitude and latitude of certain important cities.

3. `temps.csv`

Describes the temperature at a certain longitude on a certain date.

This dataset is often used as a benchmark for data drawing/visualization programs. The visualization seen in Figure 2.2 was created as part of the showcases via Lyra 1.

2.3 World War I

This dataset depicts the alliances, tensions and aggression between the countries involved. This dataset was directly taken from <https://datatoon.github.io/> which is hosted by the authors of the corresponding paper [Kim et al. 2019]. This dataset is specifically tailored for DataToon. In contrast to the two previous datasets, this dataset describes network data not tabular data. A visualization can be seen in Figure 2.3 with further details in Chapter 6.

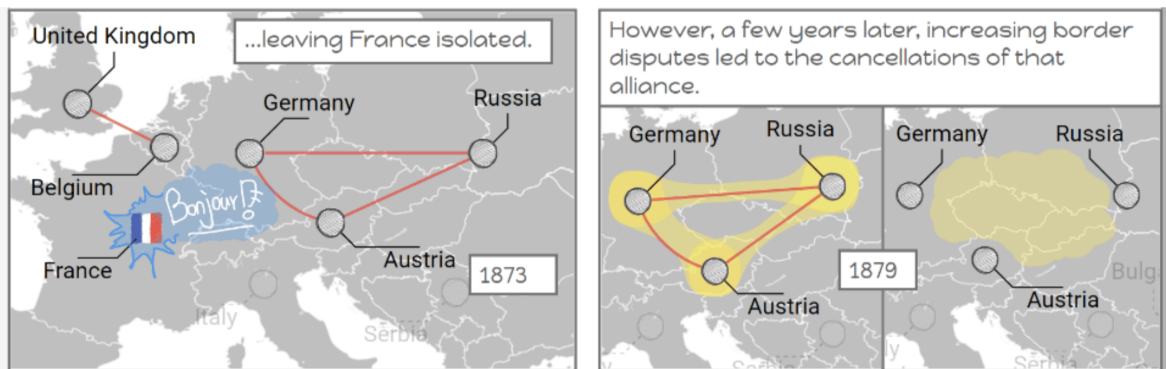


Figure 2.3: World War I. [Image extracted from DataToon <https://datatoon.github.io/datasets/ww1.json>.]

Chapter 3

Charticulator

Charticulator is a powerful, stable, interactive and easy to use data drawing application, created by Microsoft. One of the best attributes from Charticulator is that it is open source. The Microsoft Research Team provides the whole source code on GitHub [Microsoft 2021] and link to it on their website. The tool can also be launched directly from the website as a web application.

The current version of Charticulator is 2.0.3 and does not show really big changes from the version from 2019, except the a bit more modern user interface design. We can state, that all features given in the older version are also in the current version.

Our showcase video [Sackl 2021] we show a quick tutorial of Charticulator, creating a chart from the Bret's data set.

Charticulator comes with a bunch of technical features that make it easy for users to create custom charts fast. It is possible for users to import any kind of CSV dataset directly into Charticulator, immediately seeing a preview of the data in a table. On the other side, after creating a chart, the tool allows users to export their generated charts as an image (PNG, JPEG or SVG), HTML, a Charticulator Template (where the user is able to set and rename all data, axes and objects) and finally as Power BI Custom Visual.

Figure 3.1 shows the user interface of the Charticulator web application. The application consists of six panels: Fields, Scales, Glyph, Layers, Attributes, and Chart.

The Field panel at the top left shows all the available data fields, imported from a CSV file. Figure 3.1 shows the fields x , y , and $width$ from Bret's dataset. The Scales panel at the bottom left shows different scales provided for different objects within the chart. Inside the Glyph panel, the user is able to create, draw, and customize the individual data points (in this example a dot with a line and two border lines). Objects like lines, dots or rectangles can simply be dragged and dropped into this section from the menu above. A user can add as many glyphs, as he likes - in our example we created two different glyphs: one for the mentioned point and the width line, and a second one for the orange colored bars.

After adding elements, they will appear in the Layers section, where all other elements are listed as well. A user can select different layers and the Attribute panel underneath will show possible attributes of the layer/object, which the user can edit (these can be color, opacity, width, length, shape, ...). The chart itself is displayed in the large panle to the right.

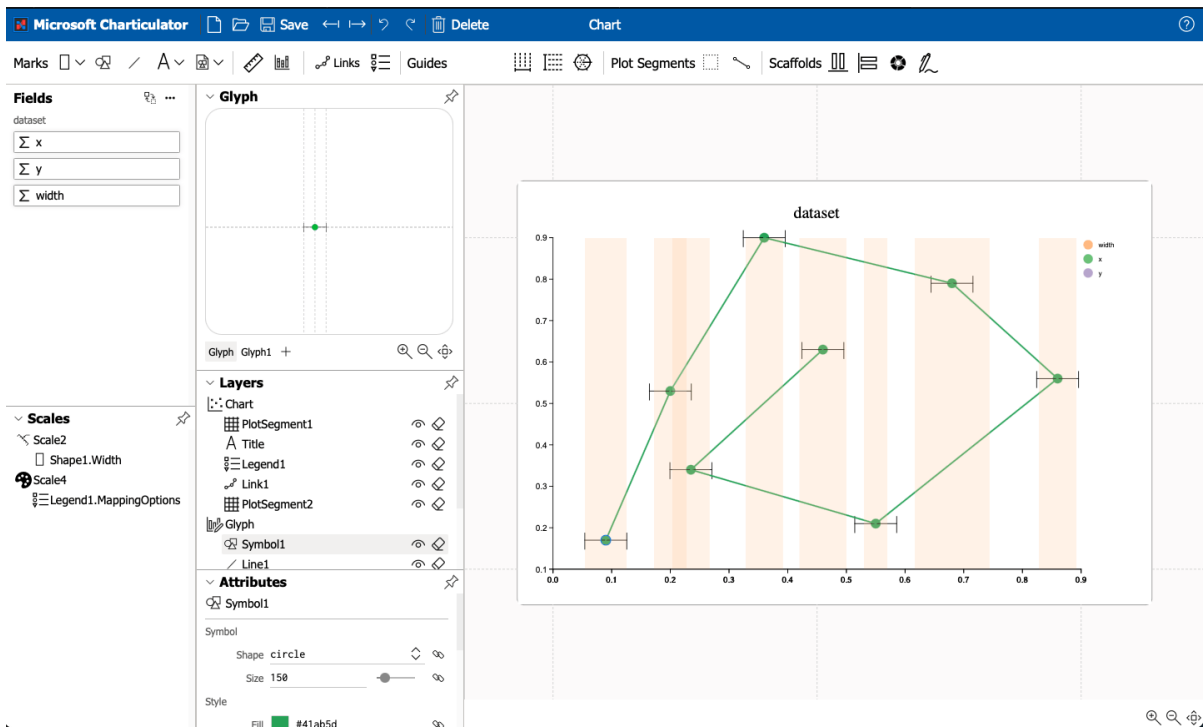


Figure 3.1: Charticator: User interface. [Image extracted from <https://charticator.com>. ©Microsoft.]

Chapter 4

Lyra 2

Lyra2 is the successor to the data drawing application Lyra1 as described in Satyanarayan and Heer [2014]. With Lyra2 a user should be able to easily generate visualizations by binding data characteristics directly to any visual properties given. In addition, it is possible to create interactive charts, for example adding a hover effect to a line chart.

Compared to Lyra1, Lyra2 got a overhaul of the UI and some added functionalities, but not all of the features of Lyra1 have been implemented as of yet. Lyra2 is still under development and it is possible to give feedback/create an issue on the GitHub page.

In our showcase video [Moser 2021], we demonstrate Lyra2 by recreating Bret Victors visualization as described in Chapter 2.

4.1 Technical Aspects

Lyra2 is a web application, but since it is open source you can either use the demo [Vega 2019] or download the source code from the respective GitHub repository [Vega 2021] and host it on a local instance.

4.1.1 User Interface

The user Interface is kept quite simple and clean, as can be seen in Figure 4.1. Lyra2 supports dragging and dropping data directly to the corresponding properties and highlights which properties are available, as can be seen in Figure 4.2.

4.1.2 Data Processing

With Lyra2 it should also be possible to do some on-demand data processing, such as filtering the dataset and only showing entries with width larger than an arbitrary value or group variables together, which would be very helpful to recreate Napoleon’s dataset as shown in Chapter 2. However, these functionalities never worked during the testing period of our survey, so we were not able to recreate the dataset.

One feature that we also want to mention is the possibility the create new features on the fly based on already existing features, in our example the width, as can be seen in Figure 4.3. As before, we were not able to type in any formula how to calculate the new data entries, we were only able to drag and drop other features there.

4.1.3 Exports

Lyra2 supports exports to popular image formats like PNG and SVG. In addition, it also supports exports as a standalone HTML page. What we found most helpful was that Lyra2 also allows to save any drawing in JSON format, which contains the complete dataset and the graph at time of export which was quite useful when encountering certain problems that will be described in the following section.

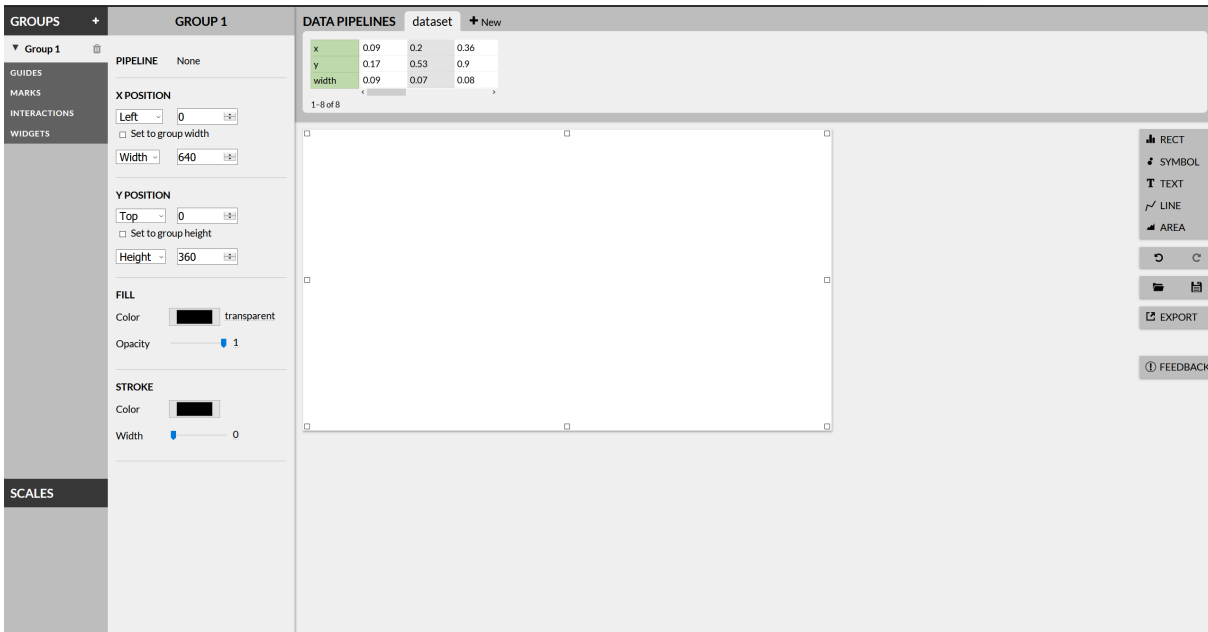


Figure 4.1: Lyra 2: User interface.



Figure 4.2: Lyra 2: User interface with drag-and-drop interaction.

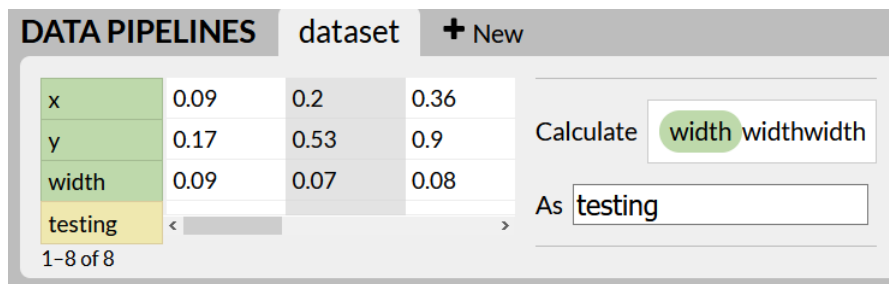


Figure 4.3: Lyra 2: User interface with data manipulation widget.

4.2 Problems

Some of the problems we encountered using Lyra2 included:

- *Limited support for custom glyphs:* It is not possible to design the visualization of data points as in Charticator or the predecessor Lyra1.
- *Limited support for visual properties:* It was not possible, for example, to bind the width in our Bret’s dataset to the width of the bars of a bar chart or the width of a line. Overall, the visual properties one could bind data to was heavily reduced compared to Lyra1.
- *Missing data processing capabilities:* In Lyra1, it was possible to filter out certain data points and apply this subset to the drawing. In Lyra2, the interface for filtering exists, but it was not possible for us to filter out any entries or even input a criterion, for example “smaller than x” into the formula field.
- *Random crashes:* Sometimes, random crashes occurred if a dataset (Bret’s) was loaded or a feature was bound to visual properties without any reason given.

4.3 Opinion

Based on the problems we encountered, we would not advise anyone to use Lyra2 in its current state. If a Lyra application needs to be used, we highly recommend using Lyra1 as it is more feature-rich and provides more stability.

Chapter 5

Data Illustrator

Data Illustrator is a research collaboration between the University of Maryland, Georgia Institute of Technology, and Adobe Systems [UMD 2018a; Liu et al. 2018]. It is a vector editing tool that allows to create data-driven visualizations and infographics. Its goal is to make visualization creation as easy as creating an illustration in a design tool with no programming skills required [UMD 2018a]. Users can use Data Illustrator to create their own visualizations by repeating and styling shapes with data-driven rules. The tool supports the creation of expressive, flexible, and parametrically defined visualizations.

As of May 2021 the latest version of the software is 1.1, which was released on 30 May 2018. Since it is a research project and no changes were done since the last release, we can conclude that it is no longer under development. There is however a possibility to ask questions, report issues and request features using a link given on the website [UMD 2018b]. Under this URL we can see that some users have indeed asked questions and creators were available to help. The following showcase video presents a simple usecase of this application[Kupiec 2021].

5.1 Technical Aspects

Data Illustrator is an online web application, which means it can be run using any modern browser, but is not self-hostable. We tested the application using Firefox 88.0 on Windows 10 Pro v19042.

5.1.1 User Interface

The user interface of Data Illustrator is clean and minimalistic, as can be seen in Figure 5.1. It consists mainly of the canvas ⑧, where the items are drawn. On the left side there are four panels: top menu bar ①, left toolbar with selection and drawing tools ③, data variables panel for importing data and seeing its summary ②, and layers panel which shows all the objects drawn on the canvas and how are they organized ④. On the bottom of the screen the data table panel is placed where the loaded datafile is shown in a table ⑤. Right side is divided into two panels - a tool bar where data can be associated with items ⑥ and property inspector where properties of objects are shown and data can be bound to these properties ⑦.

5.1.2 Data Import/Export

There are 21 demo datasets from which a user can choose, but it is also possible to use own data. The only supported data type, which can be imported is tabular data formatted as wide form .csv, and the first row of this file should specify the column names. It is also possible to import pictures, which can be placed on the canvas. When it comes to exporting created visualization the only option is to export an SVG file, or the whole project can be saved in a format specific to the application.

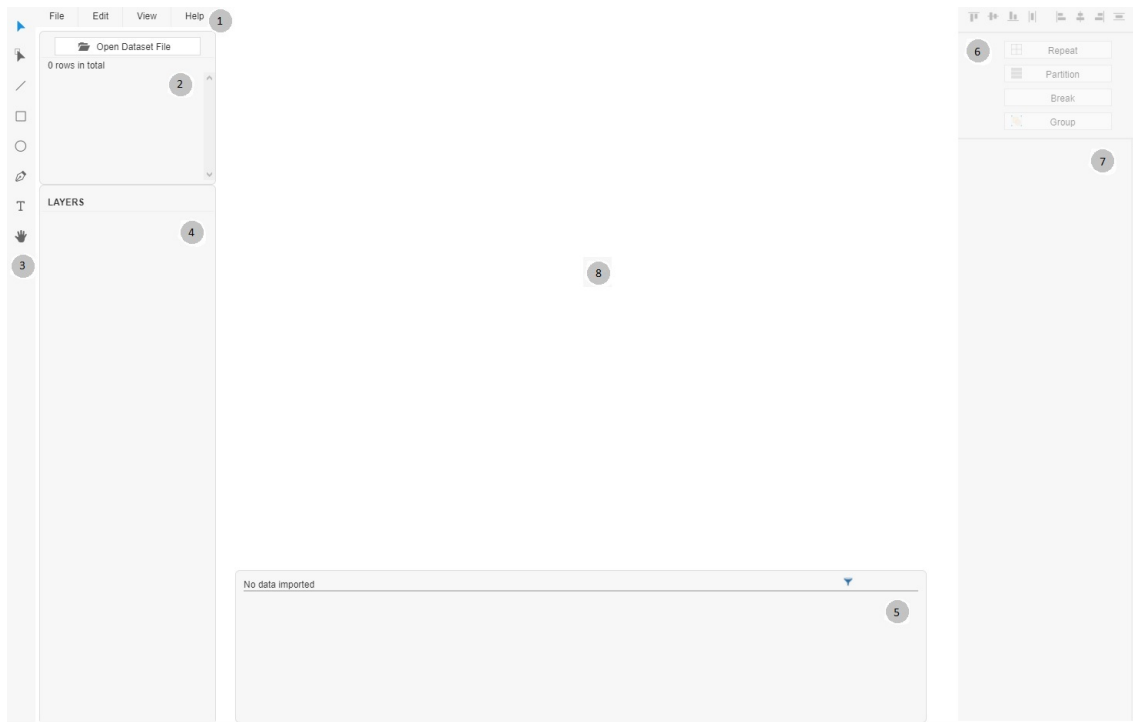


Figure 5.1: Data Illustrator: User interface.

5.1.3 Usage

To draw a shape, user can choose from three available shape tools (rectangle, ellipse or line) by clicking on the corresponding icon on the left tool bar, or by pushing a corresponding keyboard button. A shape in Data Illustrator is represented as path, just like in other vector editors. This path is a series of anchor points connected by line or curve segments. A shape can be open or closed. After drawing, the user is able to resize or reshape objects by re-positioning the anchors. In order to do it, the direct selection tool has to be used.

When working with multiple shapes, the user has an option to group or align them using the corresponding button on the right tool bar. There are two ways to generate shapes by data in Data Illustrator: repeat and partition. The detailed description of their characteristics can be seen in Figure 5.2.

For the repeat function to work, shape with a categorical data variable is needed. Any shape is allowed for repetition. The data variable must be categorical and has to contain more than one distinct values. When a categorical variable and a shape is provided, the repeat function makes multiple copies of the shape, and associate each new copy with a value and the corresponding data rows. The default effect is that shapes are put in a grid layout with two handles. These handles allow to control the number of generated objects and also how these shapes are placed in the canvas. This is achieved by dragging the handles.

In some cases, the rigidity of predefined layouts (like the ones created by repeating shapes) can limit the possibilities of visualization design. When this happens, a user can break the layout and then be able to move any object freely around.

The partition function, just like repeat, requires a shape and a categorical variable to work. When a categorical variable and a shape are given, using partition divides the shape, and associates each new part with a distinct value and the corresponding data rows. It is depended on the type of shape how a shape is divided. For example, a rectangle is partitioned into smaller rectangles, a line into multiple line segments and associate each anchor point with a value and the its data row.

A collection is a group of shapes created by a repeating or partitioning. Inside a collection, the shapes

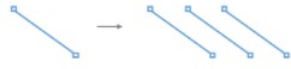





	Repeat	Partition
Concept	creates multiple copies of a shape	divides a shape into constituent parts
Shape	works for all kinds of shape	works for lines, rectangles, circles, rings and areas only
Default Layout	Repeated shapes are placed in a Grid Layout by default	Default layout depends on shape type
Example (line)		
Example (rectangle)		
Example (circle)		 (to be supported in future versions)

Figure 5.2: Data Illustrator: Repeat and partition [UMD 2018c].

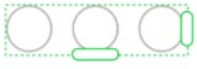



	Collection	Group
Visual Style (Selected)	 A collection is selected	 A group is selected
Visual Style (Child Selected)	 A collection child is selected	 A group child is selected
Shape Type	All the shapes in a collection have the same type (e.g. rectangle, line, path)	Shapes inside a group can have different types.
Data	All the shapes represent values of the same categorical variable. Each shape in a collection represents a unique value of the categorical variable	Shapes inside a group may or may not be data bound. If data bound, all the shapes in a group must represent the same value.

Figure 5.3: Data Illustrator: Collection and group [UMD 2018c].

are formed into a layout. Every shape from a collection are peers of each other. Because of this, an action performed on a single shape will have the same effects on all other shapes from a collection. For clarification, in Data Illustrator, the terms “group” and “collection” do not mean the same thing (Figure 5.3).

Data Illustrator allows to bind data to a visual characteristic of an object (shape, anchor point, segment or collection). A binding can be seen as a constraint on a corresponding characteristic. After the binding has been created, user can no longer change its value. After creating a data binding, Data Illustrator creates at least one numerical axis. An axis has a range control handle, which changes the scale of the visualization while dragging. Labels of axis ticks together with their number are getting updated according to the actual range of the scale.

There are several characteristics of an object to which a user can bind a value:

- X Position
- Y Position

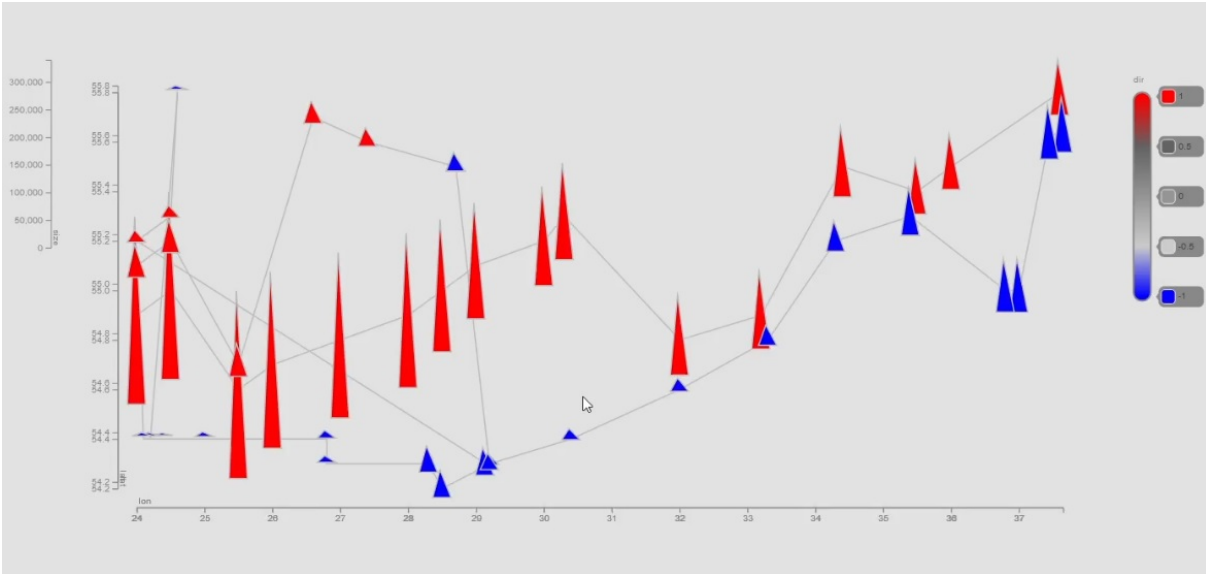


Figure 5.4: Data Illustrator: Sample visualization.

- Width
- Height
- Area
- Fill Color
- Stroke Color
- Stroke Width

5.2 Opinion

Although Data Illustrator is a research project at an early stage, it provides enough features and functionality in order to create even complex data visualizations. The authors created a comprehensive tutorial, which explains every aspect of a tool. Most of the time while using this system, it was stable and behaved in a predictive manner. But it is not bug free, especially when it comes to rescaling visualizations using axis handles. Figure 5.4 shows a sample visualization created using the Napoleon dataset.

Chapter 6

DataToon

DataToon is a drawing tool for network (graph-based) data [Kim et al. 2019; DataToon 2021], which was created due to the lack of tools out there that incorporate the idea of comics for data drawing. It introduces the concept of comic storyboarding to represent analysis and presentation with pen and touch interactions. Our showcase video [Itani 2021] shows a quick tutorial of DataToon, creating a canvas from the World War 1 dataset offered by the authors of the tool.

DataToon comes with a limited number of technical features which makes it easy for users to create custom charts. It is only possible for users to import the JSON datasets offered on the website, directly onto DataToon, immediately seeing the data on a canvas. Furthermore, the tool works best with a stylus and touch screen device; it is not mouse-friendly. Figure 6.1 shows the user interface of the DataToon web application.

The application consists of several sections:

- **Canvas:**
 - The user has the option to select either a blank canvas or one of the predefined templates with boxes for storyboarding (see Figure 6.2).
- **Settings**
 - The settings for the tool currently only ask the user whether there is a stylus or not.
- **Tool Belt**
 - The tool belt consists of 8 options in a circular menu (see Figure 6.3), namely:
 - * Drawing, as the name suggests, allows the user to draw on the canvas as they see fit. The color of the drawing and thickness of the ink can be specified in the Palette option.
 - * Eraser allows the user to erase what was drawn by either snipping around an area, or selecting the white palette and “erasing” what was drawn.
 - * Label allows the user to pop out the labels of the nodes on the canvas. The color of the labels and thickness of their lines can be set by the palette as well.
 - * Highlight allows the user to snip an area with the stylus which will highlight them using the color specified in the palette.
 - * Filter will allow the user to snip an area on the canvas and once selected, will create a new block with the area that was “filtered”.
 - * Magic allows a transition to happen, which means that it would recreate a block to cover a missing blank area on the canvas that the user drags their stylus to.

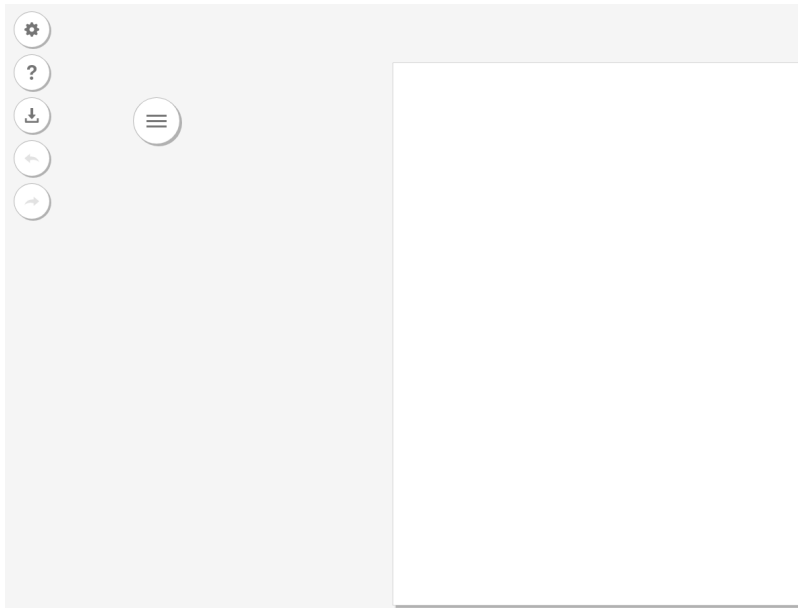


Figure 6.1: DataToon: User interface. [Image created by the authors from DataToon.]

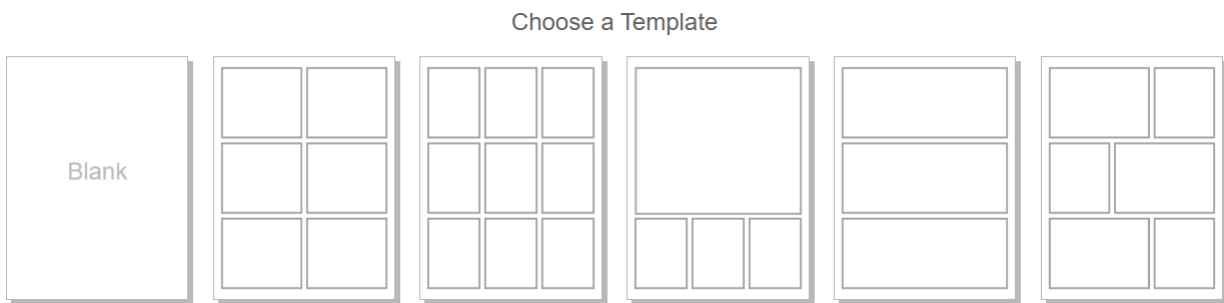


Figure 6.2: DataToon: Template selection. [Image created by the authors from DataToon.]

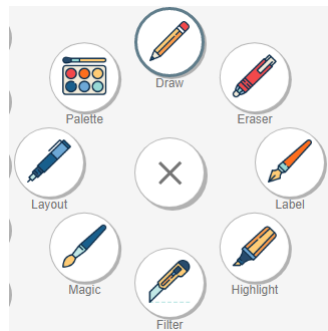


Figure 6.3: DataToon: Tool selection. [Image created by the authors from DataToon.]

- * Layout allows the user to move the items on the canvas around freely without affecting the data placed.
- * Palette as mentioned prior, is as the name suggests, contains colors for the user to select from to color their data, and line thickness.
- **Undo/Redo**
 - In the case the user wants to undo an action or bring it back.
- **Export**
 - After creating a comic, the tool allows users to export their generated results as either PNG, JSON or SVG.

Chapter 7

Conclusion

In this paper, we described the field of data drawing and listed several interesting data drawing applications. Some applications are visualization tools only and some are drawing tools only, which makes it not that easy to find real data drawing applications. We stated a definition and created a list of applications, which match this definition. Some of them we had a in-depth research and created showcase videos for.

In the sense of recommendations we would highly encourage anybody, who wants to create a non-standard custom chart to try out Charticulator from Microsoft. This tool is open source and can be launched directly from their website. It is an easy-to-handle, stable and drag-and-drop supporting online tool and provides good data import and export features. On the other hand, we can not recommend the application Lyra 2, which makes it pretty hard to use in a good way. Since the first version of Lyra was pretty promising, we thought that the new version would be a new level, but it is not indeed. Finally, DataToon is a tool for specific situations and is best used with a pen or similar tool or a good solution, if you want to create story-telling charts.

Bibliography

- DataToon [2021]. *DataToon*. 30 Apr 2021. <https://datatoon.github.io/> (cited on page 17).
- Gostiša, Karmen, Anastasiia Klimashevskaja, Jaka Konda, and Günther Moser [2019]. *Data Drawing Survey*. Graz University of Technology, 27 May 2019. <https://courses.isds.tugraz.at/ivis/surveys/ss2019/ivis-ss2019-g5-survey-data-drawing.pdf> (cited on pages I, 1).
- Itani, Jihad [2021]. *Datatoon Showcase Video*. 04 May 2021. <https://youtu.be/0z1kUoKWTsg> (cited on page 17).
- Kim, Nam Wook, Nathalie Henry Riche, Benjamin Bach, Guanpeng Xu, Matthew Brehmer, Ken Hinckley, Michel Pahud, Haijun Xia, Michael J. McGuffin, Hanspeter Pfister, et al. [2019]. *DataToon: Drawing Dynamic Network Comics with Pen + Touch Interaction*. Proc. Conference on Human Factors in Computing Systems (CHI 2019) (Glasgow, UK). ACM, 04 May 2019. doi:10.1145/3290605.3300335. <https://microsoft.com/en-us/research/uploads/prod/2019/01/datatoon.pdf> (cited on pages 4, 17).
- Kupiec, Piotr [2021]. *Data Illustrator Showcase Video*. 04 May 2021. <https://youtu.be/qmX94ibp1Do> (cited on page 13).
- 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*. Proc. Conference on Human Factors in Computing Systems (CHI 2018) (Montréal, Canada). ACM, 21 Apr 2018. doi:10.1145/3173574.3173697. <https://doi.org/10.1145/3173574.3173697> (cited on page 13).
- Microsoft [2021]. *Charticulator GitHub*. 12 May 2021. <https://github.com/Microsoft/charticulator> (cited on page 7).
- Moser, Emanuel [2021]. *Lyra2 Showcase Video*. 04 May 2021. <https://youtu.be/TWLX0zBqCI4> (cited on page 9).
- Sackl, Martin [2021]. *Charticulator Showcase Video*. 04 May 2021. <https://youtu.be/TUCnVj9W0aM> (cited on page 7).
- 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. doi:10.1111/cgf.12391. <https://idl.cs.washington.edu/files/2014-Lyra-EuroVis.pdf> (cited on page 9).
- UMD [2018a]. *Data Illustrator*. 2018. <https://data-illustrator.cs.umd.edu/about.php> (cited on page 13).
- UMD [2018b]. *Data Illustrator Forum* (2018). <https://groups.google.com/g/data-illustrator?pli=1> (cited on page 13).
- UMD [2018c]. *Data Illustrator Tutorial*. 2018. <https://data-illustrator.cs.umd.edu/tutorial.php> (cited on page 15).
- Vega [2019]. *Lyra 2 Website*. 2019. <https://vega.github.io/lyra/> (cited on page 9).

Vega [2021]. *Lyra GitHub*. 30 Apr 2021. <https://github.com/vega/lyra> (cited on page 9).