Classification and Analysis of Techniques and Tools for Data Visualization Teaching

Juan J. Cuadrado-Gallego
Department of Computer Science
Universidad de Alcalá
Madrid, Spain
Department of Computer Science
Concordia University
Montreal, Quebec, Canada
jjcg@uah.es

Yuri Demchenko
Informatics Institute
University of Amsterdam
Amsterdam, The Netherlands
y.demchenko@uva.nl

Miguel A. Losada
Department of Computer Science
University of Alcalá
Madrid, Spain
miguel.losada@edu.uah.es

Olga Ormandjieva
Department of Computer Science
Concordia University
Montreal, Quebec, Canada
ormandj@cse.concordia.ca

2021 IEEE Global Engineering Education Conference (EDUCON) | 978-1-7281-8478-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/EDUCON46332.2021.9453917

***Abstract*—Data Visualization addresses the use of graphics with the purpose to obtain or transmit the knowledge in a easier and faster way, this is it main, and in many cases unique purpose. Since their invention Data graphics has evolved and many techniques has been developed, and in the last decades, with the definition and evolution of the Data Science, Data Visualization has become to be used profusely, in that manner that, by one side, the Data Science Body of Knowledge, DS-BoK, define five knowledge area groups that should be taught when learning Data Science, in all of them Data Visualization is taken a main role for different reasons applying each knowledge area; and by other side all the Data Science development environments, open source or proprietary, include tools for performing Data Visualizations. This paper presents the results of a research carried out with the main objective of improving the teaching of data visualization using two ways: propose a new system to classify the large amount of different graphical techniques for presenting data that can be found in the literature; and analyze using different attributes quite all the most important different tools, open source and private, that are available to develop data graphics mainly form a data visualization teaching point of view.**

***Index Terms*—Data Science, Data Visualizaion, Data Science Education, Data Visualization Education, Engineering Education, Explanatory Graphs, Explorative Graphs, Open Software Visualization Tools, Propietary Software Visualization Tools.**

I. Introduction

Data visualization is a way of graphically representing information and data. Charts help us reason about quantitative information [1].

The Data Science Body of Knowledge, DS-BoK, define the knowledge area groups that should be learned in Data Science, in all of them Data Visualization is taken a main role for different reasons applying each knowledge area, but has an especially important role in Data Analytics. For that reason is mandatory to teach Data Visualization when Data Science is being taught.

Many different charts can be used for operational purposes, such as improving efficiency, monitoring processes, studying the geographic distribution of data, looking for trends and relationships, reviewing the status of projects, developing ideas, writing reports, analyzing data censuses, study sales results among others [2]. And many software, open source and proprietary, tools can be used for deal with. For that reason is very important to identify which of those tools are more efficient for an data visualization educational use.

Before carrying out the comparative analysis, we propose a new classification of the visualization techniques based on three main types: according to their treatmen, according to their shape and according to their movement. With this classification we can better understand the operation of the graphics and generate a clearer vision for the reader. In addition, by having a well-structured classification, the analysis and the comparison between the tools discussed below gives us clearer and more useful information.

Once the classification is proposed, we will carry out a comparative analysis of the main tools for data visualization. On the one hand, we will analyze open software tools, specifically R and Python. On the other hand, we will focus on four of the best positioned proprietary tools in 2020 Gartner Magic Quadrant for Analytics and Business Intelligence [3]. These tools are Power BI, Tableau, Microstrategy and Qlik.

Open source tools are those in which the user can obtain a copy without cost and legally study its source code, modify it and distribute it with other users. A software author uses his or her own copyright to guarantee those rights to all user by affixing to the code a standard, such as the General Public License (GPL), commonly referred to as "copyleft" [4].

When all kinds of programmers work freely on a program's source code, it helps improve collaboration to troubleshoot bugs and enables adaptation to different hardware platforms and needs. In fact, this has happened and open source software is well known today for its high degree of reliability and portability [5].

On the other hand all software licensed under the exclusive legal rights of the copyright holder is what we usually known as proprietary software [6]. Within proprietary software we can distinguish between software distributed on the market in a massive way and that which is not.

Proprietary software in mass distribution reserves almost all copyrights, except a license to ”run” the software on the buyer’s computer. For proprietary software not found in the mass market, any number of agreements can be agreed between the seller and the consumer. The user could, for example, be allowed to view the source code and make changes, but not distribute those changes to others. (Without such permission, consumer changes, if substantial enough, would comprise derivative works, the creation of which is a right reserved to the original author) [7].

II. Data Visualization Techniques Classification

In this paper we present a new classification of Data Graphics techniques, that, in the highest differentiate groups the graphics techniques en three different but not exclusive groups taking in to account three different attributes of the graphics:

- *Use*. The types of data visualizations are defined according to the use or treatment they make of the data, raw or processed data.
- *Shape*. The types of data visualizations are defined according to the different graphic elements use to build them, that can be resume in its shape. This is the attribute used in most of the data graphics classifications that can be found in the literature. Within this classification there are different graph subtypes that in turn have a wide variety among them.
- *Movement*. The types of data visualizations are defined according to their movement, that is, whether or not the display has elements that vary according to time.

One time that the approach to the graphics techniques has been selected, different types of graphics can be found inside them. The new classification is the following:

- Use
  - Exploratory
  - Explanatory
- Shape
  - Axes
  - Circles
  - Maps
  - Rectangles
  - Networks
  - Drawings
- Movement
  - Static
  - Dynamic

In the next subsections we are going to describe the principal features of each type of graphics.

A. Use

- *Exploratory*. The charts are made on raw data. i.e., a simple scatter plot is an exploratory chart since it only shows the raw data (a position value on the axes).
- *Explanatory*. The charts are made on processed data. Following the example, we had a scatter plot and we add a regression line to it, we obtain an explanatory graph, since we have performed an analysis on the data and we have calculated the regression line.

B. Shape

- *Axes*. An axis is a line that has many different functionalities including the reference from which the coordinates are measured, orienting the graph and all the elements found in it, acting as a vehicle through which the marks and scales are displayed and forming a frame around the graph. Therefore, it is known as an axis graph as any graph that uses axes to complement the information that is exposed in it. It is the subtype of graphics according to the form with more variety. Specifically, we have identified up to 44 different axis charts [2]. Some examples are the area chart, box plot, column chart, pareto chart, scatter plot, among others.
- *Circles*. They use circles or circular shapes to represent information. Usually they are used to show parts of a whole, although not exclusively. We have identified up to 13 different circle graphs, among which are radar chart, circular column graph, circle line graph, etc.
- *Maps*. The only way to visually display information regarding your physical (spatial) location is through map graphics. To see how data is distributed geographically there is no substitute for maps. They can be classified into six main categories: statistics, descriptive, flow, topographic (we do not analyze them, since we focus purely on the data, not on merely geographic information), weather maps and special purpose maps. Within these classifications we can find more specific subtypes such as cartograms, connection graphs, etc. Maps with a more specific purpose are identified by their own individual headings. Generally we can find three forms of representation of map graphics such as representation on a spherical or globe surface, representation on a base map or flat surface and finally three-dimensional maps.
- *Rectangles*. Also known as treemaps, they are a type of chart that displays hierarchical data as a set of nested rectangles. Each group is represented by a rectangle whose area is proportional to its value. Various dimensions (groups, subgroups, etc.) can be represented using color schemes or interactive features [8]. This type of chart is used primarily for two purposes: the first to show how the whole is divided, and the second to show how it is hierarchically organized.
- *Networks*. They show the interconnections between a set of entities where each entity is represented by a node or vertex and the connections between the nodes are represented by edges. We have identified up to 16 different

network graphs [9]. Some examples are Arc diagram, Centralized Burst, Flow chart or Organic Rhizome.

- *Drawings*. Drawing graphics or pictorial charts use images, sketches, symbols, icons, etc., instead of standard data graphic elements. They are used to make the document look more interesting, to make the subject easier to understand, or to improve communication where the appearance of the object is relevant. Within this type of graphics we include the well-known word cloud graphic, which shows with a larger size the words that are most repeated in a certain text.

C. Movement

- *Static*. We know by static graphics any graphic that does not have any interactive or dynamic element, that is, all the information is shown in a still image that does not change over time.
- *Dynamic*. The information they show varies with time. Within dynamic graphics we can distinguish between interactive graphics and animated graphics. On the one hand, interactive charts are those in which we have some control over chart elements. That interaction can be information that is displayed when we slide the cursor over certain elements of the graph, a graph with a timeline that we can move on, etc. On the other hand, animated graphics are those that by themselves vary the information they show as a function of time, that is, they vary continuously and we have no control over them. Similar to what happens with the three great typologies, a dynamic graph can be interactive and animated at the same time.

One of the characteristics of this differentiation is that, how was said in the definition the three groups are not exclusive, that means that a single data graphic will belong to the three groups, i.e. a column graph is, from the use point of view, an exploratory graph; a axis graph, from the shape point of view; and it can be static or dynamic.

III. Data Visualization Tools Analysis

In the following an analysis of the data visualization tools is presented. It have been done for the two types of software:

- *Open Source*. The analysis of the open source software has been done for R and Python, that are the main environment for Data Science. and have been carried out for their main visualization tools that are, respectively, ggplot2 and matplotlib, but when the broad of those tools dont́ reach specific techniques other specific packages are analyzed to close that gap.
- *Proprietary*. The analysis of the proprietary software have been done for the three of the four tools in the top right quadrant, leaders, in the 2020 Gartner BI Magic Quadrant, which are, Microsoft, Tableau, and Qlik; and one in the top left, challengers, Microstrategy.

And a set of attributes has been defined to analyze both visualization tools, open source and proprietary, from an educational point of view, that is, for their use for teaching the Data Visualization techniques of Data Science, classified in the previous section, taking that is to account, four attributes have been defined: broad, depth, slope and documentation. The meaning of each one of them is:

- *Broad*. Refers to the number of different types of data visualization graphic techniques according to their shape that the software is able to deal with. This attribute measure the number of techniques that can be taught using the tool.
- *Depth*. Refers to the number of characteristics of a specific data visualization graphic technique that the software is able to deal with. This attribute measures the level detail with which an specif technique can be taught, and all the aspect of the technique that can be used.
- *Slope*. Refers to the slope of the learning time that is needed to manage the software fluently. If the software it is easy to learn and apply the slope will be steep and if is difficult the curve will be flat. This attribute measure the difficulty of learning of the software by the students.
- *Documentation*. Refers to the documentation available to know the functionality and use of the software. This attribute is related with the slope but is not necessarily correlated because a lot of documentation available not imply that the software was easy to learn and use. This attribute measure the amount of information about the software that can be found and used for teaching and learning it because that impact directly in its educational use.

The results of the application of the defined attributes to the selected tools is presented in the following subsections for each tool, and when the broad of any tool is completed with the use of other tools they are presented inside the subsection of the tool.

A. ggplot2 (open source)

Graphics grammar-based R package that allows you to implement graphics by combining certain components using that grammar [10].

- *Broad*. The broad of ggplot2 is a range of 48.8%-67.85%. That means that for the total number of the 84 different types of graphics from the point of view of its shape stated in section II, ggplot2 is able to manage 41, which is near to be a half. This amount establish the lower value of the range for the broad, the upper value is stablish because ggplot2 package has extensions that can be installed and with them the number of graphics supported increases to 57. If the analysis is performed taking into account the different types of graphics, very different results are found for each type, that is, for axis graphics 28 of the 44 types are covered; for circle graphics 8 of the 13 types are covered. However, for network graphics, map graphics, drawing graphics and rectangle graphics, their support is low. To solve this problem other R packages can be used, those packages have a low broad if all the kinds of techniques are consider but they have a very high broad for the specific kind of graphics for which they have been developed. Some of them are: for network graphics ggraph [11], igraph [12], ggnetwork[13] and more [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]; for map graphics we have plotly [24], ggmap [25], leaflet [26]; for treemaps the treemapify [27] package adds all the necessary functionality; for drawing graphics although there are many variations with ggimage [28], waffle [29] or wordcloud [30] we cover some of its variants. If the movement is consider ggplot2 does not incorporate any type of movement to its graphics, in this aspect a package that can add this functionality is plotly.

- *Depth*. The depth of ggplot2 is very high.
  The high depth of ggplot2 is based on its layered structure that allows you to modify quite all the attributes invloved in the different graphics treated. i.e. only the geom_rect layer has 9 attributes to change the characteristics of rectangular shapes such as borders, line thickness, opacity, fill color, etc.
- *Slope*. The slope of the learning process is very steep.
  As we mentioned earlier, ggplot2 has a layered structure that is based on graph grammar. Graphics are built step by step by adding new elements or layers, resulting in great flexibility for personalizing them. This is a simple and clear structure to understand and can be extrapolated to any graph that supports ggplot, however different it may be from another. This allows that in general terms it is a very easy package to understand and with a learning curve that reaches the maximum very quickly.
- *Documentation*. ggplot2 has a very high degree of documentation.
  The official documentation of the package can be found at its own CRAN address and it is composed by a document of 284 pages that explains in detail each function and its corresponding parameters. In addition, being part of an open source tool package, it is very easy to find all kinds of forums and web pages created by the community.

a.1) Matplotlib (open source)

Python library to implement 2D visualizations with a structure based on multiple objects. [31]

- *Broad*. The broad of ggplot2 is a range of 50%-55.95%.
  That means that for the total number of the 84 different types of charts from the point of view of their shape set out in section II, matplotlib can manage 42, which is exactly half. This amount establish the lower value of the range for the broad, the upper value is stablish because matplotlib package has extensions that can be installed and with them the number of graphics supported increases to 47. If the analysis is focused on the different types of charts, very different results are obtained for each type, that is, for axis charts 28 of the 44 types are covered; For pie charts, 9 of the 13 types are covered. However, for network graphics, map graphics, drawing graphics, and rectangle graphics, their compatibility is low. To solve this problem other Python packages can be used, those packages have a low broad if all the kinds of techniques are consider but they have a very high broad for the specific kind of graphics for which they have been developed. Some of them are: for network graphics igraph [32] or networkx [33]; for map graphics we have plotly [34], geopandas [35] or geoplot [36]; for treemaps squarify [37] or pygal [38]; for drawing graphics waffle [29] or wordcloud [39] cover some of its variants. If the movement is consider matplotlib does not incorporate any type of movement to its graphics, in this aspect a package that can add this functionality is plotly.
- *Depth*. The depth of matplotlib is very high.
  The high depth of matplotlib is based on its multiple object structure that allows you to modify quite all the attributes invloved in the different graphics treated. i.e. only the matplotlib.pyplot.barh layer has 5 attributes to change the characteristics of columns shape as y coordinates, width, height, left (x coordinates of the left sides of the bar) and align but it also has 10 more attributes to change colors, labels, etc.
- *Slope*. The slope of the learning process is slightly steep.
  As we mentioned earlier, matplotlib has a multiple object structure. Graphics are built step by step by adding new objects, resulting in great flexibility for personalizing them. Although it is a simple structure it is not so simple to extrapolate to any other graph. This allows it to be, in general terms, not a very complicated package to enter, but in some cases it is somewhat confusing. Still if learning is fast.
- *Documentation*. matplotlib has a very high degree of documentation.
  The official documentation of the package can be found at its own web page [31], where among many other things it has an index that explains in detail each function and its corresponding parameters. In addition, being part of an open source tool library, it is very easy to find all kinds of forums, web pages and audiovisual content.

a.2) Microsoft Power BI (proprietary)

Microsoft tool launched in 2014 focused on Business Intelligence that allows to model, analyze and visualize data.

- *Broad*. The broad of POWER BI is 45.23%.
  That means that for the total number of the 84 different types of graphics from the point of view of its shape stated in section II, Tableau is able to manage 38, which is near to be a half. If the analysis is focused on the different types, very disparate results are obtained, highlighting the graphs of axes. For axis charts 22 of the 44 types are covered; rectangle graphics are supported; for circle graphics 6 of 13 types are covered; for map graphics covers the most classic statistical map types; for drawing graphics covers word cloud and waffle graphics. However, for network graphics compatibility is low. Finally, Power BI incorporates a high degree of motion into your graphics, provides support for R and Python, and allows to import new visualizations.
- *Depth*. The depth of Power BI is high.
  For each visualization, it generally has more than ten dropdowns to modify the appearance of the same. These options range from the legend, axes, labels, etc., to shading, background, shape, among others.
- *Slope*. The slope of the learning process is slightly steep.
  It is not as intuitive a tool as any of its competitors, especially at the beginning; i.e. it does not recommend

the appropriate parameters for each graph. However, once its operation and structure are understood, it is a very comfortable tool to use. One of its main advantages is the format field, since, from a single panel we can modify any aspect of the visualization.

- *Documentation*. Power BI has a high degree of documentation.
  On its official page it has its extensive documentation in the form of guides with excellent explanations and examples. In addition, it has a community forum, which works especially well with a good active user base.

a.3) Tableau (proprietary)

Comprehensive analytics platform launched by Tableau Software in May 2013 focused on business intelligence and visual analytics.

- *Broad*. The broad of Tableau is 35.71%.
  That means that for the total number of the 84 different types of graphics from the point of view of its shape stated in section II, Tableau is able to manage 30, which is close to a third. If the analysis is focused on the different types, very disparate results are obtained, highlighting the graphs of axes and maps above the others. For axis charts 22 of the 44 types are covered; rectangle graphics are supported; for map graphics it supports statistical, descriptive, flow, weather and some specific-purpose maps. However, for network graphics, drawing graphics, circles graphics their compatibility is low. Lastly, Tableau incorporates a high degree of motion into its graphics and provides support for R and Python.
- *Depth*. The depth of Tableau is Medium.
  In most visualizations we can modify aspects such as color, size, labels, although not in a deep way. It should be noted that their level of customization in terms of map graphics is higher.
- *Slope*. The slope of the learning process is steep.
  One of Tableau's undoubtedly biggest advantages is its intuitiveness. Its clean interface, the automatic identification and classification of parameters in dimensions and measures, the graphics panel that indicates what type of data to build the visualization make it a very easy tool to understand.
- *Documentation*. Tableau has a medium degree of documentation.
  On its official page it has guides and courses to learn how to use the tool. They are useful, but they can significantly slow down learning time. Being a proprietary tool, it is more difficult to find portals where specific questions are solved and there is not as much audiovisual content as for open source tools, although it has a community forum. In summary, its documentation is good but it does not have the advantages offered by a large community such as open source.

a.4) Qlik Sense (proprietary)

Complete data analysis platform launched in 2014 by Qlik.

- *Broad*. The broad of Microstrategy is 30.95%.
  That means that for the total number of the 84 different types of graphics from the point of view of its shape stated in section II, Qlik is able to manage 26, which is close to a third. If the analysis focuses on the different types, poor results very similar to Microstrategy are obtained.. For axis charts 16 of the 44 types are covered; rectangle graphics are supported; for circle graphics 3 of 13 types are covered; for map graphics covers the most classic statistical map types (choroplectics, bubbles, connections and density); for drawing graphics only cover wordcloud graph. However, for network graphics compatibility is low.
- *Depth*. The depth of Qlik is medium.
  It allows modifying the most general aspects of the visualizations at a medium level.
- *Slope*. The slope of the learning process is very steep.
  It is a very simple tool to use. To create a visualization we only have to select a type of graph and assign it the dimensions or measures of which it needs (similar to Tableau). In addition, it has a section called " Knowledge " in which, based on our data, it creates the most common views for this type of information. It is therefore one of the easiest tools to use.
- *Documentation*. Qlik has a high degree of documentation.
  Its documentation is very well structured, extensive, clean and the information is easy to find. Like the documentation, its community forum is well structured and it is easy to find a post on any topic.

a.5) Microstrategy Desktop (proprietary)

MicroStrategy Desktop is a free self-service analytics-based data discovery tool released by Microstrategy.

- *Broad*. The broad of Microstrategy is 30.95%.
  That means that for the total number of the 84 different types of graphics from the point of view of its shape stated in section II, Microstrategy is able to manage 26, which is close to a third.
  If the analysis focuses on the different types, generally good results are not obtained. For axis charts 16 of the 44 types are covered; rectangle graphics are supported; for circle graphics 3 of 13 types are covered; for map graphics covers the most classic statistical map types; for drawing graphics covers word cloud and waffle graphics. However, for network graphics compatibility is low. Finally, Microstrategy incorporates a high degree of motion into your graphics, provides support for R and Python, and allows you to import new visualizations.
- *Depth*. The depth of Microstategy is medium.
  It allows modifying the most general aspects of the visualizations at a medium level.
- *Slope*. The slope of the learning process is steep.
  Its interface is practically the same as that of Tableau. Identify and classify data into attributes and measures (in Tableau he called them dimensions and measures). It also has a dropdown menu that indicates what type of data each visualization needs to create. In general, it is a very clean and intuitive interface that allows to learn how to use the tool in a very short period of time.

- *Documentation*. Microstrategy has a medium degree of documentation
  Its documentation is too dense and its structure can be confusing. It has a community forum, but it does not have audiovisual guides and information can be difficult to find on certain occasions. In short, although it has extensive documentation pages, this is not as clear as it should be.

TABLE I
OPEN SOFTWARE SOURCE AND PROPRIETARY SOFTWARE FOR VISUALIZATION COMPARISON

Package	*Broad*	*Depth*	*Slope*	*Documentation*
Open Source Software				
ggplot2	67.85%	VH[a]	VH	VH
Matplotlib	55.95%	VH	M	VH
Proprietary Software				
Microsoft Power BI	45.23%	H	M	H
Tableau	35.71%	M	H	M
Qlik Sense	30.95%	M	VH	H
Microstrategy Desktop	30.95%	M	H	M

a. VH: Very High; H: High; M: Medium; L: Low; VL: Very Low.

Table [I] summarizes the information presented on the analysis of the visualization tools.

IV. CONCLUSIONS

This paper presents the results of a research carried out with the purpose to improve the teaching of a subject of Data Science, Data Visualization, which is currently of great interest both in the scientific and industrial communities. Data Visualization involves, fundamentally, the knowledge of the use of graphic techniques for visualizing data, and the knowledge of the use of software tools that allow to do that.

Starting from the description of a new classification of the data visualization techniques developed in this research, that allow to a more easily understanding of the analogies and differences between the different data visualization techniques are their best use depending of which kind of data want be visualized; this paper present an analysis of the tools, both open source and proprietary, that can be used to teach the data visualization techniques. To do that analysis four parameters have been defined: Broad, Deep, Slope and Documentation.

In the previous section a comparative analysis about tools have presented but some other conclusions can be added to it about the advantages and disadvantages that can be found in open source and proprietary software, free software has a great advantage over proprietary software and that is its great community that continually uploads new packages and libraries to the network to improve the functionality of these programs, and which can also be of great support when looking for solutions to certain problems and doing that improve the learning curve. Another advantage of free software is that has a greater variety of visualizations of any kind and a greater capacity for customization of these. Proprietary software also has advantages over free software such as its speed when creating any implementation, in general, only couple of fields must be selected to create any visualization that make his learning curve very fast.

In short, both options are more than recommended for using in education of Data Visualization, also because the industry demands Data Science professionals with Knowledge in open source and proprietary visualization tools. Nevertheless one additional aspect can be considered and it is which kind of tools, open source or proprietary could be taught before, and in this case perhaps the selection will be open source because they are more easy to obtain for the students because they are free and also allow them more easily to the student to focus in the learning on the technique fundamentals because they can deal more in deep with all the features of the technique.

Future works include a more extensive comparison including a large number of tools, fundamentally new open source packages of R and libraries of Python. Also experimental data of their application of the defined attributes over different sets of students will be obtained and analyzed.

V. ACKNOWLEDGMENTS

The research leading to these results has received funding from the Horizon2020 projects FAIRsFAIR (grant number 831558), MATES (grant number 591889) and EDISON (grant number 675419). Also it has been supported by the Art. 83 contract 2020/00224/001 signed between the University of Alcalá and Ultra Tendency for research in Data Science and Big Data.

REFERENCES

[1] Edward R. Tufte. *The Visual Display of Quantitative Information*. Chesire, Connecticut: Graphics Press USA, 2001. ISBN: 9780961392147.

[2] Robert L. Harris. *Information Graphis A Comprensive Illustrated Reference*. Atlanta, Georgia: Oxford University Press, 1999. ISBN: 9780195135329.

[3] *2020 Gartner Magic Quadrant for Analytics and Business Intelligence Platforms*. https://info.microsoft.com/ww-landing-2020-gartner-magic-quadrant-for-analytics-and-business-intelligence.html?LCID=EN-US. Accessed on 2020-05-21.

[4] Eric Von Hippel. “Learning from open-source software”. In: *MIT Sloan management review* 42.4 (2001), pp. 82–86.

[5] Andrea Bonaccorsi and Cristina Rossi. “Why open source software can succeed”. In: *Research policy* 32.7 (2003), pp. 1243–1258.

[6] N Pankaja and PK Mukund Raj. “Proprietary software versus open source software for education”. In: *American Journal of Engineering Research* 2.7 (2013), pp. 124–130.

[7] Jonathan Zittrain. “Normative principles for evaluating free and proprietary software”. In: *The University of Chicago Law Review* (2004), pp. 265–287.

[8] Yan Holtz. *Treemap*. https://www.data-to-viz.com/graph/treemap.html. Accessed on 2020-02-22.

[9] Cristopher Calva Cárdenas. “Estado del arte en visualización de redes con R”. B.S. Thesis. Alcalá de Henares, Spain: Universidad de Alcalá, 2019.

[10] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org.

[11] Thomas Lin Pedersen. *ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. R package version 2.0.3. 2020. URL: https://CRAN.R-project.org/package=ggraph.

[12] Gabor Csardi and Tamas Nepusz. "The igraph software package for complex network research". In: *InterJournal* Complex Systems (2006), p. 1695. URL: http://igraph.org.

[13] J.J. Allaire, Christopher Gandrud, Kenton Russell, and CJ Yetman. *networkD3: D3 JavaScript Network Graphs from R*. R package version 0.4. 2017. URL: https://CRAN.R-project.org/package=networkD3.

[14] Alex T Kalinka and Pavel Tomancak. "linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type". In: *Bioinformatics* 27 (2011), pp. 2011–2012. DOI: 10.1093/bioinformatics/btr311.

[15] B. W. Lewis. *threejs: Interactive 3D Scatter Plots, Networks and Globes*. R package version 0.3.3. 2020. URL: https://CRAN.R-project.org/package=threejs.

[16] Carter T. Butts, Ayn Leslie-Cook, Pavel N. Krivitsky, and Skye Bender-deMoll. *networkDynamic: Dynamic Extensions for Network Objects*. R package version 0.10.1. 2020. URL: https://CRAN.R-project.org/package=networkDynamic.

[17] Sacha Epskamp, Angélique O. J. Cramer, Lourens J. Waldorp, Verena D. Schmittmann, and Denny Borsboom. "qgraph: Network Visualizations of Relationships in Psychometric Data". In: *Journal of Statistical Software* 48.4 (2012), pp. 1–18. URL: http://www.jstatsoft.org/v48/i04/.

[18] Zuguang Gu, Lei Gu, Roland Eils, Matthias Schlesner, and Benedikt Brors. "circlize implements and enhances circular visualization in R". In: *Bioinformatics* 30 (19 2014), pp. 2811–2812.

[19] Bryan A. Hanson. *HiveR: 2D and 3D Hive Plots for R*. R package version 0.3.63. 2020.

[20] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer, 2002. URL: http://www.stats.ox.ac.uk/pub/MASS4.

[21] Gaston Sanchez. *arcdiagram*. 2014. URL: https://github.com/gastonstat/arcdiagram#readme.

[22] Carter T. Butts. "network: a Package for Managing Relational Data in R." In: *Journal of Statistical Software* 24.2 (2008). URL: http://www.jstatsoft.org/v24/i02/paper.

[23] François Briatte. *ggnetwork: Geometries to Plot Networks with 'ggplot2'*. R package version 0.5.8. 2020. URL: https://CRAN.R-project.org/package=ggnetwork.

[24] Carson Sievert. *plotly for R*. 2018. URL: https://plotly-r.com.

[25] David Kahle and Hadley Wickham. "ggmap: Spatial Visualization with ggplot2". In: *The R Journal* 5.1 (2013), pp. 144–161. URL: https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf.

[26] Joe Cheng, Bhaskar Karambelkar, and Yihui Xie. *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 2.0.3. 2019. URL: https://CRAN.R-project.org/package=leaflet.

[27] David Wilkins. *treemapify: Draw Treemaps in 'ggplot2'*. R package version 2.5.4. 2020. URL: https://CRAN.R-project.org/package=treemapify.

[28] Guangchuang Yu. *ggplot2: Use image in 'ggplot2'*. 2020. URL: https://cran.r-project.org/web/packages/ggimage/index.html.

[29] Bob Rudis and Dave Gandy. *waffle: Create Waffle Chart Visualizations in R*. R package version 0.7.0. 2017. URL: https://CRAN.R-project.org/package=waffle.

[30] Ian Fellows. *wordcloud: Word Clouds*. R package version 2.6. 2018. URL: https://CRAN.R-project.org/package=wordcloud.

[31] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[32] Patrick Fuller. *igraph*. 2014. URL: https://github.com/patrickfuller/jgraph.

[33] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[34] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[35] K Jordahl. *GeoPandas: Python tools for geographic data*. 2014.

[36] Aleksey Bilogur. *geoplot*. 2017. URL: https://github.com/ResidentMario/geoplot.

[37] Uri Laserson. *squarif*. 2013. URL: https://github.com/laserson/squarify.

[38] Florian Mouner. *pygal*. 2012. URL: http://www.pygal.org/en/stable/index.html.

[39] Andreas Mueller. *Wordcloud*. 2015. URL: https://github.com/amueller/word_cloud.