

SCImago Graphica: a new tool for exploring and visually communicating data

Yusef Hassan-Montero; Félix De-Moya-Anegón; Vicente P. Guerrero-Bote

Nota: Este artículo se puede leer en español en:
<https://revista.profesionaldelainformacion.com/index.php/EPI/article/view/87108>

Recommended citation:

Hassan-Montero, Yusef; De-Moya-Anegón, Félix; Guerrero-Bote, Vicente P. (2022). "SCImago Graphica: a new tool for exploring and visually communicating data". *Profesional de la información*, v. 31, n. 5, e310502.

<https://doi.org/10.3145/epi.2022.sep.02>

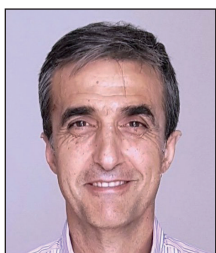
Article received on June 30th 2022
Approved on August 19th 2022



Yusef Hassan-Montero ✉
<https://orcid.org/0000-0002-4314-1175>
Universidad Internacional de La Rioja
Av. de la Paz, 137, 26006 Logroño, Spain
yusef.hassan@unir.net



Félix De-Moya-Anegón
<https://orcid.org/0000-0002-0255-8628>
SCImago Research Group
Madrid, Spain
felix.moya@scimago.es



Vicente P. Guerrero-Bote
<https://orcid.org/0000-0003-4821-9768>
Universidad de Extremadura
Plazuela Ibn Marwam, s/n
06071 Badajoz, Spain
guerrero@unex.es

Abstract

Despite the increasing number of data visualization authoring systems in recent years, it remains a challenge to simultaneously achieve high expressive power and ease of use in a single tool. In this paper we present *SCImago Graphica*, a no-code tool which allows the creation of complex visualizations by simple drag-and-drop interactions. Users bind the data variables to the different encoding channels, and specify the settings of each binding, from which the tool generates the interactive graphical display. Due to its efficiency of use, *SCImago Graphica* is not only suitable for visually communicating data, but also for exploratory data analysis. We evaluate the expressiveness and ease of use of *SCImago Graphica* through various examples of chart construction and a catalog of visualizations. The results show that *SCImago Graphica* makes it possible to create a wide variety of data visualizations quickly and easily.

Keywords

SCImago Graphica; Data visualization; Information visualization; Data communication; Data exploration; Authoring systems; No-code tools; Complex visualizations; Drag-and-drop interactions; Interactive graphs; Chart construction; Chart building; Authoring tools; Visual data communication; Exploratory analysis; Grammar-based tools; Visual mapping; Data rendering; Computer graphics; Interactive applications.

1. Introduction

In the age of big data, data visualization tools are essential to explore, understand and make sense of data. The most widely used tools for visualizing data are spreadsheet applications. In these applications the user selects the data set to be displayed, chooses which type of chart to use from a gallery, and then customize some basic aspects of the chart's appearance. Although it is a very easy-to-use interactive model, it is not without its problems. The main drawback is that it is not possible to create charts other than those in the chart gallery, which are often basic. To that, it is necessary to add that sometimes these applications include chart types that are widely considered bad practices in data visualization, such as 3D charts.

A different way of visualizing data is through textual programming, which offers authors complete control over the visual appearance and interactive behaviour of the chart, but in addition to requiring programming skills, creating a chart demands a large amount of time and effort.

It is common to describe the visualization authoring systems along two opposing dimensions: its expressiveness and its ease of use (sometimes called accessibility) (Bostock; Heer, 2009; Qin *et al.*, 2020). Expressiveness refers to the flexibility of customization and variety of visual outputs that can be created, while ease of use refers to its ease of learning and efficiency of use. Therefore, as we have seen, spreadsheet software and textual programming of visualizations would represent the two opposite extremes: a system that is easy to use but not very expressive, versus one that is very expressive but hard to use.

Over the last few years, a plethora of visualization systems (web apps, desktop applications, programming toolkits...) have emerged, occupying different spaces in the expressiveness/ease-of-use continuum. In this paper we present *SCImago Graphica*, a professional visualization authoring tool that aims to combine a high level of expressiveness with an ease-of-use interface. In addition, *SCImago Graphica* has been designed to enable both visual communication of data, as well as exploratory analysis.

2. Related work

Spreadsheets are not the only chart typology tools. Recently, modern visualization web apps have emerged that allow you to create charts with very little effort (just upload or paste the data, select a chart type, and customize it). *Datawrapper*¹ is a very popular tool among media, with which it is possible to create and publish online aesthetic and responsive charts and maps in just a few steps. Another noteworthy tool is *RAWGraphs* (Mauri *et al.*, 2017), which offers a wide gallery of visualizations, some quite sophisticated. However, as these are template-based systems, their expressiveness is limited to the chart templates in their catalogue.

Chart typologies are a restrictive and oversimplified way of thinking about graphics and graphing software, which is why Wilkinson (1999) proposes his famous *Grammar of Graphics* (*GoG*), a mathematical theory of statistical and scientific graphics. Wilkinson's *GoG* describes the fundamental principles underlying the composition of any graphic and the correct coordination of its components. The impact of *GoG* on data visualization software is undisputed, having inspired most advanced visualization systems.

Based on Wilkinson's *GoG*, Wickham (2010) proposes a layered grammar of graphics and its open-source implementation *ggplot2*, a popular package for the statistical language *R*. Implementations of low-level grammars such as *ggplot2*, but also *Protovis* (Bostock; Heer, 2009), *D3* (Bostock; Ogievetsky; Heer, 2010) or *Vega* (Satyanarayan *et al.*, 2016), have revolutionized the creation of statistical graphics using code, reducing the time and effort required without sacrificing expressiveness. More recently, high-level grammars such as *Vega-Lite* (Satyanarayan *et al.*, 2017) or *ECharts* (Li *et al.*, 2018) have been proposed, simplifying the specification of charts by being less verbose, but inevitably at the expense of some expressiveness. While all these grammar implementations have made the creation of interactive visualizations more accessible to professionals beyond engineering, specifying visualizations via imperative or declarative programming is clearly more intricate and tedious than doing so via an interactive tool.

Tableau (formerly *Polaris*) (Stolte; Tang; Hanrahan, 2002) is an interactive visualization authoring tool, inspired directly by Wilkinson's *GoG*, which has been a great commercial success. To create a chart in *Tableau*, the user only needs to bind the data attributes to be displayed with the visual encodings to be used (color, size, position...) by drag and drop. *Lyra* (Satyanarayan; Heer, 2014) (Zong *et al.*, 2020) is a tool built on top of *Vega* that, like *Tableau*, uses drag and drop to map data with visual properties. While *Tableau* is more oriented to exploratory analysis, *Lyra* offers more control over the design of the charts.

Other recent no-code visualization tools, such as *iVisDesigner* (Ren; Höllerer; Yuan, 2014), *Charticulator* (Ren; Lee; Brehmer, 2019) or *Data Illustrator* (Liu *et al.*, 2018), use interaction techniques analogous to the vector design tools, achieving expressive power comparable to programming-based systems. These tools are conceived for designing visualizations, providing the author with greater flexibility and control over the layout or marks of the chart; but their learning curve and interactive complexity do not make them suitable for data exploration.

The concept behind *SCImago Graphica* is closer to grammar-based tools, like *Tableau*, than to vector data-design tools, but with important differences in order to achieve greater expressiveness without sacrificing efficiency and ease of use.

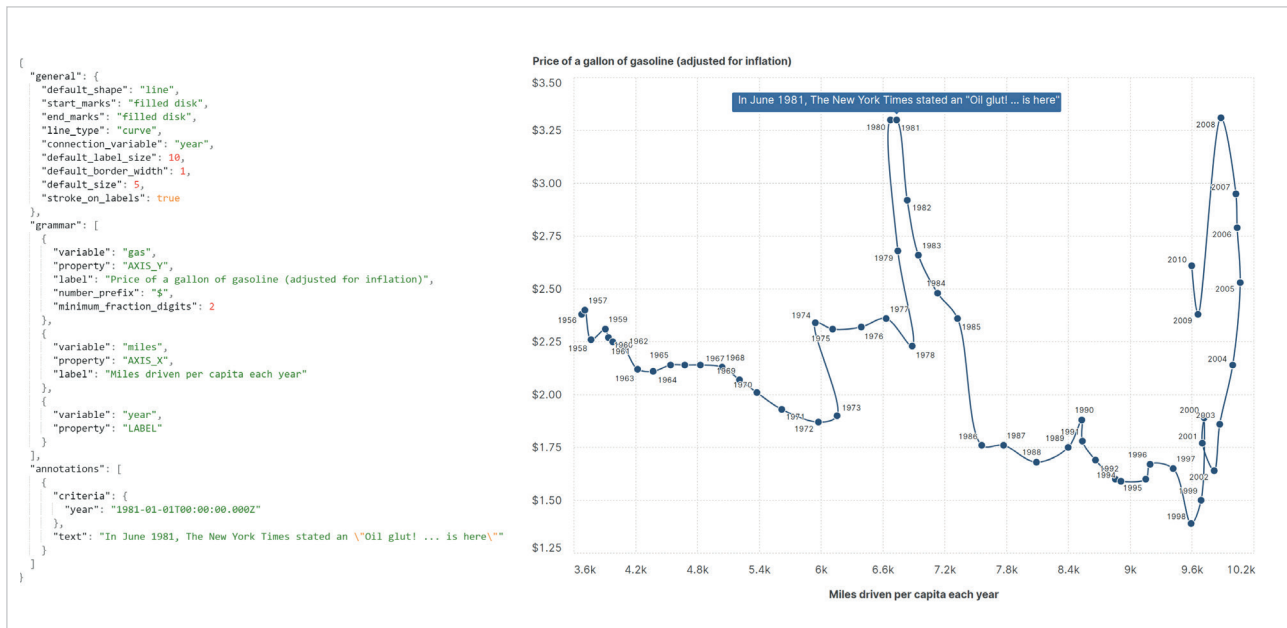


Figure 1. Grammar specification of a connected scatter plot in SCImago Graphica

3. Design

In this section we describe the fundamental components of SCImago Graphica.

3.1. Data source

Regardless of the format of the data source (CSV, Excel file...), as in other grammar-based tools like Tableau, data must be organized as Tidy Data (Wickham, 2014). In this way of organizing data, each variable must have its own column, each row represents an observation, and each cell must contain a single value. In addition, the type of each variable (number, string, date-time, or country) must be specified with the data source. The country variable type enables the creation of data maps using the name or ISO code of each country (more granular geographic types will be added in the future).

SCImago Graphica can be downloaded from:
<https://graphica.app>

One of the main differences between SCImago Graphica and other grammar-based tools is that it can work with network data as input. In network data, one subset of the data describes the nodes and their attributes, and another subset describes the relationships or links between nodes and their attributes. Although SCImago Graphica uses its own CSV-based format for network data, the desktop application includes parsers for the most common formats: GML (Graph Modelling Language), GraphML and GEXF (Graph Exchange XML Format).

3.2. Grammar specification

In SCImago Graphica, the grammatical specification of each chart employs a JSON representation. Each definition has three sections: General, Grammar and Annotations (see Figure 1). It is not mandatory to define all of them in every chart, only the needed ones.

In the General section all overall properties of the chart are defined, such as the type of mark (see Table 1), the margins, the color palette, the form of edges in the case of graphs (see Figure 2), among many other options.

The Grammar section is where data variables are mapped to encoding properties. This mapping is defined as an array of objects (where the order matters), each of which binds an encoding property to a variable. Each variable-property binding object can also include specific settings on how the mapping between them should be done: aggregation function, scale, sorting, interactive filters, etc.

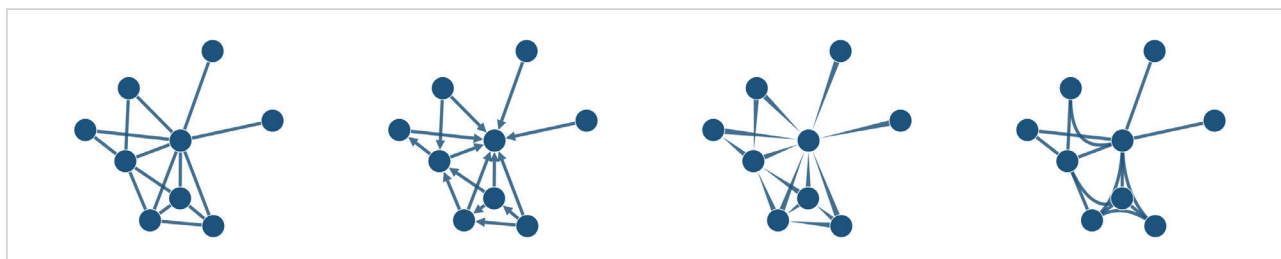
The Grammar section is not only used to encode data variables through visual encoding channels in the sense of Bertin's retinal variables (Bertin, 1983), such as position, size, color, opacity (alpha) and shapes. It is also used to encode variables by means of labels or tooltips: to define filtering rules; or to specify which categorical variables dictate the different symbols to be displayed, or how these symbols should be subdivided.

Finally, the Annotations section is used to define textual annotations attached to all those symbols in the chart that match the specified criteria.

The grammar specification in SCImago Graphica has certain similarities to Vega-Lite's unit plots specification (Santyanarayan et al., 2017), as both are simple and readable. The most notable difference is that in SCImago Graphica more

Table 1. Types of marks that can be defined in *SCImago Graphica* by property “default_shape”

●	“filled disk”	
○	“disk”	
▮	“bar”	
—	“line”	The “line_type” property allows the values \sim “none”, \sim “curve” and \perp “orthogonal”.
▲	“area”	The “line_type” property allows the values \sim “none” and \sim “curve”.
■	“filled rectangle”	
🌐	“geo”	
🕒	“circle”	The “circle_chart_type” property allows the values 🕒 “pie”, 🕒 “rose” and 🕒 “radar”.

Figure 2. Different types of edges defined by the “edge_form” property. From left to right: line, arrow, tapered and bundling, the last one based on the work of Wallinger *et al.* (2022).

than one data variable can be bound to the same encoding channel.

3.3. Visualization generation engine

The generation engine is the core of *SCImago Graphica*, the component that, from a data source and a JSON grammar specification, renders an interactive and responsive visualization. It is written from scratch in JavaScript, allowing it to run in the browser, on the server and even as part of a standalone desktop application (see 3.4). By default, the output is drawn using SVG (*Scalable Vector Graphics*), but it can also display interactive charts on *Canvas/WebGL* in combination with the *PixiJS* library².

The first task that the generation engine handles is the grammar check. For example, it checks the compatibility between the type of data variables and the encoding channels used, or if an encoding channel has been bound to more variables than it supports.

The engine also performs all data transformation tasks (aggregations, data binning, filtering...) as well as statistical computations derived from the grammar specification, such as clustering –using an algorithm based on **Clauset, Newman and Moore** (2004)–, regression analysis, or network metrics and statistical measures calculation.

Another key task of the generation engine is the computation of the chart layout, which depends on multiple factors. First, it is conditioned by the type of mark chosen; dots and bars, for example, are not visually positioned and arranged in the same way. Second, it is determined by the combination of variables that have been bound to positional visual properties (X-axis, Y-axis and small multiples). As can be seen from the examples in the Figure 3, *SCImago Graphica* shows great flexibility in how these combinations can be done. But there is a third factor that conditions the position of each symbol on the chart: the layout algorithm chosen. The default algorithm simply aligns the symbols in successive rows, and the “Avoid overlap” algorithm moves the symbols iteratively until none of them overlaps with the others. But it is when working with network data that the range of options increases considerably (Figure 4):

- Force Directed, based on **Fruchterman and Reingold** (1991);
- Force Directed+Distances, a variation of the Force Directed algorithm that in addition uses the shortest path distance between nodes in computing the forces of repulsion between them;
- Kamada and Kawai’s algorithm for undirected graphs (**Kamada; Kawai**, 1989);
- *LinLog*, an algorithm that uses the energy model of **Noack** (2007), which is particularly useful for depicting clustering

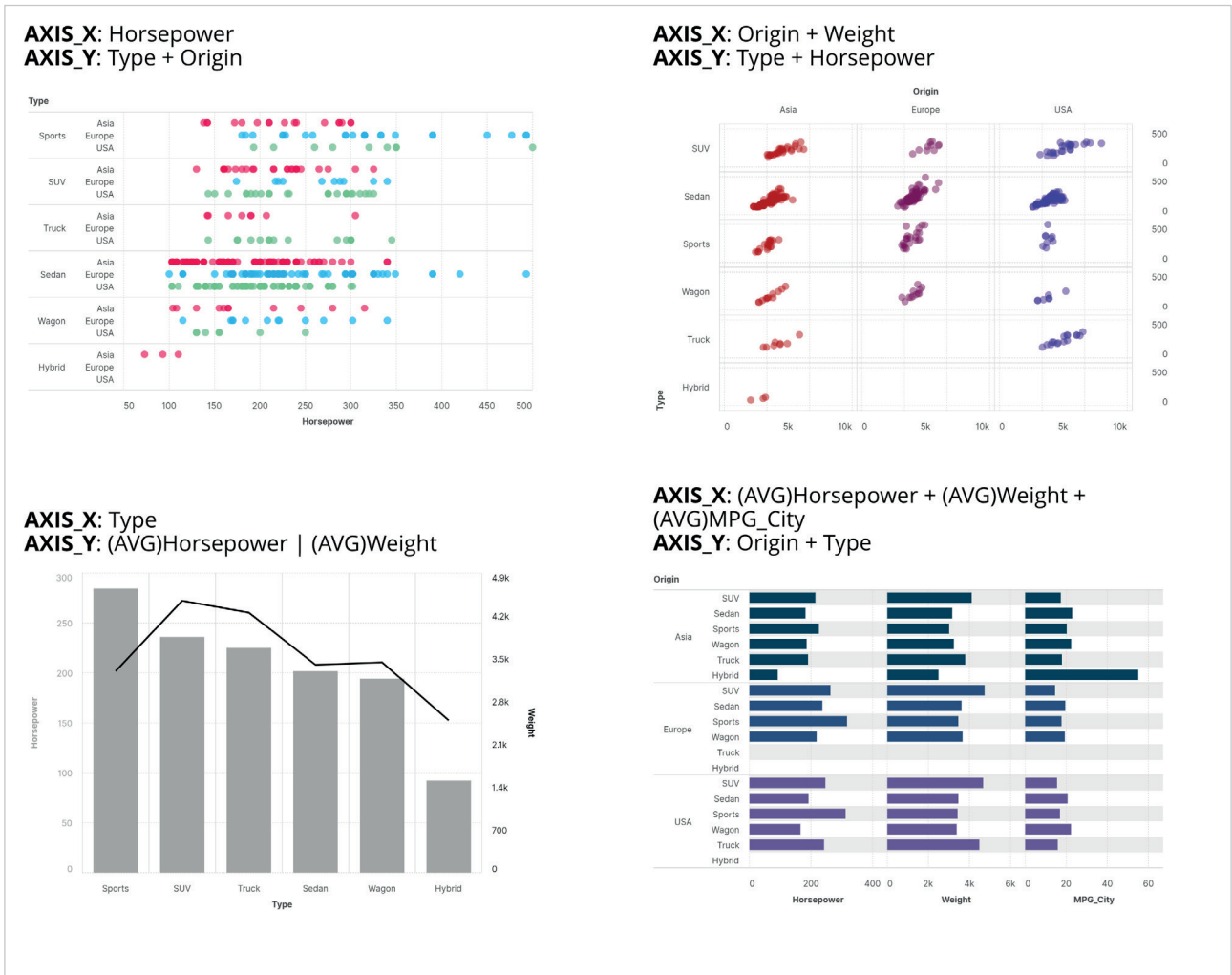


Figure 3. Examples of layouts as a result of using different combinations of variables and positional properties

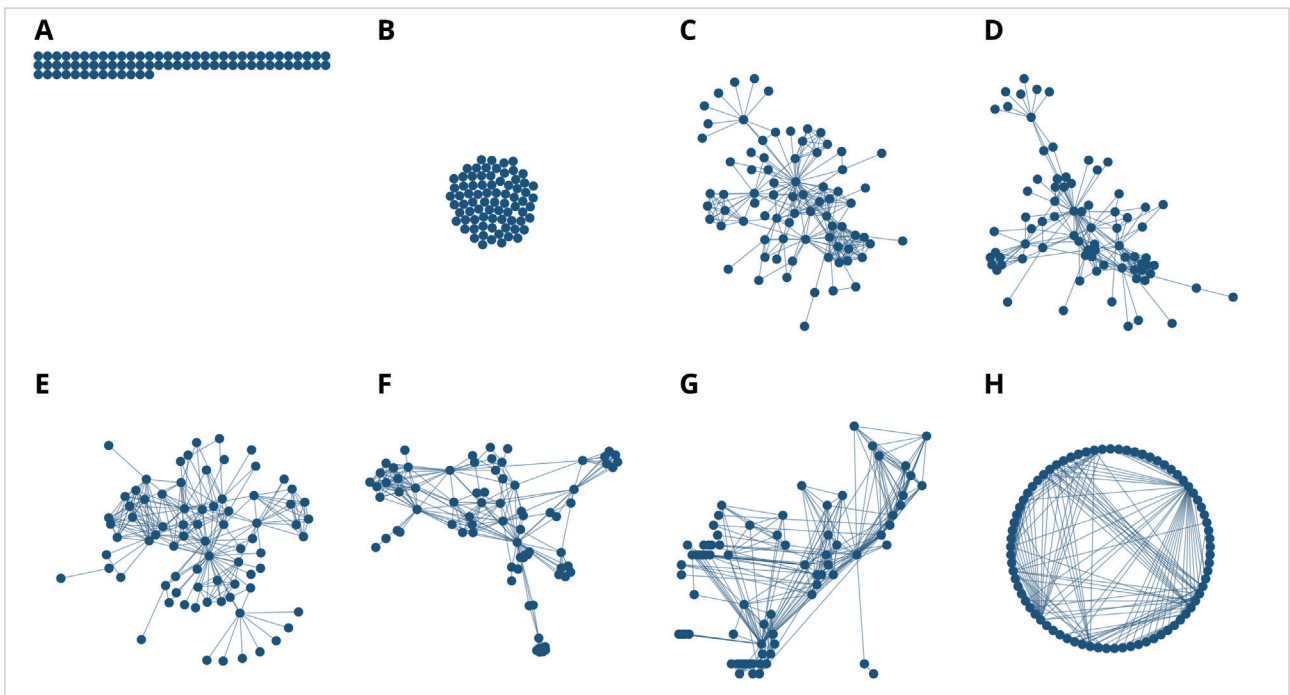


Figure 4. Main layout algorithms in SCImago Graphica: A) Default. B) Avoid Overlapping. C) Force Directed. D) Force Directed+Distances. E) Kamada and Kawai. F) LinLog. G) Dagre Top-Down. H) Circular.

- relationships through the positions and distances between nodes (**Noack, 2009**);
- *Dagre Top-Down* and *Dagre Left-Right*, that are hierarchical layout algorithms³; and
- *Circular*, a simple algorithm that places the nodes of a graph on a circle.

Layout algorithms can be combined with fixed positions, connecting the X or Y position to a quantitative axis, as shown in Figure 5.

The generated visualizations, in addition to being responsive –dynamically adapting to the available width and height–, can be interactive. Interactions that can be specified in *SCImago Graphica* include tooltips, zoom and panning, hyperlinks, highlighting when hovering, and interactive filtering.

3.4. User interface

To make data exploration and visualization simple and effortless with *SCImago Graphica*, a standalone desktop application was built. The *Electron* framework⁴ was used, which allowed the fast development of a cross-platform application (*Windows, MacOS and Linux*) using JavaScript, HTML, and CSS.

As **Gammel, Tory and Storey (2010)** note, one of the common barriers in the data visualization process is the choice of which data variables answer the goals and questions that the user is trying to address. For this reason, after loading the dataset, a small chart with the distribution of each variable is displayed in the header of each column of the data table (Figure 6), helping the user to become familiar with the data.

The application enables the creation of multiple visualizations from the same data source, organized in tabs (Figure 7.1).

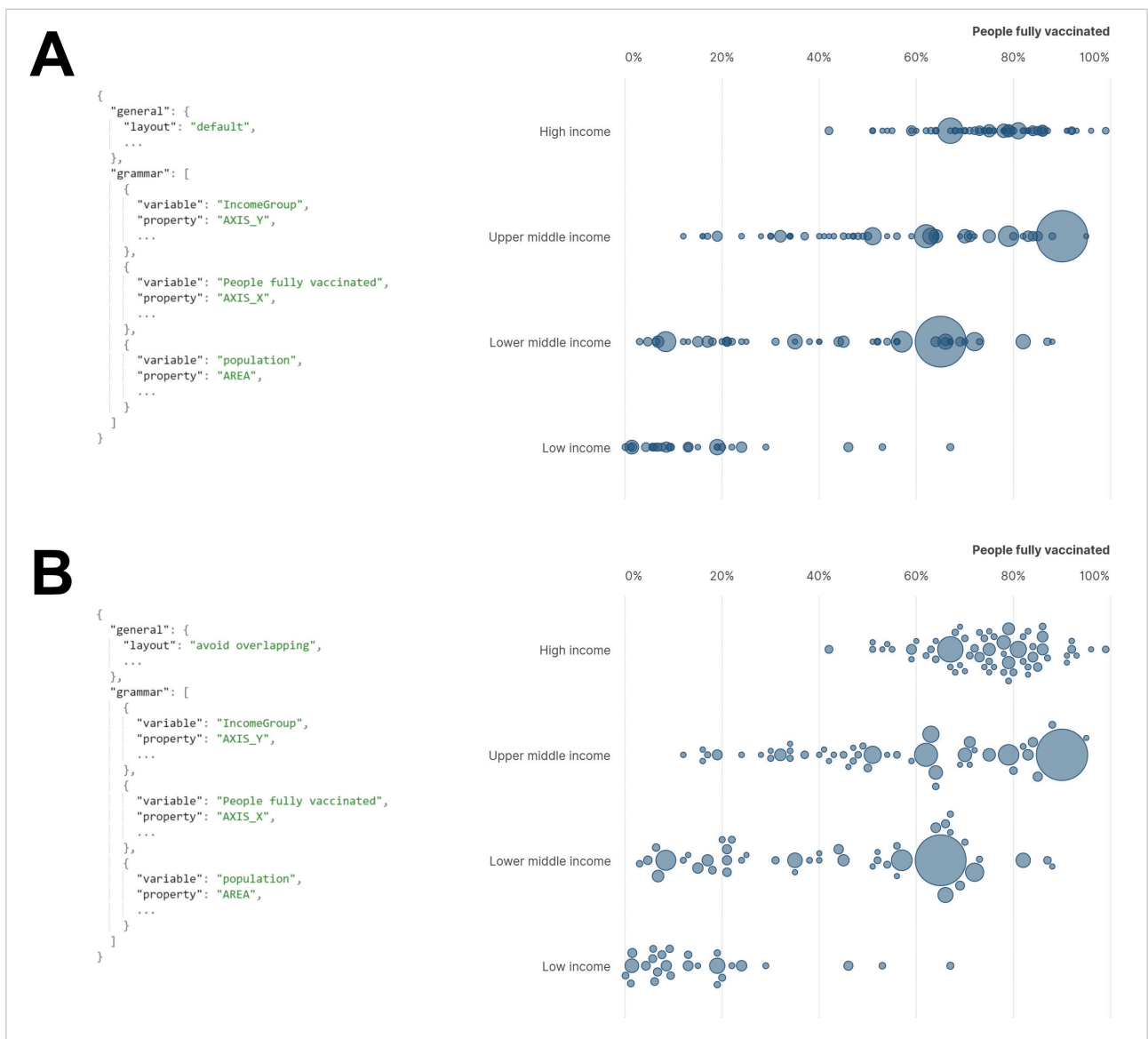


Figure 5. A) Example of layout by positional properties. B) Example of layout by positional properties combined with overlap avoiding algorithm. The Y position of each symbol is given by the categorical variable “IncomeGroup” and by the layout algorithm application, whereas X position is exclusively given by the quantitative variable “People fully vaccinated”.

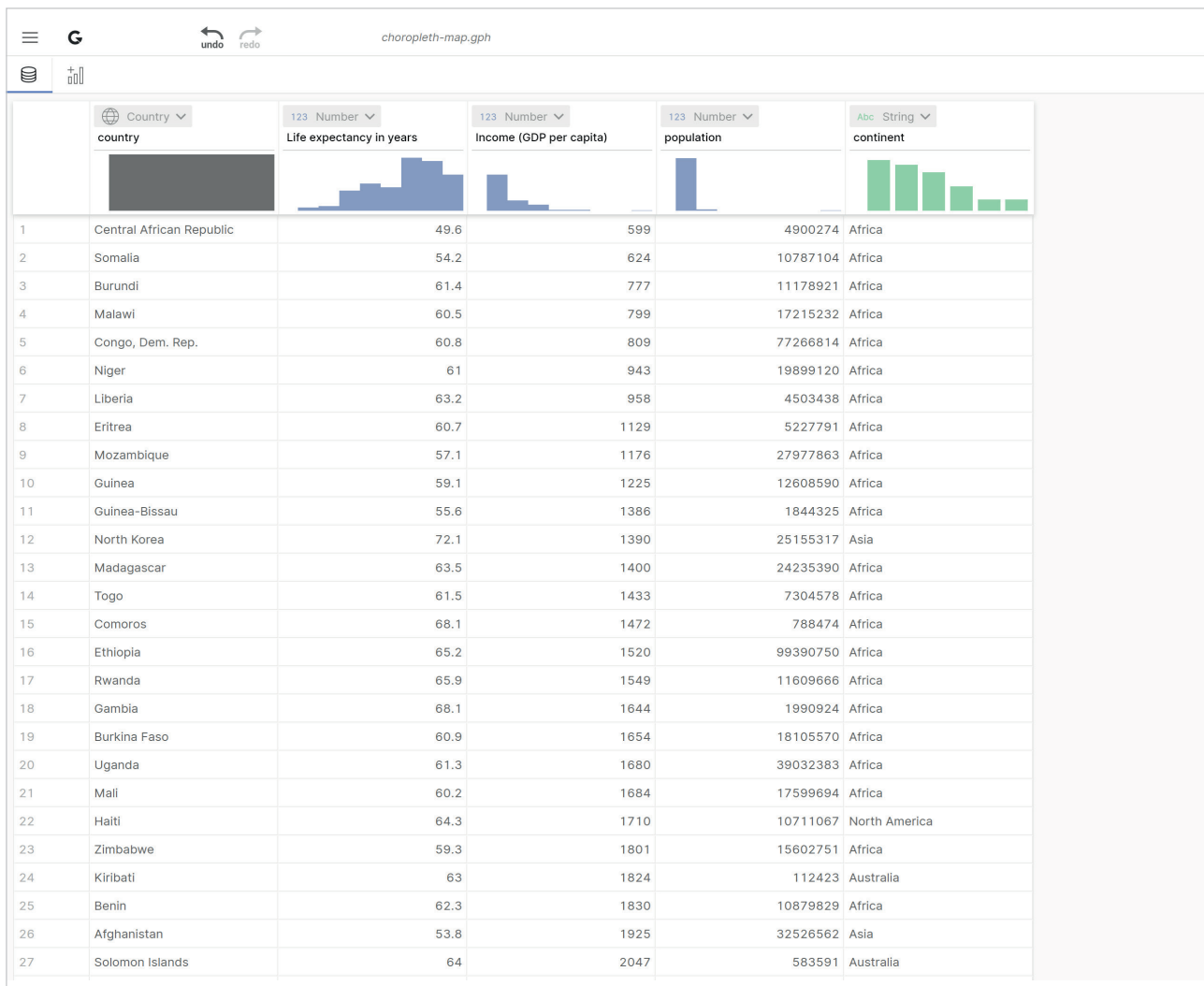


Figure 6. Initial screen of SCImago Graphica after loading the data

The interactive model for composing a chart is similar to other grammar-based tools: the user binds data variables (Figure 7.2) to encoding shelves (Figure 7.3) by drag-and-drop. By clicking on the variables dropped on each shelf, specific properties (aggregation function, format, legend visibility, etc.) can be modified. The visualization is displayed in the central area (Figure 7.4), in which the user can add annotations, edit titles, or resize the chart in an interactive way. Lastly, the right sidebar (Figure 7.5) allows the user to define all the general properties of the visualization.

The designed interface not only supports quick design and composition of charts, but also exploration and understanding of the data. The user can easily bind variables to coding channels, switch variables from one coding shelf to another, or undo and redo actions.

All charts created with the application can be exported to PNG, SVG or to the HTML+JavaScript code needed to be published online as an interactive and responsive data visualization. The generated code, in addition to the required libraries and styles, includes the JSON grammar specification of the chart.

4. Evaluation

As stated in the objectives of this work, our proposed tool seeks to achieve both a high degree of expressiveness and ease of use, qualities that are evaluated below.

4.1. Ease of use

Although it may seem that the most objective way to compare the ease of use of a visualization authoring tool with the others is through a comparative study, the reality is that it has serious limitations (Ren *et al.*, 2018). Visualization authoring tools are complex systems, differing in their design philosophy, interaction models, underlying technology, features supported or target audience. By comparing completion times or counting interactions, for example, conclusions can only be drawn in relation to the specific task evaluated, but never about the overall usability of the tools.

For this reason, in this paper we have opted to analyze some of the key differences of SCImago Graphica with regard

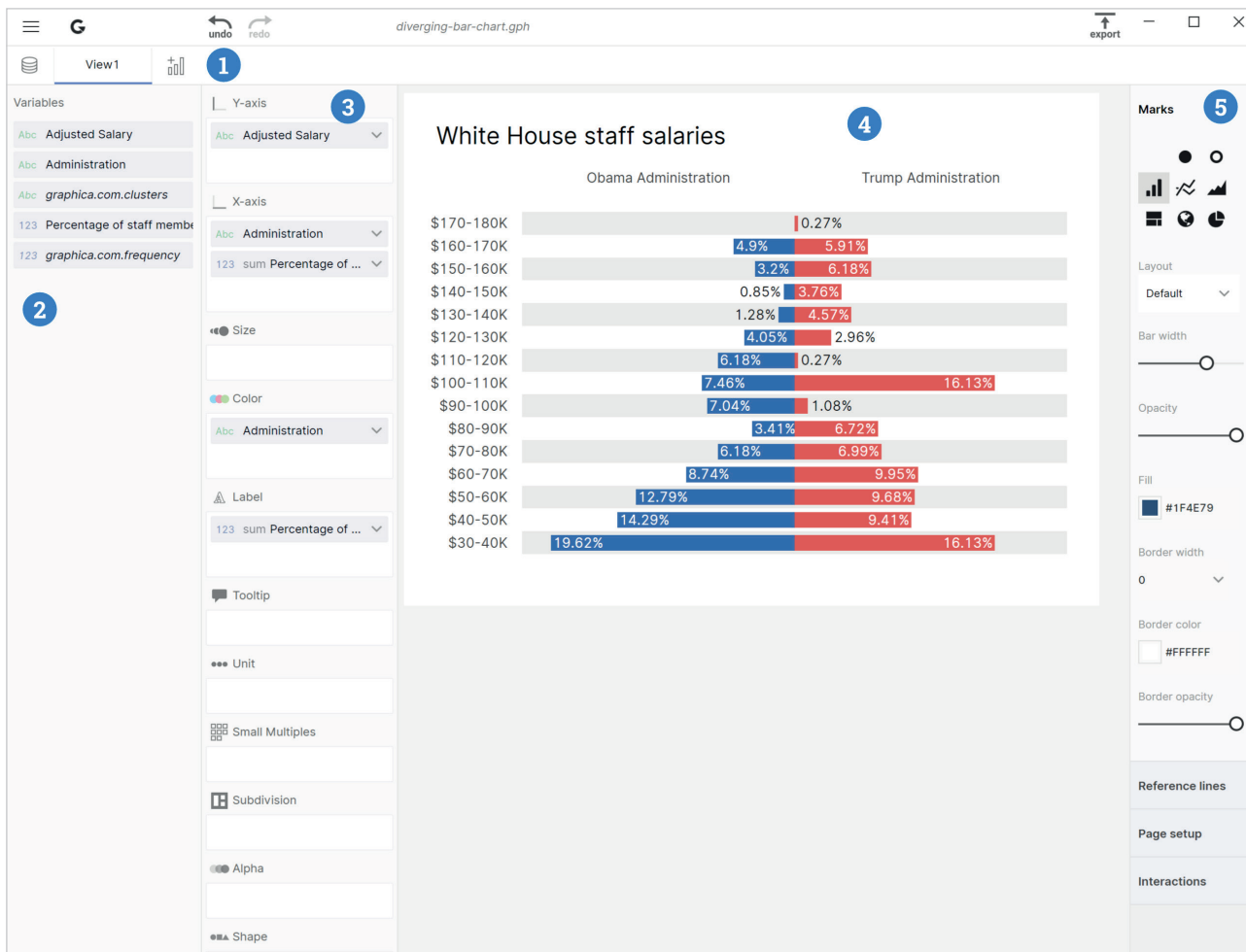


Figure 7. SCImago Graphica interface for data visualization by visual mapping

to the other tools, through a series of examples. This evaluation should be seen as a first approach to understand the fundamental characteristics of the tool and its underlying complexity.

The following examples in Figure 8 are discussed below:

A) Histograms

To analyze the distribution of a quantitative variable, one of the most used charts are histograms. To create a histogram in SCImago Graphica, a user only has to perform three actions: select 'bar' as the mark type, bind the quantitative variable to the X-axis, and bind the variable "graphica.com.frequency" to the Y-axis. This last variable is calculated by the application, which refers to the count or number of occurrences. In this example the user does not need to specify that a data binning should be performed, nor what should be the bin width, as these operations are automatically performed by the tool.

B) Small multiple

In small multiple displays, data are presented in the form of small charts of the same type that share the same scale. This way of breaking down the data display facilitates quick visual comparisons among categories or time periods. Despite its benefits, grammar-based tools usually require the user to select a categorical variable for the rows, and another one for the columns, variables that may not exist in the dataset, and therefore must be created expressly to achieve the desired layout. In SCImago Graphica, by contrast, it is enough to drop a categorical or date variable on the "Small Multiple" shelf, and the tool will subdivide the available space to fit the different small charts. This simple way of splitting the data displays is particularly useful for exploratory data analysis.

C) Parallel coordinate

Parallel coordinate plots are of huge value for multivariate data analysis because of their ability to reveal correlations, similarities, or anomalies in the data. For this reason, it is surprising that in the most popular data visualization tools there is no simple way to create them. In the case of SCImago Graphica, it is only needed to bind as many variables as desired to visualize to the same axis (X-axis or Y-axis) and to select the line as the type of mark.

Although the examples analyzed do not allow an overall assessment of the usability of SCImago Graphica, they do pro-

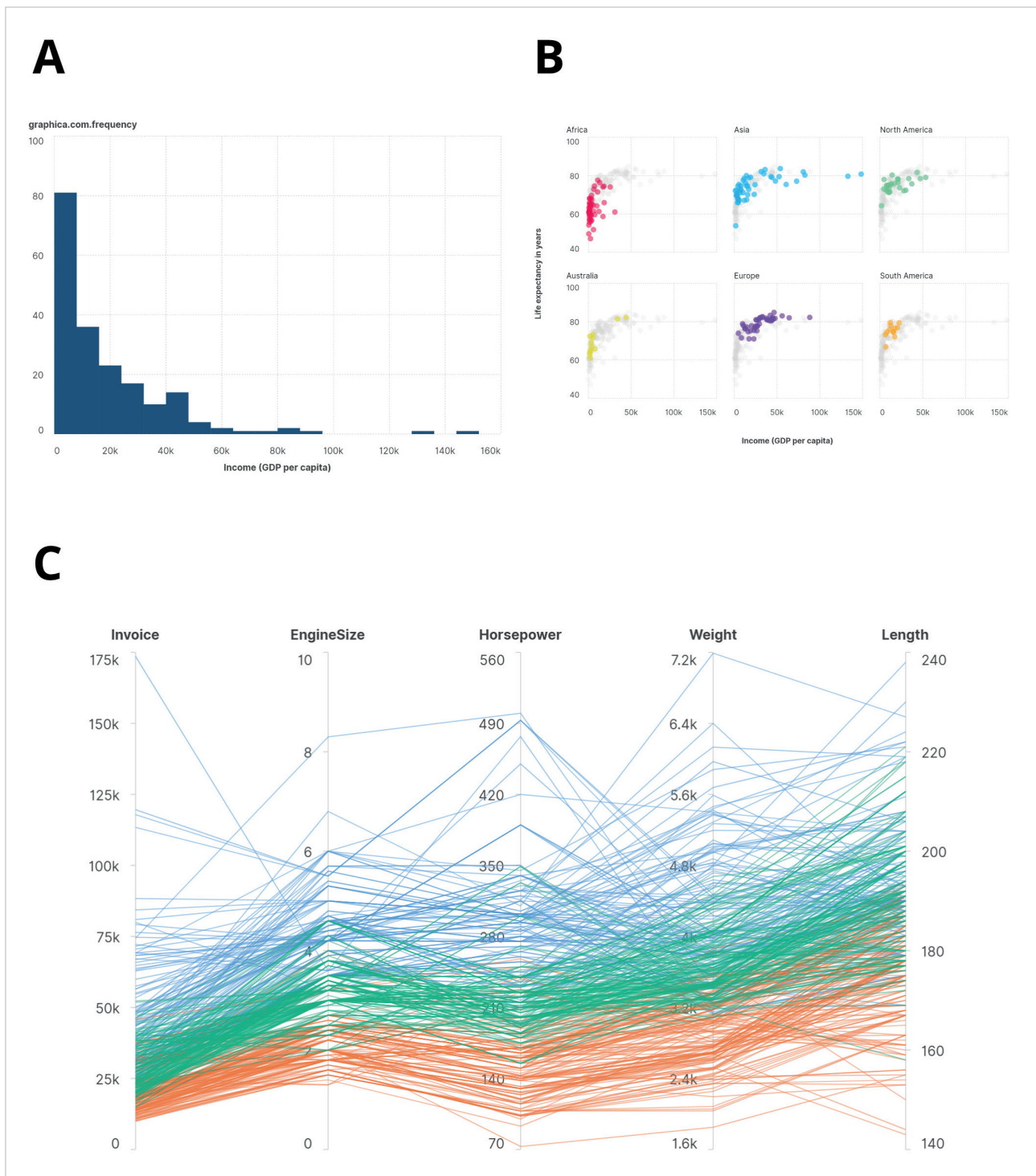


Figure 8. A) Histogram. B) Small multiple. C) Parallel coordinate

vide insight into how the tool tries to anticipate the user’s intention and reduce the number of actions required.

4.2. Expressiveness

The scope of possible design configurations that a grammar-based allows cannot be captured through user studies, so **Ren et al. (2018)** recommend, as a way of evaluating the expressiveness of visualization tools, providing a gallery of varied examples of charts created with them. As the same authors point out, one of the benefits of providing these galleries is that they can serve as a means of comparing expressiveness between different tools.

We provide a *Data Viz Catalog* in progress, which can be accessed online⁵. The examples in the online catalog are classified by function (comparisons, distribution, correlation, part-to-whole, evolution, map and network), which makes it easier to find those that enable the desired type of analysis. As can be seen (Figure 9), the examples include from basic



Figure 9. Data Viz Catalog

and frequent chart types, such as bar charts, to more complex ones, such as contiguous cartograms. Given that *SCImago Graphica* is a grammar-based tool, not a template-based tool, the examples in the catalog represent only a small fraction of all its combinatorial possibilities.

As our Data Viz Catalog show, although *SCImago Graphica* does not reach the level of expressiveness of low-level programming grammars, it is more expressive in several respects than other interactive grammar-based tools. Its capacity to visualize network data, or the flexibility of its layouts, are some of its most outstanding points.

5. Discussion and future work

We have presented a novel data visualization authoring tool, which can be used for both visual data communication and exploratory analysis. The expressive power of the tool has been demonstrated through a catalog of data visualizations, while its ease of use has been inspected through several examples of chart building. Obviously, the usability of the tool requires further inquiry. Although the revised examples illustrate the tool's capability to create advanced charts with few interactions, both user studies and analysis of usage in real-world scenarios will be necessary to further understand other usability attributes.

While drag-and-drop visual mapping enables to create data visualizations easily and efficiently, it can be a barrier for novice users (Grammel; Tory; Storey, 2010). The solution in these cases may be the integration of a chart recommendation feature. 'Show me' (Mackinlay; Hanrahan; Stolte, 2007) suggests chart types compatible with the variables the user selects to display. *Keshif* (Yalçın; Elmqvist; Bederson, 2018) automatically generates exploration-oriented dashboards with the least user effort. *Voyager* (Wongsuphasawat et al., 2015), on the other hand, is a faceted browsing system of recommended charts chosen on the basis of perceptual and statistical measures.

We believe that an important function that should also be addressed by these chart recommendation systems is to shorten the learning curve of grammar-based tools. Therefore, the future recommendation system to be integrated in *SCImago Graphica* will not only be designed to simplify the creation of charts for novices, but also to help them learn how to reach these solutions through visual mapping.

6. Notes

1. <https://github.com/datawrapper/datawrapper>
2. <https://pixijs.com>
3. <https://github.com/dagrejs/dagre>
4. <https://www.electronjs.org>
5. <https://graphica.app/catalogue>

7. References

- Bertin, Jacques** (1983). *Semiology of graphics*. Madison, Wisconsin: The University of Wisconsin Press. Translated by W. J. Berg. ISBN: 978 0 299090609
- Bostock, Michael; Heer, Jeffrey** (2009). "Protovis: A graphical toolkit for visualization". *IEEE Transactions on visualization and computer graphics* (Proc. InfoVis), 2009.
<https://doi.org/10.1109/TVCG.2009.174>
- Bostock, Michael; Ogievetsky, Vadim; Heer, Jeffrey** (2010). "D3: Data-driven documents". *IEEE Transactions on visualization and computer graphics*, v. 17, n. 12, December 2011.
<http://vis.stanford.edu/papers/d3>
- Clauset, Aaron; Newman, Mark E. J.; Moore, Christopher** (2004). "Finding community structure in very large networks". *Physical review E*, v. 70, 066111.
<https://journals.aps.org/pre/abstract/10.1103/PhysRevE.70.066111>
- Fruchterman, Thomas M. J.; Reingold, Edward M.** (1991). "Graph drawing by force-directed placement". *Software - practice & experience*, v. 21, n. 11, pp. 1129-1164.
<https://doi.org/10.1002/spe.4380211102>
- Gammel, Lars; Tory, Melanie; Storey, Margaret-Anne** (2010). "How information visualization novices construct visualizations". *IEEE Transactions on visualization and computer graphics*, v. 16, n. 6, pp. 943-952.
<https://doi.org/10.1109/TVCG.2010.164>
- Kamada, Tomihisa; Kawai, Satoru** (1989). "An algorithm for drawing general undirected graphs". *Information processing letters*, v. 31, n. 1, pp. 7-15.
[https://doi.org/10.1016/0020-0190\(89\)90102-6](https://doi.org/10.1016/0020-0190(89)90102-6)
- Li, Deqing; Mei, Honghui; Shen, Yi; Su, Shuang; Zhang, Wenli; Wang, Junting; Zu, Ming; Chen, Wei** (2018). "ECharts: A declarative framework for rapid construction of web-based visualization". *Visual informatics 2*, pp. 136-146.
<https://doi.org/10.1016/j.visinf.2018.04.011>
- Liu, Zhicheng; Thompson, John; Wilson, Alan; Dontcheva, Mira; Delorey, James; Grigg, Sam; Kerr, Bernard; Stasko, John** (2018). "Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring". *CHI 2018*, April 21-26, Montréal, QC, Canada.
<https://doi.org/10.1145/3173574.3173697>
- Mackinlay, Jock D.; Hanrahan, Pat; Stolte, Chris** (2007). "Show me: Automatic presentation for visual analysis". *IEEE Transactions on visualization and computer graphics*, v. 13, n. 6.
<https://doi.org/10.1109/TVCG.2007.70594>
- Mauri, Michele; Elli, Tommaso; Caviglia, Giorgio; Uboldi, Giorgio; Azzi, Matteo** (2017). "RAWGraphs: A visualisation platform to create open outputs". *CHIItaly'17*. September 18-20, Cagliari, Italy.
<https://doi.org/10.1145/3125571.3125585>
- McNeill, Graham; Hale, Scott A.** (2019). "Viz-Blocks: Building visualizations and documents in the browser". *EuroVis 2019 - Short papers*. The Eurographics Association.
<https://diglib.org/handle/10.2312/evs20191177>
- Noack, Andreas** (2007). "Energy models for graph clustering". *Journal of graph algorithms and applications*, v. 11, n. 2, pp. 453-480.
<https://doi.org/10.7155/jgaa.00154>
- Noack, Andreas** (2009). "Modularity clustering is force-directed layout". *Physical review E*, n. 79, 026102.
<https://doi.org/10.1103/PhysRevE.79.026102>
- Qin, Xuedi; Luo, Yuyu; Tang, Nan; Li, Guoliang** (2020). "Making data visualization more efficient and effective: a survey". *The VLDB Journal*, v. 29, pp. 93-117.
<https://doi.org/10.1007/s00778-019-00588-3>
- Ren, Donghao; Höllerer, Tobias; Yuan, Xiaoru** (2014). "iVisDesigner: Expressive interactive design of information visualizations". *IEEE Transactions on visualization and computer graphics*, v. 20, n. 12, pp. 2092-2101.
<https://doi.org/10.1109/TVCG.2014.2346291>
- Ren, Donghao; Lee, Bongshin; Brehmer, Matthew** (2019). "Charticulator: Interactive Construction of Bespoke Chart Layouts". *IEEE Transactions on visualization and computer graphics* (InfoVis 2), v. 25, n. 1.
- Ren, Donghao; Lee, Bongshin; Brehmer, Matthew; Riche, Nathalie-Henry** (2018). "Reflecting on the evaluation of visu-

alization authoring systems: Position paper". *IEEE Evaluation and beyond - Methodological approaches for visualization (BELIV)*, pp. 86-92.

<https://doi.org/10.1109/BELIV.2018.8634297>

Satyanarayan, Arvind; Heer, Jeffrey (2014). "Lyra: An interactive visualization design environment". *Computer graphics forum*, v. 33, pp. 351-360. Wiley Online Library.

<https://doi.org/10.1111/cgf.12391>

Satyanarayan, Arvind; Moritz, Dominik; Wongsuphasawat, Kanit; Heer, Jeffrey (2017). "Vega-Lite: A grammar of interactive graphics". *IEEE Transactions on visualization and computer graphics* (Proc. IEEE InfoVis).

<https://doi.org/10.1109/tvcg.2016.2599030>

Satyanarayan, Arvind; Russell, Ryan; Hoffswell, Jane; Heer, J. (2016). "Reactive Vega: A streaming dataflow architecture for declarative interactive visualization". *IEEE Trans. visualization & computer graphics* (Proc. InfoVis).

<http://vis.csail.mit.edu/pubs/reactive-vega.pdf>

Stolte, Chris; Tang, Diane; Hanrahan, Pat (2002). "Polaris: A system for query, analysis, and visualization of multidimensional relational databases". *IEEE Transactions on visualization and computer graphics*, v. 8, n. 1, pp. 52-65.

<https://doi.org/10.1109/2945.981851>

Wallinger, Markus; Archambault, Daniel; Auber, David; Nöllenburg, Martin; Peltonen, Jaakko (2022). "Edge-path bundling: A less ambiguous edge bundling approach". *IEEE Transactions on visualization and computer graphics*, pp. 313-323, v. 28, n. 1.

<https://doi.org/10.1109/TVCG.2021.3114795>

Wickham, Hadley (2010). "A layered grammar of graphics". *Journal of computational and graphical statistics*, v. 19, n. 1, pp. 3-28.

<https://doi.org/10.1198/jcgs.2009.07098>

Wickham, Hadley (2014). "Tidy data". *Journal of statistical software*, v. 59, n. 10.

<https://doi.org/10.18637/jss.v059.i10>

Wilkinson, Leland (1999). *The grammar of graphics*. New York: Springer. ISBN: 978 0 387245447

Wongsuphasawat, Kanit; Moritz, Dominik; Anand, Anushka; Mackinlay, Jock; Howe, Bill; Heer, Jeffrey (2015). "Voyager: Exploratory analysis via faceted browsing of visualization recommendations". *IEEE Transactions on visualization and computer graphics*, v. 22, n. 1, pp. 649-658.

<https://doi.org/10.1109/TVCG.2015.2467191>

Yalçın, Mehmet-Adil; Elmqvist, Niklas; Bederson, Benjamin B. (2018). "Keshif: Rapid and expressive tabular data exploration for novices". *IEEE Transactions on visualization and computer graphics*, v. 24, n. 8, pp. 2339-2352.

<http://users.umiaccs.umd.edu/~elm/projects/keshif/keshif.pdf>

Zong, Jonathan; Barnwal, Dhiraj; Neogy, Rupayan; Satyanarayan, Arvind (2020). "Lyra 2: Designing interactive visualizations by demonstration". *IEEE Transactions on visualization and computer graphics*, v. 27, n. 2, pp. 304-314.

<https://doi.org/10.1109/TVCG.2020.3030367>



<http://www.profesionaldelainformacion.com>

Welcome to **EPI**
International scientific journal

e-ISSN: 1699-2407

<https://doi.org/10.3145/EPI>

JIF2021=3.596 (Q2)

Bimonthly journal on Information & Communication

SJR2021=0.831 (Q1)