Producing Accessible Statistics Diagrams in R

Fitzpatrick, Donal; Godfrey, A. Jonathan R.; Sorge, Volker

[Link to publication on Research at Birmingham portal](#)

Producing Accessible Statistics Diagrams in R

Donal Fitzpatrick
Dublin City University, Ireland
donal.fitzpatrick@dcu.ie

A. Jonathan R. Godfrey
Massey University, New Zealand
A.J.Godfrey@massey.ac.nz

Volker Sorge
University of Birmingham, UK
V.Sorge@cs.bham.ac.uk

ABSTRACT

Blind people are at risk of being left behind in the information age if efforts are not made to improve the access to information that is not traditionally conveyed in text, whether that text be accessed in braille, audio, or a computer's screen reading software. Most graphics summarise a scene or some aspect of data that the author hopes will inform their audience; good statistical graphics are commonly used to great effect for the sighted world, but are practically useless to a blind audience. Our work aims to provide an accessible way for blind users to easily, efficiently, and most importantly accurately, explore and query the data contained in diagrams such as bar charts, box plots, time series, and many more. We employ the statistical software environment R to compute rich semantics for these diagrams and make them web accessible by supporting screen reading and interactive exploration.

CCS Concepts

Human-centered computing, Accessibility technologies

Keywords

STEM Accessibility, accessible diagrams, Statistics

1. INTRODUCTION

It has been argued in [4] that blind people need to become producers of graphics, not just consumers of the information in those graphics if they are to have the same opportunities and outcomes in education and employment as their sighted peers take for granted. Providing a blind person with the confidence that they can independently produce a document for the sighted world is a gold standard for accessibility of the tools used to produce that document.

Since the emergence of products such as Nomad [7] in the late 80s, ongoing efforts have attempted to provide accessible solutions to enable blind people to explore a range of different types of diagram. Such solutions, however, all rely on utilising extra hardware to provide this increased level

of access [2, 5, 1]. The work described here is based on the principle that the blind user should not need to make use of any additional hardware in order to explore the diagram. Another core precept is that the blind statistician should be able to obtain the same information from the graph as their sighted peers.

As simple textual summarisation is generally not enough to convey the meaning of a graphic, we aim to support statisticians in engaging with diagrams that goes beyond simply viewing the arrangement of data. the process of producing accessible diagrams begins inside R itself by augmenting the statistical model and semantically enriching the generated graphics to make them accessible. This results in diagrams whose individual components are more conducive to interactive exploration using a hierarchical navigation model that supports screen reading, together with synchronised highlighting and magnification. We first present an overview of some of the statistical diagrams we are currently working with (Sec. 2, give an overview of the process that allows us to make them web accessible in Sec. 3, before elaborating on our generic hierarchical navigation model in Sec. 4.

2. STATISTICAL GRAPHICS

Broadly speaking, statistical graphics fall into two categories; information is depicted using either spaces or points, but in both cases, it is the comparison between areas or points that should convey the salient features within the data being represented to the intended audience in a way that is possible but less efficient or effective in prose. We have initially concentrated on working with a number of particularly important diagrams, that can roughly be divided into two groups.

2.1 Discrete Data Graphs

Bar charts are a common form used to represent the frequency of data points in different categories. Particular types of bar charts include histograms, where data is clustered into discrete intervals that are given on the x axis, while the frequency of points in each interval is given along the y axis. A histogram uses area to convey information to represent frequencies as an effective visual medium, but the information itself does not need to be represented this way. Tabulating might be seen as a reasonable alternative to a histogram, but this would lack some manipulations that are possible using the graphic. For example, a sighted user would scan across the bars and notice the peaks and any gaps or outliers. Determining these features from a tabulated set of frequencies is possible, but is no longer standard
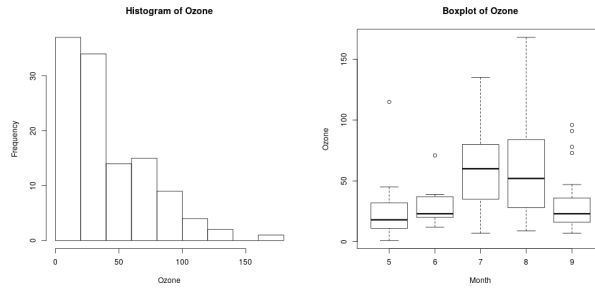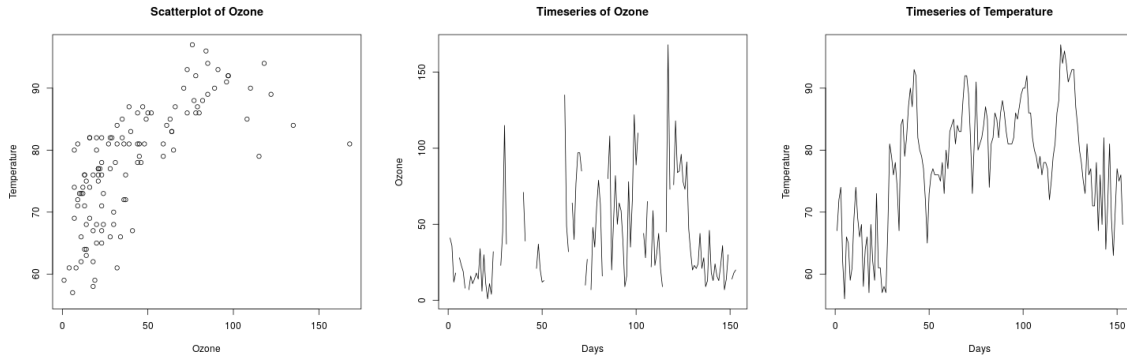
Figure 1: Discrete data graphs

Figure 2: Continuous data graphs

practice.

An example of a histogram is given on the left of Figure 1. As the majority of examples we present, it is generated from sample data on air quality given in R and can therefore be easily reproduced. The particular histogram represents Ozone values over a period of 153, days and it contains both outliers as well as gaps in the data.

A boxplot (also referred to as a box and whisker plot) uses both areas and points to convey different aspects of the data being represented. The box is bounded at its ends using the lower and upper quartile and the median of the data is represented as a line segment that bifurcates the box. The user determines if the median is closer to either quartile to help assess the symmetry of the data. The whiskers extend out to the extremes of the data (minimum and maximum) as long as these are not thought to be too far from the bulk of the data as represented by the box. If points are too extreme, the whiskers extend as far as the furthest point that is not considered unusual. The outliers are then identified with single point markers. It is also common to use comparative boxplots to highlight the dissimilarities between subgroups within a dataset. One might draw a boxplot for males and another for females and then be able to compare the medians, quartiles, presence of outliers and the like.

2.2 Continuous Data Graphs

Although these graphs and their underlying data are not necessarily continuous in a strictly mathematical sense, they are meant to visually convey information that relies on the continuous interpretation of all data. That is, contrary to the discrete graphs, where, for instance, single bars conveyed interesting information in their own right, in contin-

uous data graphs the visual representation of single data points is meaningless without considering their relationship to some or all other data points in the graph. Therefore, we can say that while the data can be discrete, their interpretation has to be continuous.

As an example consider the scatter plot on the very left of Figure 2. The data is discrete, in the sense that there is a finite number of data points that are clearly distinct. However, when reading the graph one is not concerned with the single points and their value, but rather with the curve represented by the overall arrangement of the points in the two-dimensional grid, as well as with respect to the uniformity distribution data points or clusters they form. Consequently, presenting data point by point to a blind reader would be absolutely useless.

3. CREATING ACCESSIBLE DIAGRAMS

We generate accessible diagrams using the R statistical software application [8]. We produce diagrams in scalable vector graphics (SVG) together with rich semantic information in a bespoke XML format for the *DIAGcess* Javascript library [9], which enables screen reading and interactive exploration of diagrams in any browser. In this section we describe the process and the software packages involved.

R is a programming language and software environment for statistical computing that allows the efficient manipulation of statistical data and their graphical representation offering a host of different diagrams and rendering options. As R is commonly used by working statisticians, both in academia and industry, integrating the ability to generate accessible graphics directly should decrease the hurdle for

adoption. R has two major graphics systems, a base system that has existed since the initial developments of R and the alternative grid system. Since the latter provides low-level tools to create advanced statistical graphics and to programmatically augment and manipulate them, we have chosen it as the basis for our work, together with the gridSVG package [6] as the output device to generate SVG. Again, R has a variety of graphics' devices that create scalable vector graphics (SVG), however only gridSVG provides sufficiently detailed hierarchical structures in the SVG to support the semantic interpretation and manipulation that we need.

We semantically enrich the statistical data by directly augmenting the grid diagram model and generating XML with the structured semantic content. We have implemented the enrichment process as an extension to R's *BrailleR* [3] package, an add-on specifically targeted to the needs of blind users. In its initial form, BrailleR was intended to summarise graphical information available in text form by describing only what the sighted user can see, but without trying to convey any meta-information that could be gained by interpreting the shape of the graph. For example, in the case of a histogram, the generated text would only contain the total number of items, the midpoints of bars, (also called bins) and the counts of objects in each bin, while the shape of the distribution is left to the user's imagination to determine.

In our current work, however, we wish to go beyond this, enabling a blind user to perceive the same information as their sighted peers. Consequently, we use BrailleR to compute rich semantic information on a diagram, its components and their relationship to each other, generating a multi-layered hierarchy that we represent as in a bespoke XML structure to be directly used by the *DIAGcess* web front end.

DIAGcess is a Javascript library that allows for screen reading and interactive exploration of annotated SVG diagrams in any browser. It was originally designed for the navigation of chemistry diagrams [9] but has since been extended to deal with other, similarly annotated diagrams. It provides a hierarchical navigation model that enables the exploration of a diagram component-wise on different levels, while voicing information on the selected components together with magnification and synchronised highlighting.

The navigation model is derived from the generated XML structure, and associated to corresponding graphical components via the element IDs in the SVG, thereby effectively annotating the diagram with semantic and speech information. Both SVG and XML are loaded into the browsers DOM at run time, navigation is achieved via Javascript methods acting on the hierarchical structure, while enacting visual effects using CSS and SVG viewport changes. For uniform treatment of SVG regardless of its structure or source, it needs to be transformed into a canonical form. In particular, in order to achieve a homogeneous user experience for synchronised highlighting and magnification, SVG containing transformations of graphical components need to be rewritten such that any transformations are applied locally only.

4. GENERIC NAVIGATION MODEL

We employ a hierarchical navigation model that has been originally developed for the presentation of chemical diagrams [10]. The idea is to provide the user with a summary overview of the diagram they read, while also providing the possibility to explore components in more detail should they wish to do so. We thereby support a simple arrow key navigation between elements, using up/down to switch between layers in the hierarchy and left/right to move between components on the same layer.

For statistical diagrams we generally work with three layers that comprise: (1) top level summary, (2) major component layer, and (3) single component exploration. Note that the single component exploration can be further broken down into multiple hierarchical layers depending on the complexity of the information that is presented, and that navigation between major components is only possible on the second layer. That is, when exploring a single component on layer three, one can not move directly into an neighbouring component without moving back up one layer first. This has not only the advantage that component description and navigation can be fully localised, but also that the reader retains their orientation, which component of the diagram they are currently exploring.

During navigation the components of the diagram under consideration are screen read either with a short or detailed description, depending on the user's preference settings, together with synchronised highlighting and optional magnification of the elements in the SVG.

We describe our particular navigation model for graphs in general, before breaking down the differences in treating discrete and continuous data graphs. We illustrate our exposition with the example of the Ozone histogram from Figure 1, where Figure 3 presents some of the navigation steps in different states of highlighting and magnification.

The top level summary consists of the title only or, in verbose mode, the title together with a summary of the range of the graph. In the histogram example, the top level is aurally rendered as either "Histogram of Ozone" or "Histogram showing 9 bars for Ozone over the range 0 to 150 and Frequency from 0 to 30." and visually corresponds to the left-most diagram in Figure 3,

The major component layer for every two-dimensional function graph consists of the following four elements, where the first three are common to all graphs:

Title component: This comprises the title of the diagram as well as further information such as a subtitle or additional text. This information can be navigated during single component exploration.

X axis: This summarises values and range of the X axis. This corresponds to the second from the left image in Figure 3 for our histogram example, which is aurally rendered as "X axis Ozone" and "X axis Ozone ranges from 0 to 150" in verbose mode. In single component exploration the axis title and the single ticks can be read step by step. E.g., the exploration of "Tick mark 0" is shown in the middle diagram in Figure 3.

Y axis: This is analogous to the X axis.

Data content: This is the actual important information of the graph and its presentation as well as step-wise exploration varies for the different types of graphs.

For histograms the centre (see Figure 3 second from the right) is described in terms of number of bars and optionally in terms of the highest bars. These bars can then be explored stepwise from left to right on the single component level, presenting details on the values they represent. For
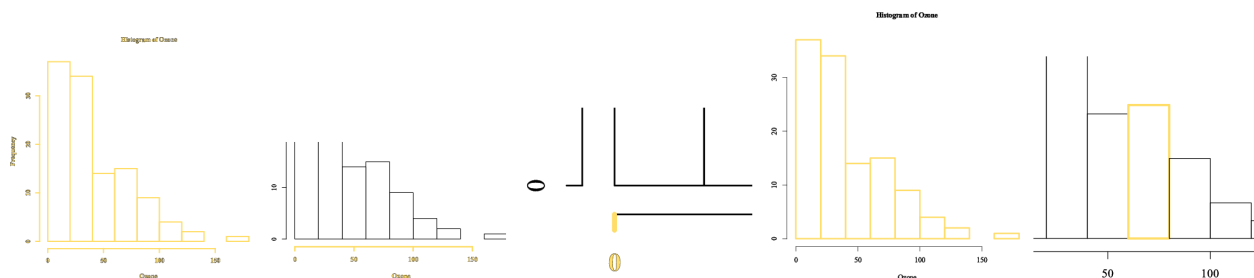
Figure 3: Step-wise exploration of a histogram

instance, the rightmost diagram in Figure 3 presents visually the exploration of fourth bar of the Ozone histogram, which is described as "Bar 4 at 70 with value 15" or "Bar 4 between x values 60 and 80 with y value 15 and density 0.00647" in verbose mode, respectively.

To enable a similarly meaningful exploration for the reader in the case of continuous data graphs we often need to artificially break up their data into single components. Consider time sequences for Ozone and Temperature in Figure 2) as an example. The time sequence for Ozone is only a partial function, that is there are actual gaps in the graph, which give us natural points to discretise the graph and present these separately in single component exploration.

If this option is not available — for example in the time-series for temperature — we break the data into sufficiently small chunks in BrailleR before the SVG is generated. During exploration components are then presented by summarising the data points they represent as well as relating it to the neighbouring elements by describing whether values decrease or increase with respect to them.

5. CONCLUSIONS AND FUTURE WORK

The presented work is implemented in the BrailleR package, which is available on the R Project's main package repository service known as CRAN (Comprehensive R Archive Network), and via GitHub. Acceptance onto the CRAN server demands a high standard of package validity and the existence of documentation. Although development benefitted from the fact that two of the participants are themselves blind, we have obtained feedback from four other power users, which was generally very positive but also pointed to some problems integrating with existing dyslexia tools. We are currently planning a more structured user study to obtain more statistically relevant feedback. Also at present, the production of accessible statistical diagrams works well for the gridSVG package as it provides sufficiently rich SVG, however the objective is to extend it also to other renderers in R. But despite this limitation we believe the work is a significant step towards providing blind people with greater access to statistical diagrams and we feel it goes some considerable way to enabling both students and professionals greater possibilities to equally participate in both education and employment.

While there are many different graph types that are what might be thought of as standard or traditional, the diversity of applications to which R has been applied means there are packages that create numerous different types of graphic, even some that might not be thought of as a statistical graph. There is room to develop the functionality using the existing building blocks for more two-dimensional graphs, other discrete data graphs (pie charts, stacked bar charts etc.), and such other graphics as directed networks and flow charts. In principle, if a graphic can be drawn by an R package using the grid graphics system, then we ought to be able to write the support functionality to interpret that graph. However, what bears further scrutiny is the question of whether all graphs readily lend themselves to the type of hierarchical exploration discussed here. One possible example of such a case is that of Venn Diagrams. Such graphics often depict the intersection, or union, of sets of data and are visually represented by overlapping circles. Further investigation is definitely required to devise alternative navigational strategies for cases such as these.

6. REFERENCES

[1] D. Fitzpatrick and D. McMullen. Distance learning of graphically intensive material for visually impaired students. In *Int. Conference on Computers for Handicapped Persons*, pages 219–225. Springer, 2008.

[2] J. Gardner. Can mainstream graphics be accessible? *Information Technology and Disabilities*, 14(1), 2014.

[3] A.J.R. Godfrey, T. Bilton, and V. Sorge. *BrailleR: Improved Access for Blind Users*, 2017. R package version 0.25.6.

[4] A.J.R. Godfrey and M.T. Loots. Advice from blind teachers on how to teach statistics to blind students. *Journal of Statistics Education*, 23(3):1–28, 2015.

[5] D. McMullen and D. Fitzpatrick. Combining web services and daisy for the production and delivery of audio tactile diagrams. In *Int. Conference on Computers for Handicapped Persons*, pages 12–19. Springer, 2010.

[6] P. Murrell and S. Potter. The gridSVG package. *The R Journal*, 6(1):133–143, June 2014.

[7] D. Parkes. Audio tactile systems for designing and learning complex environments as a vision impaired person: Static and dynamic spatial information access. *Learning Environment Technology: Selected papers from LETA 94,*, pages 219–223., (1994).

[8] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[9] V. Sorge. Polyfilling accessible chemistry diagrams. In *International Conference on Computers Helping People with Special Needs*, pages 43–50. Springer, 2016.

[10] V. Sorge, M. Lee, and S. Wilkinson. End-to-end solution for accessible chemical diagrams. In *Proc. of the 12th W4A Conference*, pages 6:1–6:10. ACM, 2015.