SCImago Graphica: una nueva aplicación para explorar y comunicar datos visualmente

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

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

Note: This article can be read in its English original version on:
https://revista.profesionaldelainformacion.com/index.php/EPI/article/view/87108

Cómo citar este artículo.
Este artículo es una traducción. Por favor cite el original inglés:

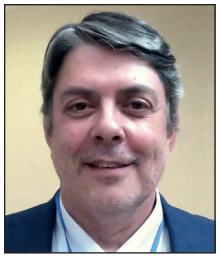
Hassan-Montero, Yusef; **De-Moya-Anegón, Félix**; **Guerrero-Bote, Vicente P.** (2022). "*SCImago Graphica*: a new tool for exploring and visually communicating data". *Profesional de la información*, v. 31, n. 5, e310502.

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

Artículo recibido el 30-06-2022
Aceptación definitiva: 19-08-2022

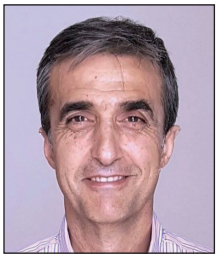
Yusef Hassan-Montero ✉
https://orcid.org/0000-0002-4314-1175

Universidad Internacional de La Rioja
Av. de la Paz, 137
26006 Logroño, España
yusef.hassan@unir.net



Félix De-Moya-Anegón
https://orcid.org/0000-0002-0255-8628

SCImago Research Group
Madrid, España
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, España
guerrero@unex.es

Resumen

A pesar del creciente número de sistemas de creación de visualización de datos en los últimos años, sigue siendo un desafío combinar en un mismo sistema un alto poder expresivo y facilidad de uso. En este artículo presentamos *SCImago Graphica*, una aplicación no-code (el usuario no necesita programar) que permite la creación de visualizaciones complejas mediante interacciones simples de arrastrar y soltar. Los usuarios vinculan las variables de datos a los diferentes canales de codificación y definen las opciones específicas de cada vinculación, y el software genera la visualización gráfica interactiva a partir de esa información. Gracias a su eficiencia de uso, *SCImago Graphica* no solo es adecuada para la comunicación visual de datos, sino también para el análisis exploratorio de datos. Evaluamos la expresividad y facilidad de uso de *SCImago Graphica* a través de varios ejemplos de construcción de gráficos y un catálogo de visualizaciones. Los resultados muestran que *SCImago Graphica* permite crear una amplia variedad de visualizaciones de datos de forma rápida y sencilla.

Palabras clave

SCImago Graphica; Visualización de datos; Visualización de información; Comunicación de datos; Exploración de datos; Sistemas de autoría; Herramientas no-code; Visualizaciones complejas; Interacciones de arrastrar y soltar; Gráficos interactivos; Elaboración de gráficos; Comunicación visual de datos; Herramientas basadas en gramática; Mapeo visual; Representación de datos; Gráficos de ordenador; Aplicaciones interactivas.

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. Introducción

En la era de los datos masivos, las aplicaciones de visualización de datos son esenciales para explorar, comprender y dar sentido a los datos. Las más utilizadas son las hojas de cálculo, en las que el usuario selecciona el conjunto de datos a representar, elige qué tipo de gráfico usar de una galería y luego personaliza algunos aspectos básicos de la apariencia del gráfico. Aunque es un modelo interactivo muy fácil de usar, no está exento de problemas. El principal inconveniente es que no es posible crear gráficos distintos de los de la galería de gráficos, que suelen ser básicos. A eso hay que añadir que en ocasiones estas aplicaciones incluyen tipos de gráficos que son considerados malas prácticas en la visualización de datos, como los gráficos 3D.

Una forma diferente de visualizar datos es a través de la programación textual, que ofrece al autor un control total sobre la apariencia visual y el comportamiento interactivo del gráfico, aunque además de requerir habilidades de programación, la creación de un gráfico exige una gran cantidad de tiempo y esfuerzo.

Es común describir los sistemas de visualización en dos dimensiones opuestas: su expresividad y su facilidad de uso (a veces llamada accesibilidad) (**Bostock**; **Heer**, 2009; **Qin** *et al.*, 2020). La expresividad se refiere a la flexibilidad de personalización y la variedad de resultados visuales que se pueden crear, mientras que la facilidad de uso se refiere a su facilidad de aprendizaje y eficiencia de uso. Por lo tanto, como hemos visto, las hojas de cálculo y la programación textual de visualizaciones representarían los dos extremos opuestos: un sistema fácil de usar pero poco expresivo, *versus* uno muy expresivo pero difícil de usar.

En los últimos años han surgido multitud de sistemas de visualización (aplicaciones web, aplicaciones de escritorio, kits de herramientas de programación...) que ocupan diferentes espacios en el continuo expresividad/facilidad de uso. En este artículo presentamos *SCImago Graphica*, una herramienta profesional de creación de visualizaciones de datos que tiene como objetivo combinar un alto nivel de expresividad con una interfaz fácil de usar. Además, *SCImago Graphica* ha sido diseñado para permitir tanto la comunicación visual de datos como el análisis exploratorio.

2. Trabajos relacionados

Las hojas de cálculo no son los únicos programas basados en tipología de gráficos. Recientemente, han surgido modernas aplicaciones web de visualización que permiten crear gráficos con muy poco esfuerzo (simplemente cargar o pegar los datos, seleccionar un tipo de gráfico y personalizarlo). *Datawrapper*[1] es una aplicación muy popular entre los medios de comunicación, con la que es posible crear y publicar online gráficos y mapas estéticos y responsivos (se adaptan dinámicamente al ancho y alto disponibles) en solo unos pocos pasos. Otro software destacable es *RAWGraphs* (**Mauri** *et al.*, 2017), que ofrece una amplia galería de visualizaciones, algunas bastante sofisticadas. Sin embargo, al tratarse de sistemas basados en plantillas, su expresividad se limita a las plantillas de gráficos que forman su catálogo.

Las tipologías de gráficos son una forma restrictiva y demasiado simplificada de enfocar los gráficos y sus softwares, razón por la cual **Wilkinson** (1999) propuso su famosa *Gramática de gráficos* (*GoG)*, una teoría matemática de gráficos estadísticos y científicos. La *GoG* de Wilkinson describe los principios fundamentales que subyacen a la composición de cualquier gráfico y la correcta coordinación de sus componentes. El impacto de *GoG* en el software de visualización de datos es indiscutible, ya que ha inspirado los sistemas de visualización más avanzados.

Basado en la *GoG* de Wilkinson, **Wickham** (2010) propuso una gramática de gráficos en capas y su implementación de código abierto *ggplot2*, un paquete popular para el lenguaje estadístico *R*. Implementaciones de gramáticas de bajo nivel como *ggplot2*, y también *Protovis* (**Bostock**; **Heer**, 2009), *D3* (**Bostock**; **Ogievetsky**; **Heer**, 2010) o *Vega* (**Satyanarayan** *et al.*, 2016), han revolucionado la creación de gráficos estadísticos que precisan programación, reduciendo el tiempo

y el esfuerzo necesarios sin sacrificar la expresividad. Más recientemente, gramáticas de alto nivel como *Vega-Lite* (**Satyanarayan** *et al.*, 2017) o *ECharts* (**Li** *et al.*, 2018), han simplificado la especificación de los gráficos, haciéndola menos verbosa, pero inevitablemente a expensas de cierta expresividad. Si bien todas estas implementaciones gramaticales han hecho que la creación de visualizaciones interactivas sea más accesible para profesionales más allá de la ingeniería, especificar visualizaciones a través de programación imperativa o declarativa es claramente más complejo y tedioso que hacerlo a través de una herramienta interactiva.

Tableau (anteriormente *Polaris*) (**Stolte**; **Tang**; **Hanrahan**, 2002) es un programa de creación de visualizaciones interactivas, inspirado directamente en la *GoG* de Wilkinson, que ha tenido un gran éxito comercial. Para crear un gráfico en *Tableau*, el usuario solo necesita vincular los atributos de datos a representar con las codificaciones visuales que se utilizarán (color, tamaño, posición...) arrastrando y soltando. *Lyra* (**Satyanarayan**; **Heer**, 2014) (**Zong** *et al.*, 2020) está construido sobre *Vega* y, al igual que *Tableau*, usa arrastrar y soltar para mapear datos con propiedades visuales. Si *Tableau* está más orientado al análisis exploratorio, *Lyra* ofrece un mayor control sobre el diseño de los gráficos.

Otros programas de visualización recientes no-code (en los que el usuario no necesita programar), como *iVisDesigner* (**Ren**; **Höllerer**; **Yuan**, 2014), *Charticulator* (**Ren**; **Lee**; **Brehmer**, 2019) o *Data Illustrator* (**Liu** *et al.*, 2018), emplean técnicas de interacción análogas a los herramientas de diseño vectorial, consiguiendo una capacidad expresiva equiparable a los sistemas basados en programación. Están concebidos para el diseño de visualizaciones, brindando al autor mayor flexibilidad y control sobre la composición visual o las marcas del gráfico, pero su curva de aprendizaje y complejidad interactiva no los hacen adecuados para la exploración de datos.

El enfoque de *SCImago Graphica* está más cerca de los programas basados en gramática, como *Tableau*, que de los de diseño vectorial de datos, pero con importantes diferencias para lograr una mayor expresividad sin sacrificar la eficiencia y la facilidad de uso.

3. Diseño

En esta sección describimos los componentes básicos de *SCImago Graphica.*

3.1. Fuente de datos

SCImago Graphica puede descargarse de:
https://graphica.app

Independientemente del formato de la fuente de datos (CSV, archivo *Excel*...), al igual que en otras herramientas basadas en la gramática, los datos deben organizarse como *Tidy Data* (**Wickham**, 2014). En esta manera de organizar los datos, cada variable debe tener su propia columna, cada fila representa una observación y cada celda debe contener un solo valor. Además, el tipo de cada variable (número, cadena de caracteres, fecha y hora, o país) debe especificarse con la fuente de datos. La variable país permite la creación de mapas de datos utilizando el nombre o el código *ISO* de cada país (en el futuro se agregarán tipos geográficos más granulares).

Una de las principales diferencias entre *SCImago Graphica* y otras aplicaciones basadas en gramática es que puede trabajar con datos de red como entrada. En los datos de red, un subconjunto de datos describe los nodos y sus atributos, y

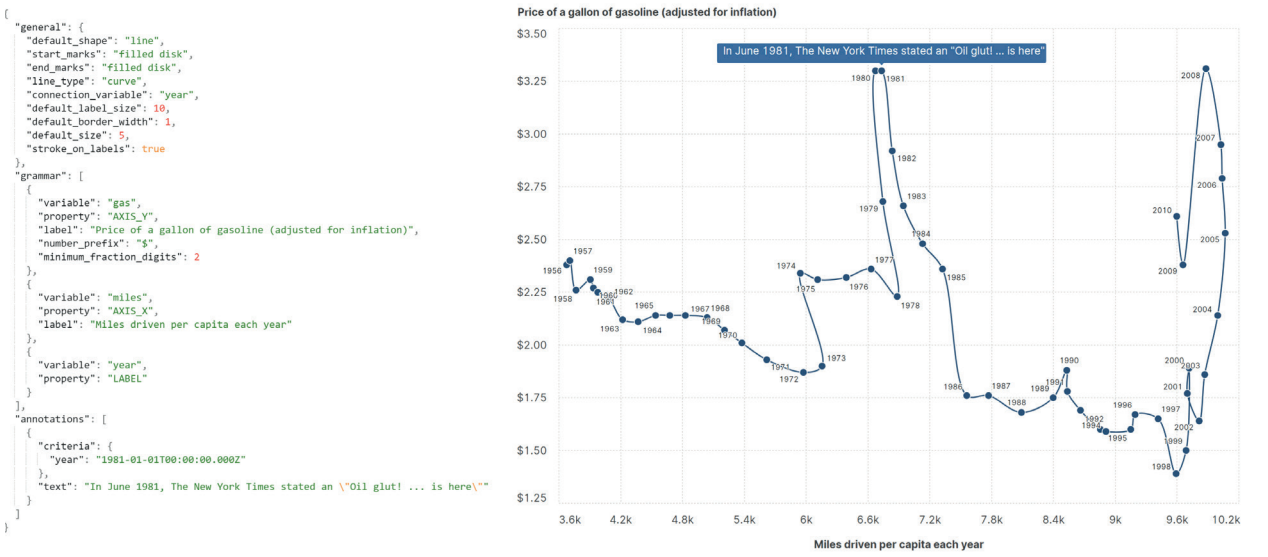


Figura 1. Especificación gramatical de un diagrama de dispersión conectado en *SCImago Graphica*

otro subconjunto describe las relaciones o vínculos entre los nodos y sus atributos. Aunque *SCImago Graphica* utiliza su propio formato basado en CSV para datos de red, la aplicación de escritorio incluye la posibilidad de importar los formatos más comunes: *GML* (*Graph Modelling Language*), *GraphML* y *GEXF* (*Graph Exchange XML Format*).

3.2. Especificación de la gramática

En *SCImago Graphica*, la especificación gramatical de cada gráfico emplea una representación JSON. Cada definición tiene tres secciones: *general*, *grammar* y *annotations* (ver figura 1). No es obligatorio definirlas todas en cada gráfico, solo las necesarias.

En la sección *general* se definen todas las propiedades generales del gráfico, como el tipo de marca (ver tabla 1), los márgenes, la paleta de colores, la forma de los enlaces en el caso de grafos (ver figura 2), entre muchas otras opciones

Tabla 1. Tipos de marcas que se pueden definir en *SCImago Graphica* por la propiedad "default_shape"

	"none"	
●	"filled disk"	
○	"disk"	
	"bar"	
	"line"	La propiedad "line_type" permite los valores "none", "curve" y "orthogonal".
	"area"	La propiedad "line_type" permite los valores "none" y "curve".
	"filled rectangle"	
	"geo"	
	"circle"	La propiedad "circle_chart_type" permite los valores "pie", "rose" y "radar".

La sección *grammar* es donde las variables de datos se asignan a las propiedades de codificación. Este mapeo se define como una matriz de objetos (donde el orden es importante), cada uno de los cuales vincula una propiedad de codificación a una variable. Cada objeto de enlace entre propiedad y variable también puede incluir configuraciones específicas sobre cómo se debe realizar el mapeo entre ellos: función de agregación, escala, ordenación, filtros interactivos, etc.

La sección *grammar* no solo se utiliza para codificar variables de datos a través de canales de codificación visual en el sentido de las variables retinales de **Bertin** (1983), como posición, tamaño, color, opacidad (alfa) y formas. También se utiliza para codificar variables mediante etiquetas o *tooltips* (descripciones emergentes): para definir reglas de filtrado, o para especificar qué variables categóricas dictan los diferentes símbolos a mostrar, o cómo se deben subdividir estos símbolos.

Finalmente, la sección *annotations* se utiliza para definir anotaciones textuales adjuntas a todos los símbolos en el gráfico que coincidan con los criterios especificados.

La especificación de la gramática en *SCImago Graphica* tiene ciertas similitudes con la especificación de gráficos unitarios de *Vega-Lite* (**Satyanarayan** *et al.*, 2017), ya que ambos son simples y legibles. La diferencia más notable es que en *SCImago Graphica* se puede vincular más de una variable de datos al mismo canal de codificación.

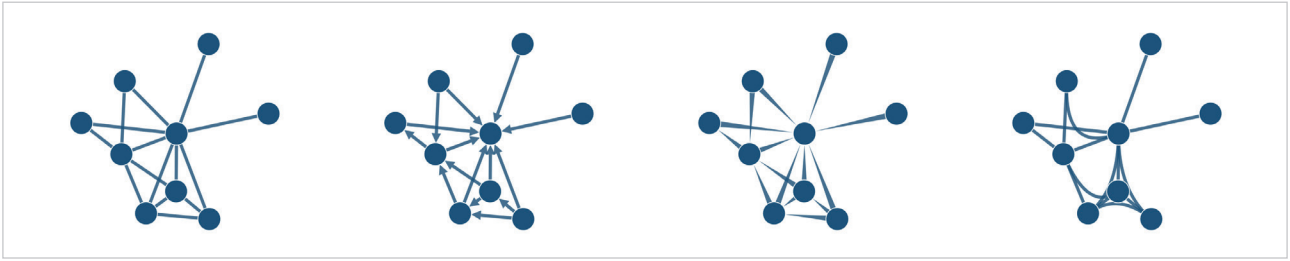


Figura 2. Diferentes tipos de enlaces definidos por la propiedad "edge_form". De izquierda a derecha: 'line', 'arrow', 'tapered' y 'bundling', este último basado en el trabajo de **Wallinger** *et al.* (2022).

3.3. Motor de generación de visualización

El motor de generación es el núcleo de *SCImago Graphica*, el componente que, a partir de una fuente de datos y una especificación de gramática JSON, genera una visualización interactiva y responsiva. Está escrito desde cero en JavaScript, lo que le permite ejecutarse en el navegador, en el servidor e incluso como parte de una aplicación de escritorio independiente (ver sección 3.4). De manera predeterminada, la salida se dibuja usando SVG (*Scalable Vector Graphics*), pero también puede mostrar gráficos interactivos en *Canvas/WebGL* en combinación con la biblioteca *PixiJS*[2].

La primera tarea que maneja el motor de generación es la revisión gramatical. Por ejemplo, comprueba la compatibilidad entre el tipo de variables de datos y los canales de codificación utilizados, o si un canal de codificación se ha vinculado a más variables de las que admite.

El motor también realiza todas las tareas de transformación de datos (agregaciones, *binning* de datos, filtrado...) así como cálculos estadísticos derivados de la especificación gramatical, como *clustering* –mediante un algoritmo basado en **Clauset**, **Newman** y **Moore** (2004)–, análisis de regresión, o cálculo de métricas de red y medidas estadísticas.

Otra tarea clave del motor de generación es el cálculo de la composición visual del gráfico, que depende de múltiples factores. En primer lugar, está condicionado por el tipo de marca elegido; los puntos y las barras, por ejemplo, no se colocan ni organizan visualmente de la misma manera. En segundo lugar, se determina por la combinación de variables que han sido ligadas a propiedades visuales posicionales (eje X, eje Y y pequeños múltiplos). Como puede verse en los ejemplos de la Figura 3, *SCImago Graphica* muestra una gran flexibilidad en la forma en que se pueden realizar estas combinaciones. Pero hay un tercer factor que condiciona la posición de cada símbolo en el gráfico: el algoritmo de composición visual (*layout*) elegido. El algoritmo predeterminado simplemente alinea los símbolos en filas sucesivas, y el algoritmo "*Avoid overlap*" mueve los símbolos iterativamente hasta que ninguno de ellos se superpone con los demás. Pero es cuando se trabaja con datos de red que el abanico de opciones aumenta considerablemente (Figura 4):

- *Force Directed*, basado en **Fruchterman** y **Reingold** (1991);
- *Force Directed+Distances*, una variación del algoritmo *Force Directed* que además utiliza la distancia de ruta más corta entre nodos para calcular las fuerzas de repulsión entre ellos;

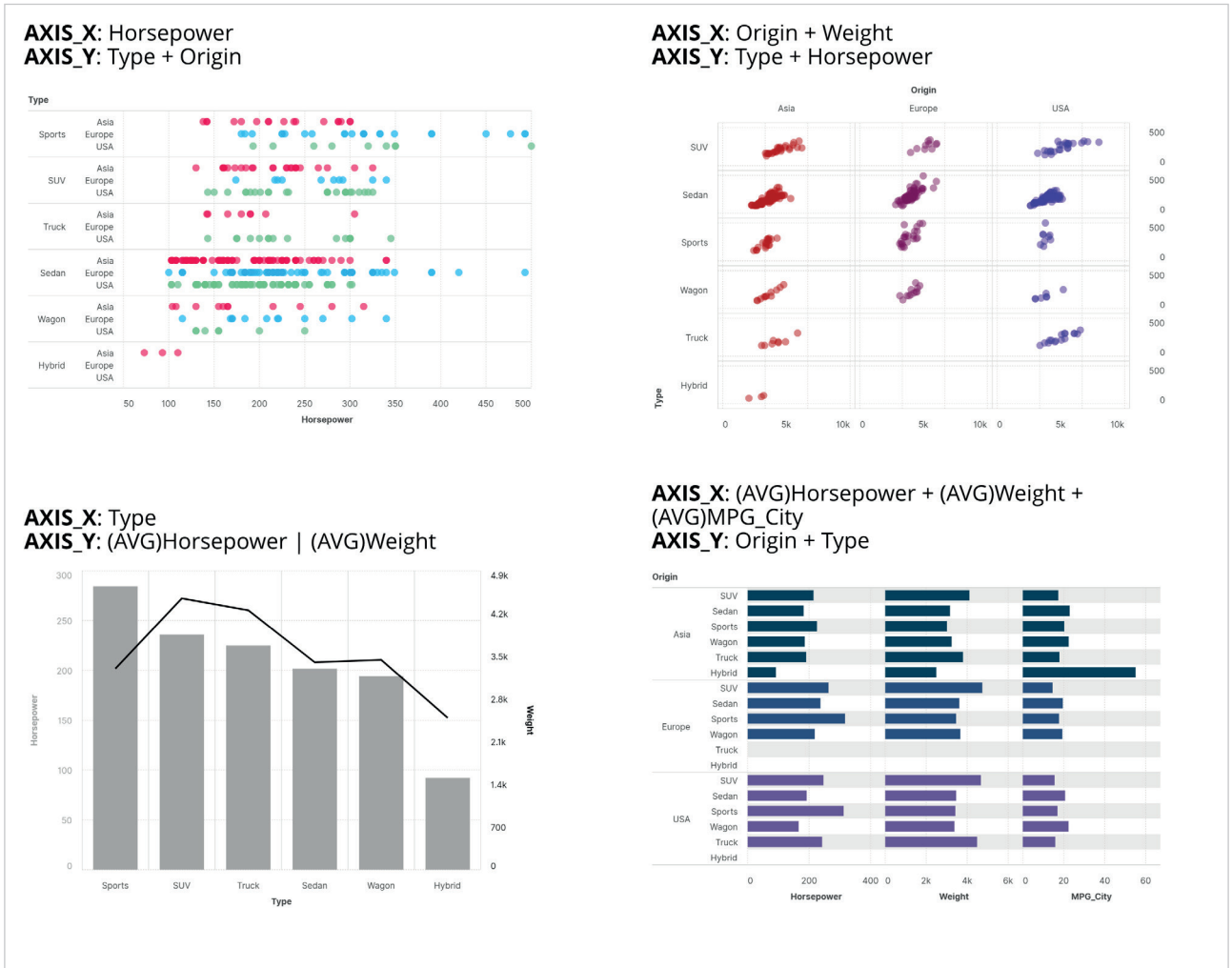


Figura 3. Ejemplos de diferentes composiciones visuales como resultado del uso de diferentes combinaciones de variables y propiedades posicionales

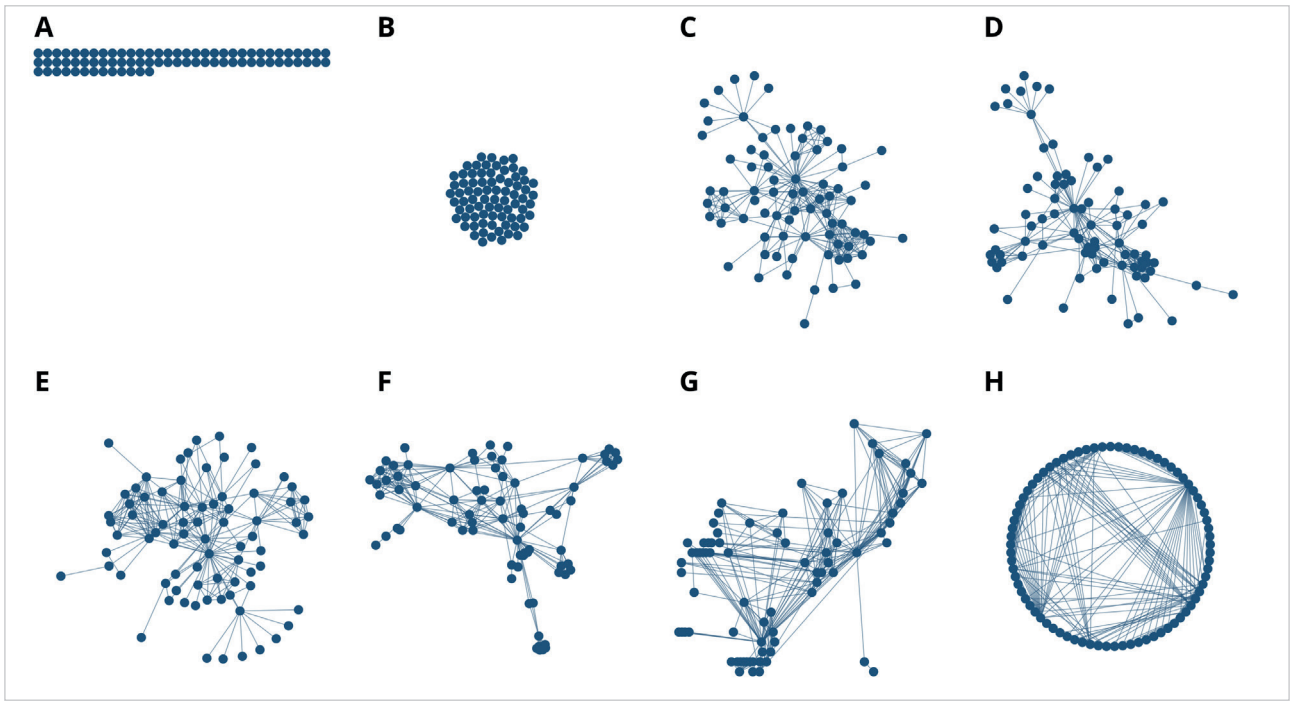


Figura 4. Principales algoritmos de composición visual en *SCImago Graphica:* A) *Default*. B) *Avoid Overlapping*. C) *Force Directed*. D) *Force Directed+Distances*. E) Kamada and Kawai. F) *LinLog*. G) *Dagre Top-Down*. H) *Circular*.

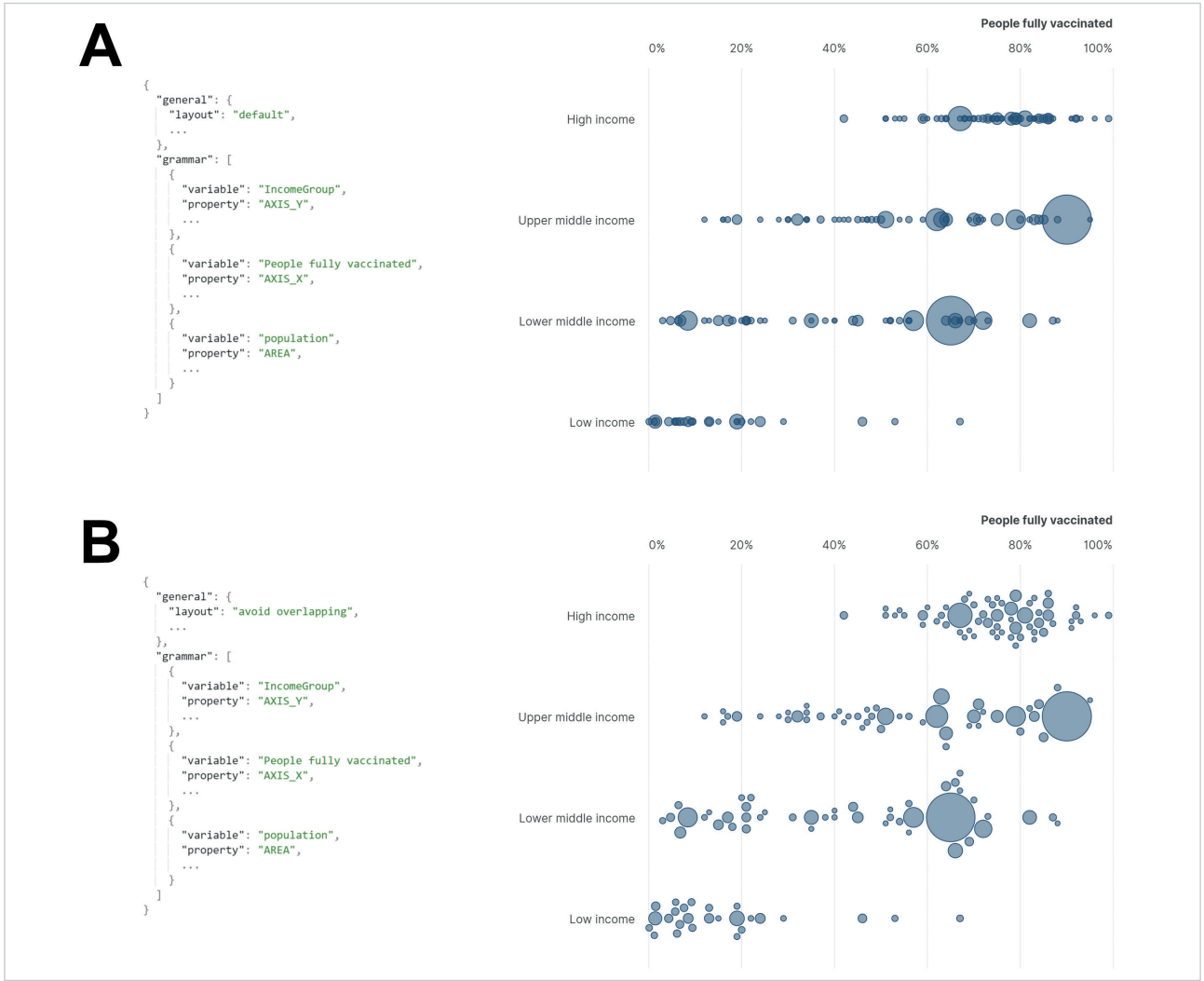


Figura 5. A) Ejemplo de composición visual por propiedades posicionales. B) Ejemplo de composición visual combinando propiedades posicionales con un algoritmo que evita la superposición. La posición Y de cada símbolo viene dada por la variable categórica "IncomeGroup" y por la aplicación del algoritmo, mientras que la posición X viene dada exclusivamente por la variable cuantitativa "People fully vaccinated".

- el algoritmo de Kamada y Kawai para grafos no dirigidos (**Kamada**; **Kawai**, 1989);
- *LinLog*, un algoritmo que usa el modelo de energía de **Noack** (2007), que es particularmente útil para representar relaciones de agrupamiento a través de las posiciones y distancias entre nodos (**Noack**, 2009);
- *Dagre Top-Down* y *Dagre Left-Right*, que son algoritmos de disposición jerárquica[3]; y
- *Circular*, un algoritmo simple que coloca los nodos de un gráfico en un círculo.

Los algoritmos de composición visual se pueden combinar con posiciones fijas, conectando la posición X o Y a un eje cuantitativo, como se muestra en la Figura 5.

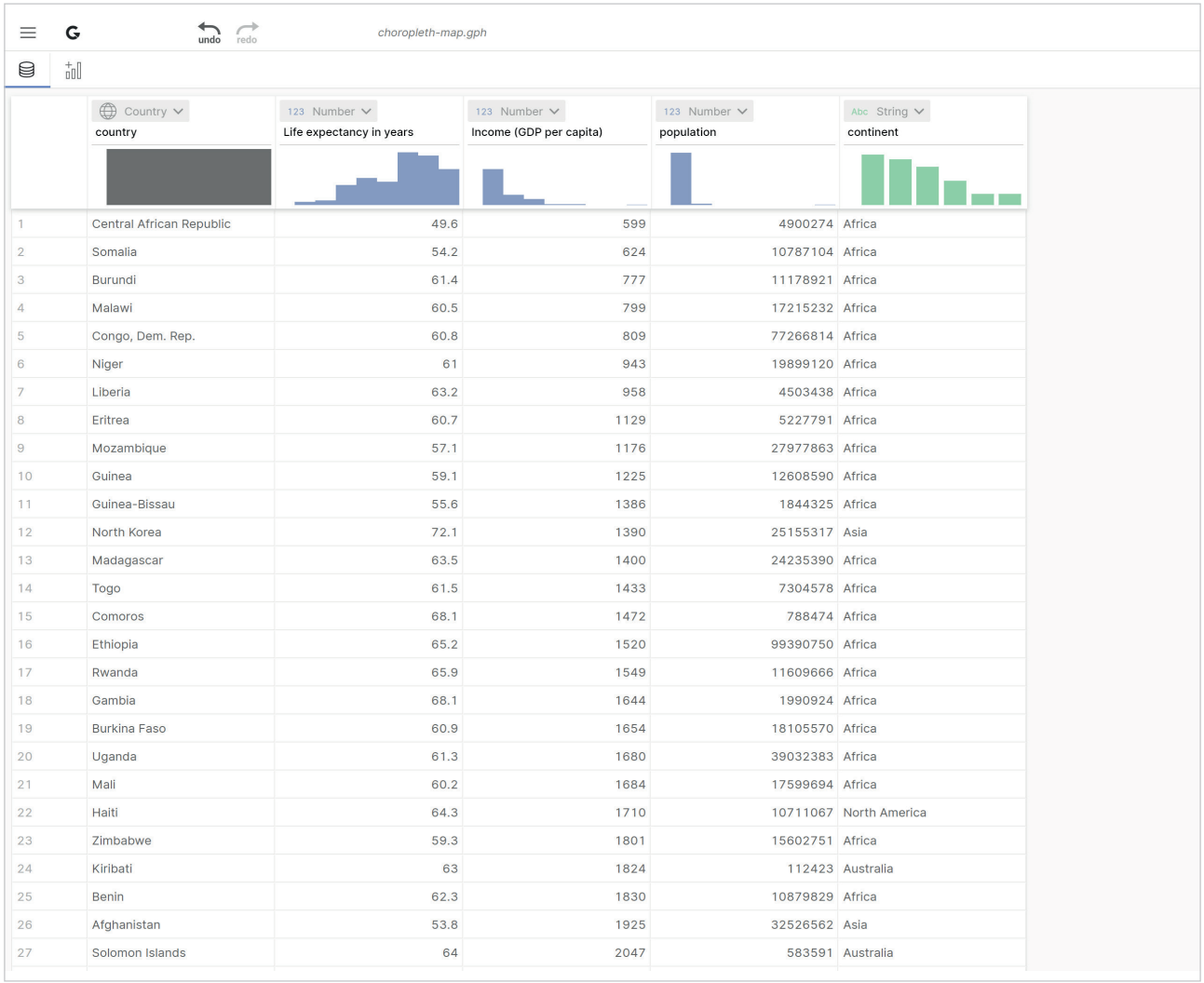
Las visualizaciones generadas, además de ser responsive –se adaptan dinámicamente al ancho y alto disponibles–, pueden ser interactivas. Las interacciones que se pueden especificar en *SCImago Graphica* incluyen *tooltips* (descripciones emergentes), zoom y panorámica, hipervínculos, resaltado al sobrevolar con el puntero del ratón y filtrado interactivo.

3.4. Interfaz de usuario

Para hacer que la exploración y visualización de datos sea simple y sin esfuerzo con *SCImago Graphica*, se creó una aplicación de escritorio independiente. Se utilizó el framework *Electron*[4], que permitió el rápido desarrollo de una aplicación multiplataforma (*Windows*, *MacOS* y *Linux*) usando JavaScript, HTML y CSS.

Como señalan **Grammel**, **Tory** y **Storey** (2010), una de las barreras comunes en el proceso de visualización de datos es la elección de qué variables de datos responden a los objetivos y preguntas que el usuario intenta abordar. Por esta razón, después de cargar el conjunto de datos, se muestra un pequeño gráfico con la distribución de cada variable en el encabezado de cada columna de la tabla de datos (figura 6), lo cual ayuda al usuario a familiarizarse con los datos.

La aplicación permite la creación de múltiples visualizaciones a partir de la misma fuente de datos, organizadas en pestañas (figura 7.1). El modelo interactivo para componer un gráfico es similar a otras herramientas basadas en gramáticas: el usuario vincula las variables de datos (figura 7.2) a los estantes de codificación (figura 7.3) arrastrando y soltando. Al hacer clic en las variables soltadas en cada estante, se pueden modificar propiedades específicas (función de agregación,



	Country country	123 Number Life expectancy in years	123 Number Income (GDP per capita)	123 Number population	Abc String continent
1	Central African Republic	49.6	599	4900274	Africa
2	Somalia	54.2	624	10787104	Africa
3	Burundi	61.4	777	11178921	Africa
4	Malawi	60.5	799	17215232	Africa
5	Congo, Dem. Rep.	60.8	809	77266814	Africa
6	Niger	61	943	19899120	Africa
7	Liberia	63.2	958	4503438	Africa
8	Eritrea	60.7	1129	5227791	Africa
9	Mozambique	57.1	1176	27977863	Africa
10	Guinea	59.1	1225	12608590	Africa
11	Guinea-Bissau	55.6	1386	1844325	Africa
12	North Korea	72.1	1390	25155317	Asia
13	Madagascar	63.5	1400	24235390	Africa
14	Togo	61.5	1433	7304578	Africa
15	Comoros	68.1	1472	788474	Africa
16	Ethiopia	65.2	1520	99390750	Africa
17	Rwanda	65.9	1549	11609666	Africa
18	Gambia	68.1	1644	1990924	Africa
19	Burkina Faso	60.9	1654	18105570	Africa
20	Uganda	61.3	1680	39032383	Africa
21	Mali	60.2	1684	17599694	Africa
22	Haiti	64.3	1710	10711067	North America
23	Zimbabwe	59.3	1801	15602751	Africa
24	Kiribati	63	1824	112423	Australia
25	Benin	62.3	1830	10879829	Africa
26	Afghanistan	53.8	1925	32526562	Asia
27	Solomon Islands	64	2047	583591	Australia

Figura 6. Pantalla inicial de *SCImago Graphica* después de cargar los datos

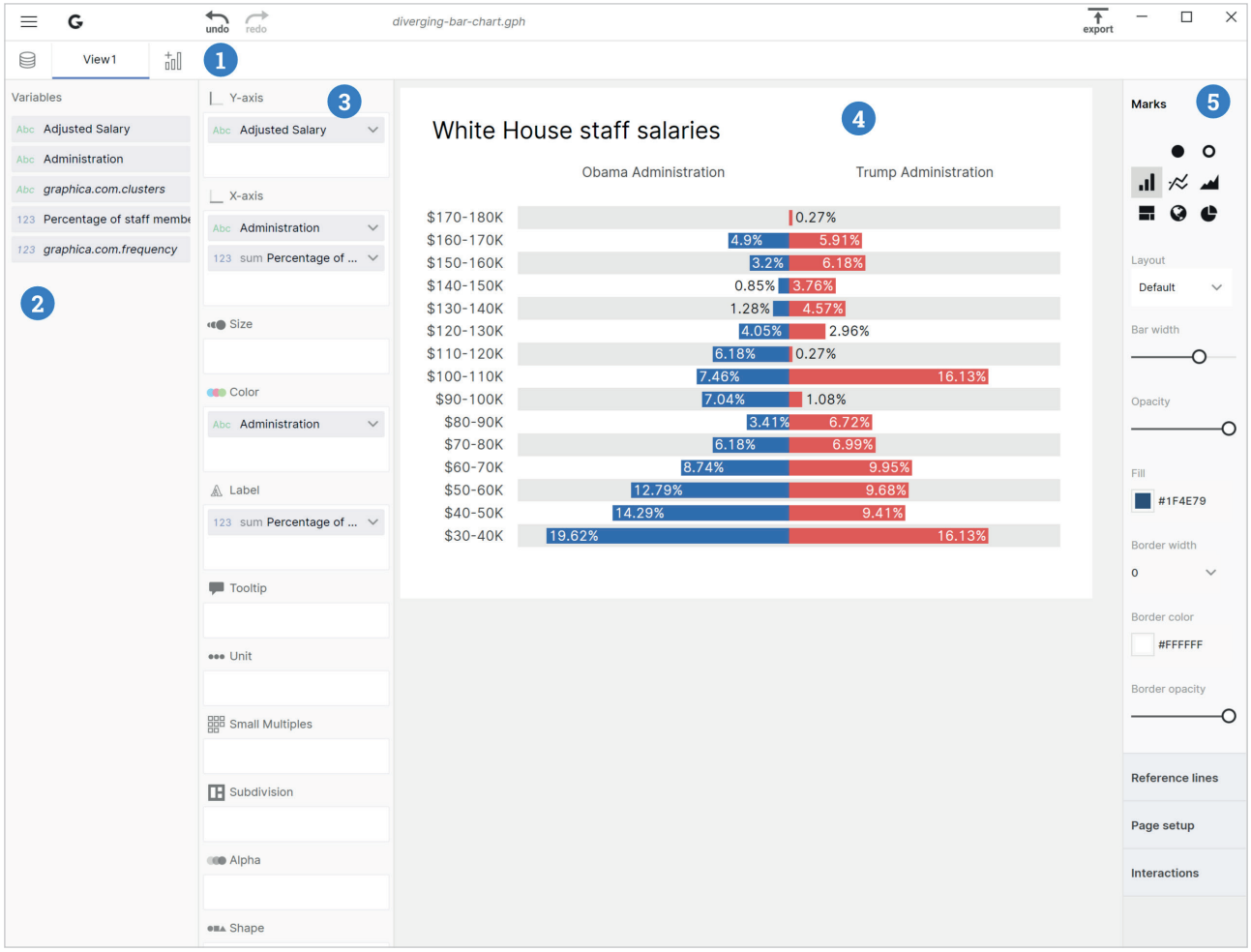


Figura 7. Interfaz *SCImago Graphica* para visualización de datos mediante mapeo visual

formato, visibilidad de la leyenda, etc.). La visualización se muestra en el área central (figura 7.4), en la que el usuario puede agregar anotaciones, editar títulos o cambiar el tamaño del gráfico de forma interactiva. Por último, la barra lateral derecha (figura 7.5) permite definir todas las propiedades generales de la visualización.

La interfaz no solo permite el rápido diseño y la composición de gráficos, sino también la exploración y comprensión de los datos. El usuario puede vincular fácilmente variables a canales de codificación, cambiar variables de un estante de codificación a otro o deshacer y rehacer acciones.

Todos los gráficos creados con la aplicación se pueden exportar a PNG, SVG o al código HTML+JavaScript necesario para su publicación online como una visualización de datos interactiva y responsiva. El código generado, además de las bibliotecas y los estilos necesarios, incluye la especificación gramatical en JSON del gráfico.

4. Evaluación

Tal y como se recoge en los objetivos de este trabajo, el programa que proponemos busca conseguir tanto un alto grado de expresividad como de facilidad de uso, cualidades que se evalúan a continuación.

4.1. Facilidad de uso

Aunque pueda parecer que la manera más objetiva de comparar la facilidad de uso de un software de creación de visualizaciones con los demás es a través de un estudio comparativo, la realidad es que tiene serias limitaciones (**Ren** *et al.*, 2018). Los softwares de creación de visualizaciones son sistemas complejos, que difieren en su filosofía de diseño, modelos de interacción, tecnología subyacente, funciones admitidas o público objetivo. Al comparar los tiempos de finalización de tareas o el número de interacciones, por ejemplo, solo se pueden sacar conclusiones en relación con la tarea específica evaluada, pero nunca sobre la usabilidad general.

Por ello, en este trabajo hemos optado por analizar algunas de las principales diferencias de *SCImago Graphica* respecto al resto de herramientas, a través de una serie de ejemplos. Esta evaluación debe verse como un primer acercamiento para comprender las características fundamentales de la herramienta y su complejidad subyacente.

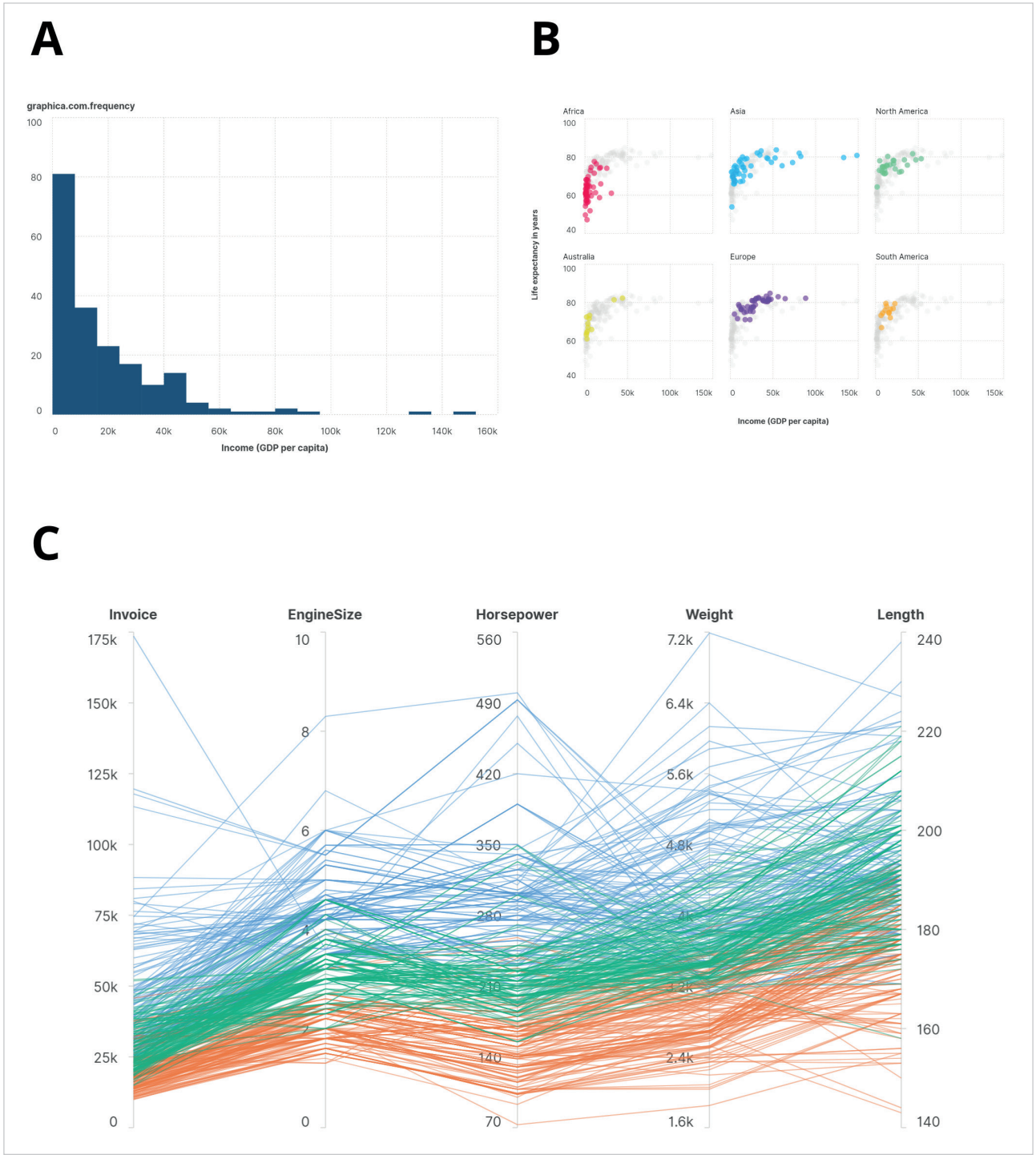


Figura 8. A) Histograma. B) Múltiples pequeños. C) Coordenadas paralelas

A continuación se discuten los ejemplos presentados en la figura 8:

A) Histogramas

Para analizar la distribución de una variable cuantitativa, uno de los gráficos más utilizados son los histogramas. Para crear un histograma en *SCImago Graphica*, un usuario solo tiene que realizar tres acciones: seleccionar 'bar' como tipo de marca, vincular la variable cuantitativa al eje X y vincular la variable "graphica.com.frequency" al eje Y. Esta última variable la calcula la aplicación, y se refiere al conteo o número de ocurrencias. En este ejemplo el usuario no necesita especificar que debe realizarse una agrupación de valores (*binning*), ni cuál debe ser el tamaño del intervalo (*bin*), ya que son operaciones que realiza la propia herramienta.

B) Múltiples pequeños

En visualizaciones de múltiples pequeños, los datos se presentan en forma de pequeños gráficos del mismo tipo que comparten la misma escala. Esta forma de desglosar la visualización de datos facilita comparaciones visuales rápidas entre categorías o períodos de tiempo. A pesar de sus beneficios, las aplicaciones basadas en gramática generalmente requieren que el usuario seleccione una variable categórica para las filas y otra para las columnas, variables que pueden

no existir en el conjunto de datos y, por lo tanto, deben crearse expresamente para lograr el diseño deseado. En *SCImago Graphica*, por el contrario, basta con colocar una variable categórica o de fecha en el estante "Small multiple", y el programa subdividirá el espacio disponible para ubicar los diferentes gráficos pequeños. Esta forma simple de dividir las representaciones de datos es particularmente útil para el análisis exploratorio de datos.

C) Coordenadas paralelas

Los gráficos de coordenadas paralelas son de gran valor para el análisis de datos multivariados debido a su capacidad para revelar correlaciones, similitudes o anomalías en los datos. Por ello, sorprende que en los softwares de visualización de datos más populares no exista una forma sencilla de crearlos. En el caso de *SCImago Graphica*, solo es necesario vincular tantas variables como se desee visualizar a un mismo eje (eje X o eje Y) y seleccionar la línea como tipo de marca.

Si bien los ejemplos analizados no permiten una valoración global de la usabilidad de *SCImago Graphica*, sí dan una idea de cómo la herramienta intenta anticiparse a la intención del usuario y reducir el número de acciones requeridas.

4.2. Expresividad

El rango de las posibles configuraciones de diseño que permite una herramienta basada en gramática no se puede evaluar a través de estudios de usuarios, por lo que **Ren** *et al.* (2018) recomiendan, como una forma de evaluar la expresividad de estas herramientas, ofrecer una galería de ejemplos variados de gráficos creados con ellas. Como señalan los mismos autores, uno de los beneficios de estas galerías es que pueden servir como medio para comparar la expresividad entre diferentes herramientas.

Proporcionamos un catálogo de visualizaciones de datos en actualización permanente, al que se puede acceder online[5]. Los ejemplos de este catálogo están clasificados por función (comparaciones, distribución, correlación, parte-a-todo, evolución, mapa y red), lo que facilita encontrar los que permitan el tipo de análisis deseado. Como se puede observar (figura 9), los ejemplos incluyen desde tipos de gráficos básicos y frecuentes, como los gráficos de barras, hasta otros más complejos, como los cartogramas contiguos. Dado que *SCImago Graphica* está basada en gramática, no en plantillas, los ejemplos del catálogo representan solo una pequeña fracción de todas sus posibilidades combinatorias.

Como muestra nuestro catálogo de visualizaciones, aunque *SCImago Graphica* no alcanza el nivel de expresividad de las gramáticas de programación de bajo nivel, es más expresiva en varios aspectos que otras aplicaciones interactivas basadas en gramática. Su capacidad para visualizar datos de red, o la flexibilidad de sus composiciones visuales, son algunos de sus puntos más destacados.

5. Discusión y futuros trabajos

En este trabajo se ha presentado un nuevo programa de visualización de datos, que se puede utilizar tanto para la comunicación visual de datos como para el análisis exploratorio. Su poder expresivo se ha demostrado a través de un catálogo de

Figura 9. Catálogo de visualización de datos

visualizaciones de datos, mientras que su facilidad de uso se ha inspeccionado a través de varios ejemplos de creación de gráficos. Obviamente, la usabilidad de la aplicación requiere un estudio más profundo. Aunque los ejemplos revisados ilustran la capacidad de *SCImago Graphica* para crear gráficos avanzados con pocas interacciones, serán necesarios tanto los estudios de usuarios como el análisis del uso en escenarios del mundo real para comprender mejor otros atributos de usabilidad.

Si bien el mapeo visual mediante arrastrar y soltar permite crear visualizaciones de datos de manera fácil y eficiente, puede ser una barrera para los usuarios principiantes (**Grammel**; **Tory**; **Storey**, 2010). La solución en estos casos puede ser la integración de una función de recomendación de gráficos. "Show me" (**Mackinlay**; **Hanrahan**; **Stolte**, 2007) sugiere tipos de gráficos compatibles con las variables que el usuario selecciona para representar. *Keshif* (**Yalçın**; **Elmqvist**; **Bederson**, 2018) genera automáticamente tableros orientados a la exploración con el menor esfuerzo posible del usuario. *Voyager* (**Wongsuphasawat** *et al.*, 2015), por otro lado, es un sistema de navegación facetado de gráficos recomendados elegidos en base a medidas perceptuales y estadísticas.

Creemos que una función importante que también deben abordar estos sistemas de recomendación de gráficos basados en gramática es acortar su curva de aprendizaje. Por lo tanto, el futuro sistema de recomendaciones que se integrará en *SCImago Graphica* no solo estará diseñado para simplificar la creación de gráficos para los principiantes, sino también para ayudar a aprender cómo llegar a estas soluciones a través del mapeo visual.

6. Notas

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. Referencias

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, Cristopher** (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

Grammel, 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

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". *CHItaly'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.eg.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 visualization 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.umiacs.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