The BrailleR Project

A. Jonathan R. Godfrey
Institute of Fundamental Sciences, Massey University
Palmerston North, New Zealand

Abstract

If a blind student is to truly gain access to statistical analyses, they will need to be able to successfully complete a course in statistics at university level. To do this, they must know how the graphical techniques used in the sighted world look and are used. Generation of tactile images can show the blind student what a particular graph does in a general sense, but greater understanding will come from knowing how these graphs are generated by sighted students.

Once the student has completed their first course in statistics, they may embark on research at a university, or head out into industry to apply their knowledge. Irrespective of the direction they choose, they will need certainty in being able to independently create graphs for the sighted readers of their work. Creation of tactile images that provide the same representation of the images placed in documents do provide a solution, but all too often blind people do not have access to the right software and hardware to generate tactile images for themselves with the immediacy required.

The BrailleR project is an extension to the R statistical software application. It aims to provide textual information to the blind user in conjunction with the graph that would be placed in the final report. Key to its success are the fact that the look and feel of R is heavily based on a statistical programming language known as S, which creates a record of the elements that are processed into a graph. BrailleR relies on the retention of these objects and processes the information into a concise textual representation of the graph at the same time the graph is created, without much extra work being required by the blind user. This textual information will enhance the experience of any tactile images the user may have available. BrailleR could increase the capacity of blind people needing to create graphs independently in many mathematical and scientific endeavours and should therefore improve their educational and employment prospects.

1 Introduction

Access to information is crucial for the blind person's success in education, but transferring the knowledge about the existence of techniques into actually being able to perform those tasks is what will make the blind person employable. This presentation builds upon one aspect of my presentation [4] at the 2009 Workshop on E-Inclusion in Mathematics and Sciences.

As blind users, we are currently forced to find solutions that work for us as individuals. It would be preferable to have some community wide set of common practices that are well understood and used. Attempts to share knowledge such as the Summer University[1], Access to Science website[2], and email lists such as the blindmath list[3] hosted by the National Federation of the Blind[4] are wonderful resources, but this author believes a fundamental step change is required if the blind community is to actually fulfil its potential in mathematical and scientific disciplines.

The BrailleR project[5] aims to turn information presented in a visual medium into a medium that is simple to work with, efficient, and complementary to the skill set of as wide a group of blind people as possible. With this in mind, a plain text solution is favoured above tactile images as there is no need for any additional adaptive technology (hardware or software) over that used for the blind person's other work. Information presented in text is readable in braille or heard via synthetic speech, and is therefore only limited to the user's skill set.

G. Sutcliffe, A. Voronkov (eds.): DEIMS 2012, volume 1, issue: 1, pp. 1-7

Other work is being done to create tactile images that replace the printed ones sighted people deal with all the time in statistical work. Tactile images do have a role to play in educating the blind and providing them with access to information, but their greatest drawback remains their lack of immediacy. Rendering a graphic into a master and then producing this for consumption are tasks receiving much attention, but the value for such efforts is probably greatest for situations where one master can be used to create multiple copies for wider consumption, perhaps even internationally. Under a sighted paradigm, many graphs in statistics work are developed for one-time use and discarded. It may be some time until the effectiveness of tactile image hardware and software is ready to meet this need and in the meantime we must find ways of gaining access to the graphical information without reliance on another person's ability to translate the visual medium into something more useful for us. We must also acknowledge that the number of blind people with their own tactile image hardware and software is regrettably low. Perhaps we can look forward to the time when most blind people can have access to a refreshable tactile display that can display images that compare with the best that is possible in hardcopy today.

2 Accessibility of graphics in statistical software

Access to graphical representations of information from mathematical or statistical software is quite limited, and therefore limiting the blind user's capacity. To this author's knowledge, no mathematical or statistical software has the capability of directly linking to any hardware or software solutions that make the information immediately available. A very small number of applications do have some capacity to link in with scalable vector graphics, but this capacity does not in itself create access.

In addition to the inability to render graphs in usable form, many software applications are making use of graphics when once the information would be presented in text form with an accompanying graph. — SPSS[6] is an example of this, and to a lesser extent so is Minitab[7]. Some software applications have retained their historical links to the days when graphs and tables were rendered in well controlled monospace fonts — SAS[8] and Minitab are just two examples. In these applications the user may be able to produce an ASCII graphic instead of the more commonly used high resolution graphs expected of today's software (and user).

Alternatives to producing graphics do exist, but the shift in the way the sighted world is operating with everyone expected to use a computer to complete tasks, means the practical value of these (old fashioned) techniques is reducing.

Attempts such as the IGraph-Lite system, see [3], to extract useful information from graphical information created by another person set a standard that must be surpassed when the need for the blind person to create the graph is to be considered. The scripts for screen reader software that attempt to interpret the graphs created by common spreadsheet software are another example of what has been tried in the past. Both of these attempts are limited to the capacity of the system being used to create the graphs, and in both cases, the range of graph options are limited.

I therefore conclude that until such time as adaptive technology for creating immediate access via tactile images is commonly available, other solutions will continue to be relevant. In fact, even once the immediacy issue is overcome, I still feel there will be a place for solutions such as BrailleR as a complementary solution rather than a substitute.

3 Accessibility of R statistical software

I have reported on the ability to use R statistical software[9] almost immediately after installation; only one minor change is required and can be achieved in less than a minute even when explained via email or a telephone call. Students at the 2011 Summer University held in the Czech Republic[10] were able to do this for themselves in a classroom setting.

At the 2011 Summer University, I showed students that they could use R to gain access to the information presented in a graph, in a manner almost unparalleled by other statistical software — an obvious exception to this rule is the commercial software known as S-Plus[11], which has the same syntax as R because both were based on the same programming language called S[12] developed at Bell Labs in the mid 1970s, principally by John Chambers. The public release is more generally recognized through publication of the 'Brown Book' [1].

In its newer form [2], the S language is object oriented and functional, in that every task is based on the creation of the needed data structure that is then manimpulated by functions and methods to generate the desired outcome. The data object is often discarded once it has served its purpose, but can be stored if the user so chooses.

A simple way for blind users to access the information used to create a graph is to ask R (or S-Plus) to print the object, using the `print` command. For example, the sighted user wanting a histogram of 1000 randomly chosen values from a standard normal distribution would type

```
hist(rnorm(1000))
```

while the blind user wanting to know how this histogram would look could type:

```
print(hist(rnorm(1000)))
```

The output from such a command appears in Figure 1. As well as providing the user with the text shown in this figure, R will also have created the histogram (not shown). If the user can interpret the text content given in the figure, they will have some idea of what appears in the histogram. Not all of the printed information is relevant to the end user however so some further education or perhaps further processing is still required.

This approach was the one I demonstrated at the 2011 Summer University. It is workable but not elegant. On some occasions the data structure listed is just a list of the actual data itself and is therefore not giving the blind user the same information as the sighted user gets from a graph. The blind user needs some sort of summarisation tool to replace the graph, and in many instances is unlikely to have the skill to develop a sound statistical solution for themselves. It is also reliant on the user knowing what the various elements printed out are, and how they link to the visual object created for the sighted world.

4 BrailleR in action

The BrailleR project is aimed at taking the information created by various functions in R and creating a text printout that reduces the amount of information that needs to be processed by the blind user. In cases like that for the histogram example previously introduced, this is a fairly simple task. For other situations, some further work will need to be done so that the textual information is a useful summary of the graphical information without crossing over the line of interpretting the information for the user. The necessary functions are packaged for public distribution and form the BrailleR package [5].

Discussion of the BrailleR Project and its successes and opportunities for further success is reliant on understanding just two of the structures used in the S language. These are “methods” and “classes”.

Writing a function that creates a text representation of the histogram developed by R is made easy because the object created in the background is given a “class” as part of its creation.

Many commands will lead to an object being created, sometimes with a class attribute being explicitly stated, sometimes implicitly, but sometimes without any class attribute at all. Assigning a class to an object means that we can write functions that relate to all objects of a particular class using a general

```
> print(hist(rnorm(1000)))
$breaks
 [1] -3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  2.5  3.0  3.5
[16]  4.0

$counts
 [1]   1   8  20  45  89 165 198 195 119  89  48  15   7   0   1

$intensities
 [1] 0.002 0.016 0.040 0.090 0.178 0.330 0.396 0.390 0.238 0.178 0.096 0.030
[13] 0.014 0.000 0.002

$density
 [1] 0.002 0.016 0.040 0.090 0.178 0.330 0.396 0.390 0.238 0.178 0.096 0.030
[13] 0.014 0.000 0.002

$mids
 [1] -3.25 -2.75 -2.25 -1.75 -1.25 -0.75 -0.25  0.25  0.75  1.25  1.75  2.25
[13]  2.75  3.25  3.75

$xname
[1] "rnorm(1000)"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

Figure 1: Output resulting from printing a histogram object

approach. We might need a function to print the object out in an easy to use fashion; we may need to plot the results in a graph; we may need to create a different kind of object that summarizes the original object in some way. These are just three tasks common to objects of many classes.

The `hist` command used earlier does create an object with an explicit class attribute. The class is called a “histogram” and only one specific function exists for this class, that being a function to plot the histogram. Sighted users don’t need an explicit print function for a histogram, nor does this summary graph need further summarisation.

If we write a series of commands that perform a similar task on a selection of different classes we then call each function in this family of functions a “method”. A complete method will have a base function that informs the software that there is a family of functions written for different classes, and that a method has been written for the default action, which is applied if no specific method exists for a class. For example, the print method includes specific functions

```
print,  print.default,  print.matrix, ...
```

If we issue the command `print(x)`, and we know that *x* is a matrix, the `print` method will employ the `print.matrix` function to display the matrix. The example given earlier that printed the results of

```
> VI(hist(rnorm(1000)))
This is a histogram, with rnorm(1000)  marked on the x-axis,
unless you explicitly used the xlab argument.
There are a total of 1000 elements for this variable.
It has 15 bins with equal widths, starting at -3.5 and ending at 4 .
The mids and counts for the bins are...
 mid = -3.25   count = 1
 mid = -2.75   count = 8
 mid = -2.25   count = 20
 mid = -1.75   count = 45
 mid = -1.25   count = 89
 mid = -0.75   count = 165
 mid = -0.25   count = 198
 mid = 0.25   count = 195
 mid = 0.75   count = 119
 mid = 1.25   count = 89
 mid = 1.75   count = 48
 mid = 2.25   count = 15
 mid = 2.75   count = 7
 mid = 3.25   count = 0
 mid = 3.75   count = 1
```

Figure 2: Output resulting from using the `VI` method on a histogram object.

the histogram object used the `print.default` function as there is no `print.histogram` function in existence.

The usefulness of methods is dependent on the use of classes being employed when objects are created. Not all objects are given a class so the default method must be constructed carefully. There are actually only a few basic data structures to work with, the easiest and most common of which is called a “list”. The printed results in Figure 1 are a list, but note the last element of the list that states the class of the object. Adding this extra attribute to the data object is a minor matter that can have very powerful consequences.

The beginnings of the BrailleR Project were formed on the idea of writing a method that would provide the summarised text version of the graph object created. To this end, a method was started with the functions:

```
VI,   VI.default,   VI.histogram
```

For illustration purposes and for the sake of brevity, only the outcome of applying the `VI` function to the histogram object printed in Figure 1 is now given. The `VI` function sees that the object in question is a histogram and applies the `VI.histogram` function which does the work. The results from applying the `VI` method on the histogram appear in Figure 2. HistObject.

There is some difficulty in situations where the class of the graph object is not specified. In such cases, the `VI` method will revert to the printing out of the object as a list. It is possible to call any of the method’s functions directly and this is how I am circumventing the lack of a class at present. The functions will pass the arguments to the vanilla R function to do the work, but know the desired graph type and therefore be able to interpret the arguments and returned list object in a suitable manner. Functions can be written specially that match the form of the method functions, so that if the class attribute is included in future versions of R, the code will function as desired.

Further development of functions that do not explicitly create a graph, but would use the `plot` method to do so on most occasions, can also be developed quite easily. For example, fitting a linear model is often followed by an analysis of the residuals from that model. Sighted people create a set of diagnostic plots to assist them in this regard, but we can create functions that extract the necessary information using vanilla R commands.

Another issue for blind users of any mathematical or statistical software is how the work that has been done will be transferred into a form that can be included in reports or assignments. Standard methods of working with R are possible as a blind user. I have to date found just one task that sighted users find very easy but was beyond the reach of the blind user; this is no longer an issue.

A sighted user can highlight a section of the output window (including commands and results) and copy the text into a document using the mouse. This task is done easily in word processing or text documents using keyboard commands by the blind user, but is often impractical or impossible when done within some software applications. I have been able to provide a solution within the BrailleR Project, by extracting a tool developed by another R user for a completely different purpose. The add-on package for R, called ‘TeachingDemos’ [6], provided this tool. The original purpose was to quickly retain the output from an R session for distribution to students, and later in its development, to create output files (in MS Word or HTML) that would log the progress through an assignment question or project. The code for the basic text output and commands being saved into plain text files now forms part of the BrailleR package.

5 The work ahead

The BrailleR Project is likely to evolve over time. The work on the project will be dependent on interest being shown by blind users of R, and perhaps some willingness being shown by people who could make the work so much easier.

It is my hope that I can encourage the R development core team to create more classes. The additions are almost trivial from their perspective, but the workload for me as a programmer will reduce markedly if I can write simpler code to interpret the graphs. Knowing what type of graph has been created by an object rather than writing code that makes a reasonable attempt at guessing what type of graph was created is a key example.

I will also need feedback from students and users of R, and perhaps their lecturers, tutors, and teachers, to see which ideas are working well for the blind users. At present, the attempts being made are based on my own experiences and desires; adding perspectives is welcomed as a crucial pathway for the project’s development.

A key contribution is possible in the area of choosing appropriate default text for the BrailleR output. The text representation needs to be more efficient than using R in its vanilla form. To this end, I have put some time into thinking about which text will be most suitable for an audience that will be listening to the output using synthetic speech, perhaps in the user’s second language, as well as the braile readers who may or may not be able to use contracted english braille. At first, I had thought this task would be made easier by reviewing resources created by transcribers who record books for the blind, but my investigations show that the descriptions used in spoken words do not always equate to efficient braille or synthetic speech and often provide more interpretation than I believe is appropriate in educational settings.

Collaboration from others in any role will feed my own enthusiasm for the project, especially those whose skills and knowledge complement my own. I am already grateful for the additions to the BrailleR Project that have come from the work of others who have shown a willingness to contribute.

Web resources noted in the text

[1]The home page for the Summer University is found at `http://icchp-su.net/`

[2]The Access to Science web resource is found at `http://www.access2science.com/`

[3]Join the blindmath list by visiting `http://www.nfbnet.org/mailman/listinfo/blindmath_nfbnet.org`

[4]The NFB is found at `http://www.nfb.org/`

[5]The BrailleR Project home page is at `http://r-resources.massey.ac.nz/BrailleR`

[6]SPSS is the common name of a statistical software product now owned by IBM. A demo version is available from `http://www-01.ibm.com/software/analytics/spss/downloads/`

[7]Minitab is commercial statistical software. Obtain a demo version from `http://www.minitab.com/`

[8]SAS is commercial statistical software, usually purchased by institutions rather than individuals due to its high cost. It does not have a demo version but can be reviewed at `http://www.sas.com/`

[9]R is available for free download from `http://www.R-project.org`

[10]See `http://icchp-su.net/?q=node/108` for the details of the workshop I ran on R at the 2011 Summer University.

[11]S-Plus is commercial software built on the S language and now marketed by TIBCO Software Inc.

[12]visit `http://en.wikipedia.org/wiki/S_(programming_language)` for a useful summary of the history of S.

References

[1] R. A. Becker and J. M. Chambers. *S: An Interactive Environment for Data Analysis and Graphics*. Wadsworth & Brooks/Cole, Pacific Grove, California, 1984.

[2] R. A. Becker, J. M. Chambers, and A. R. Wilks. *The New S Language: A Programming Environment for Data Analysis and Graphics*. Pacific Grove, California, 1988.

[3] L. Ferres, P. Verkhogliad, G. Lindgaard, L. Boucher, A. Chretien, and M. Lachance. Improving accessibility to statistical graphs: The igraph-lite system. *ASSETS'07*, pages 67–74, 2007.

[4] A. Jonathan R. Godfrey. Are statistics courses accessible? In *Proceedings of the Workshop on E-Inclusion in Mathematics and Science 2009*, pages 72–80, Fukuoka, Japan, 2009.

[5] A. Jonathan R. Godfrey. *BrailleR: Improved access for blind useRs*, 2011. R package version 0.1, available from `http://R-resources.massey.ac.nz/BrailleR`.

[6] Greg Snow. *TeachingDemos: Demonstrations for teaching and learning*, 2010. R package version 2.7, available from `http://CRAN.R-project.org/package=TeachingDemos`.