G Model
EPP 1352 No. of Pages 9
ARTICLE IN PRESS
Evaluation and Program Planning xxx (2016) xxx–xxx
Contents lists available at ScienceDirect
Evaluation and Program Planning
journal homepage: www.elsevier.com/locate/evalprogplan

R-CMap—An open-source software for concept mapping

Haim Bar, PhD.[a,*], Lucas Mentch, PhD.[b]

[a] Department of Statistics, University of Connecticut, United States
[b] University of Pittsburgh, Dept. of Statistics, United States

ARTICLE INFO

Article history:
Received 26 July 2016
Accepted 22 August 2016
Available online xxx

Keywords:
Concept mapping
Open-source
The R programming language

ABSTRACT

Planning and evaluating projects often involves input from many stakeholders. Fusing and organizing many different ideas, opinions, and interpretations into a coherent and acceptable plan or project evaluation is challenging. This is especially true when seeking contributions from a large number of participants, especially when not all can participate in group discussions, or when some prefer to contribute their perspectives anonymously. One of the major breakthroughs in the area of evaluation and program planning has been the use of graphical tools to represent the brainstorming process. This provides a quantitative framework for organizing ideas and general concepts into simple-to-interpret graphs. We developed a new, open-source concept mapping software called R-CMap, which is implemented in R. This software provides a graphical user interface to guide users through the analytical process of concept mapping. The R-CMap software allows users to generate a variety of plots, including cluster maps, point rating and cluster rating maps, as well as pattern matching and go-zone plots. Additionally, R-CMap is capable of generating detailed reports that contain useful statistical summaries of the data. The plots and reports can be embedded in Microsoft Office tools such as Word and PowerPoint, where users may manually adjust various plot and table features to achieve the best visual results in their presentations and official reports. The graphical user interface of R-CMap allows users to define cluster names, change the number of clusters, select rating variables for relevant plots, and importantly, select subsets of respondents by demographic criteria. The latter is particularly useful to project managers in order to identify different patterns of preferences by subpopulations. R-CMap is user-friendly, and does not require any programming experience. However, proficient R users can add to its functionality by directly accessing built-in functions in R and sharing new features with the concept mapping community.

1. Introduction

“Well begun is half done”[1]; is the essence of project planning and the key to the final outcome of the project. However, planning complex projects usually involves input from many sources, including future consumers, managers, engineers, etc. Assembling and analyzing input from diverse sources with varying levels of expertise and different priorities, and converting these inputs into project objectives and detailed plans of execution is a major challenge. Planners require tools and methods to assess which goals are perceived as important and feasible, and how they relate to each other. To this end, the group concept mapping approach provides the visualization and analytical tools with which planners can:

1. identify specific project goals,
2. determine the required resources and expertise,
3. assign priorities, and
4. define objective measures of success.

Group concept mapping helps planners achieve these goals by employing a six-step process (Trochim & McLinden, 2016, this issue). In this article we focus on two of the six steps, namely, (i) the multivariate statistical analysis, which provides the mathematical foundations for the inferential process, and (ii) the generation and interpretation of the concept maps, which facilitate the analysis by summarizing input from a large number of stakeholders in graphical form. These two steps are critical to understanding the composite thinking among a diverse group of stakeholders with varying levels of expertise, knowledge,

* Corresponding author.
E-mail address: haim.bar@uconn.edu (L. Mentch).
[1] Attributed to Aristotle.

http://dx.doi.org/10.1016/j.evalprogplan.2016.08.018

responsibility, and interest, and to translating the input into a prioritized plan of action.

We introduce a new, freely available, open source concept mapping tool, called R-CMap, which not only implements the traditional multivariate analysis and graphical representations, but also introduces new analytical and graphical tools for concept mapping. Furthermore, it allows users who are familiar with the statistical programming language R to add functionality and features. Based on similar experiences in other fields (e.g. bioinformatics) we anticipate that the release of an open source concept mapping software will contribute to accelerated advances in the fields of planning and evaluation, as more researchers will be able to contribute and incorporate new methods and visualization tools. In the following section we very briefly describe the programming language R and some of its main features and capabilities. We then discuss some implementation details and provide examples of its statistical and graphical capabilities, and conclude with a discussion.

2. A brief introduction to R

The R programming language (R Core Team, 2016) is a powerful programming tool used extensively throughout the scientific community to perform statistical analysis, computing, graphing, and data mining that is freely available under the terms of the GNU General Public License as published by the Free Software Foundation, on multiple platforms, including Windows, MacOS, and linux. R contains a wide range of statistical tools, including linear regression, hypothesis testing, categorical data analysis, time series, and clustering, just to name a few. It is constantly evolving, with new features and tools becoming available frequently. R users also benefit from its collaborative nature, since they can download contributed packages that extend the core (but very extensive) functionality of the language, or make their own implementations available to the public as packages, typically distributed via the Comprehensive R Archive Network (http://www.cran.r-project.org). Installing and updating contribute packages is very simple even to the novice user, via the install.packages() and update.packages() commands.

R is an implementation of the S programming language (Becker & Chambers, 1984), which was first designed and developed at Bell Labs in 1975–1976. The first version of R became available in 1997, and became officially part of the GNU project (www.gnu.org). The GNU project was initiated by Richard Stallman from MIT in 1983, with the goal of developing free software. R is free of charge and can be distributed under the GNU General Public License, but the word "free" in "free software" actually refers more broadly to the user's freedom to use the software, adapt it to their needs, and share it by redistributing the original, as well as adapted versions. To read more about the free-software philosophy and how it differs from "open-source", see (Stallman, 2014).

R programs are "scripted", which means that the user does not need to perform additional steps to run the code. In particular, in contrast to some other languages (like C and Java), one does not have to compile or link the code, and R programs are usually entirely portable from one operating system to another, meaning that in most cases, an R program which was created on one computer will run on any other computer without needing any modifications. Another advantage of scripted languages, such as R, is that one can run individual lines or commands, without running an entire program. This is very useful for debugging, performing simple calculations, and creating customized functions.

Most practitioners use R in a "command line" mode, as opposed to a "graphical mode". Thus, people who are used to Excel or Visual Basic, or other menu-driven tools, may find it less intuitive initially. However, many tools, such as RStudio (RStudio, 2015), offer user-friendly features like viewing current values of variables and data structures and viewing and manipulating plots to assist less technical users. Two very useful recent additions to the RStudio development environment include "Shiny" (Chang, Cheng, Allaire, Xie, McPherson, 2016) which allows developers to deploy R programs as web applications, and "knitr" (Xie, 2016) which allows users to integrate text with R code and easily generate publication-ready documents. The current version of R-CMap uses the shiny package in a local-host mode, which means that when one starts the program, "Shiny" starts a web-server on the user's machine. We will soon introduce a version of our software which uses the remote-host mode, which means that the web server runs on a remote machine which is accessible through the internet. The remote-host version allows users to use R-CMap without having to install R or any additional packages on their computer.

In addition to the tools like RStudio and the documentation provided by authors of thousands of free packages, users can also find countless tutorials, online documentation sources, active discussion groups, and books. This contributes to the fast rate in which R is adopted by different sectors in both academia and industry. Though this flexibility and functionality are available, it is worth stressing again that our R-CMap package can be executed via a GUI, eliminating the need for technical programming skills from users.

3. R-CMap—statistical and graphical capabilities

3.1. Foundational improvements and extensions

Implementing our concept mapping software in R allows users across a variety of platforms to have open and free access to well-known concept mapping functionalities. Another major advantage to using an open-source language such as R is that we may incorporate a wide variety of statistical and graphical functions already available in order to extend the current concept mapping methodology to new situations, and develop improvements at each stage of the process to help users better understand the data. Some of the extensions and improvements described below are available in the current version while others require more extensive refinement and will be released in future updates.

3.2. New options for multidimensional scaling

Throughout this paper, we discuss our procedures in terms of a classic framework in which a number of participants are given several statements and asked to sort them into piles as they see fit. The sorted piles are represented as incidence matrices, one for each participant, where cell (i,j) contains 1 if the participant put statements i and j in the same pile and 0 if the statements were placed in different piles. The incidence matrices are added, and their sum is denoted by S. It is then normalized by dividing by the number of participants, to construct a similarity matrix, denoted by S^*. The normalization step insures that if two statements co-occur in the same pile for every participant, then the similarity score is 1. On the other end of the spectrum, a similarity score of 0 represents two statements that were not put in the same pile by any participant. Equivalently, one can consider the derived distance matrix, $D = I - S^*$, where I is the identity matrix, and both I and S^* are $k \times k$ matrices, and k is the number of statements. Thus, column i in the distance matrix contains all the pairwise distances between statement i and all the other statements. These pairwise distances are not sufficient to determine the overall configuration of all k statements in a k-dimensional space. Furthermore, it is impossible to visualize a k-dimensional space. Therefore, the next step in the concept mapping process involves transforming the