ScattR – A Shiny App for Exploratory Data Analysis

Keywords: R, shiny, data exploration, scatterplot, imputation visualization

1. INTRODUCTION

Any statistical data analysis aiming to answer scientific questions, derive solutions to given problems, or make predictions about unknown outcomes, should be preceded by an intensive exploration of the (initially unknown) data structure. However, there is always a trade-off between the time available to examine the data and the time needed to develop sophisticated modeling approaches for an in-depth analysis. Particularly, this applies when datasets are large, carry many features of interest and inherit a hierarchical structure. Using the software R [1] and the package *shiny* we created *ScattR*, which offers a user-friendly interface to compensate time usually consumed by coding in Statistical Software for an interactive data exploration with the scatterplot [2].

2. METHODS

ScattR is designed to work with different input formats by using the *haven* and *readr* package. Regarding proprietary input formats, we developed a series of functions to guess adequate column types for the relevant input fields of the application such that the user is only presented with meaningful variables for each option. The *ggplot2* package [3] offers a flexible and customizable environment to create an informative scatterplot based on the user's needs.

2.1. USER INPUT

The user can either upload a file from disk directly into the application or use a previously created internal R object. The file can be in plain .csv format or in the platform-dependent formats .dta (Stata) or .sas7bdat (SAS). Next, the user can define up to three columns as so called "layers", which are used for subsetting inherent hierarchical structures. It might be sensible to put the sequence of layers in logical order such that the bottom layer can be deduced from higher layers.

2.2. FEATURES

Variables suitable for plotting on the axes of the scatterplot are guessed automatically, as well as factor variables which are used for plotting in higher dimensions by coloring or grouping the observations. The user can switch between standard and logarithmic axes scaling to handle highly-skewed data. The axes limits and labels are always adapted dynamically given all currently plotted units. One input field can be assigned with a variable that is displayed upon a mouse-click on a scatter dot. Properties of the distribution of the data, like the total number of observations, zero values, negative values and missing values of the axes variables are calculated simultaneously. Adjustable color transparency of the scatter dots and rectangular binning are available to tackle over-plotting. Furthermore, the user can use a local regression smoother to find a relationship in the plotted variables, not immediately visible.

2.3. LAYOUT

The application is organized in a top-down manner and starts with supplying a file to the application. Then it is subdivided by the options data selection, standard plot, advanced plot and export. In figure 1 the user wants to explore the "ses" dataset from package *laeken* [5] which is already available in her R environment. Initially, the user can choose between loading the complete dataset, which is in general more suitable for smaller to mid-size datasets, and/or filtering the dataset by predefined layers. The latter is useful for larger datasets or categorical variables with many values, which would cause a messy plot when used in the color, column or row dimension. The user defined the variable "location" as the first layer and the variable "NACE1" as the second layer. Hence, a value from layer 1 has to be selected to subset the data. The following layer selections are deduced from the first one and are optional selection inputs. This means that upon selecting a value from layer 1 (here "AT1") the user is only presented with values of layer 2 that exist for layer 1 (here "H-Hotels", among others). The standard plot options cover the variable selection and scaling of the axes as well as the mouse click event. Advanced plot features provide coloring, grouping and other fine-tuning options. At last, the user can download and save the created plot to a .png file.

3. RESULTS

Initially created to get a deeper insight in the data structures of business surveys, we consider *ScattR* suitable for a broad range of applications, such as the visualization of imputed data [6] or detecting and communicating potentially erroneous data [7]. To visualize imputed data, the user can compare imputed values to observed values by creating/choosing a variable that indicates the presence of imputed values. The user might then check her imputation model on deeper aggregates of the data by either subsetting through the layers of the data or assigning column and/or row dimensions. This can increase confidence whether the imputation model is working as intended by reducing time required for coding substantially. To communicate striking patterns or observations the user utilizes the id-variable to process and review the relevant units. In order to do that, the application features a "Lookup" panel, where the user can search the current plotted observations by their id-variable and look up all remaining variables in the data set. Although it is more efficient to handle outlier-detection automatically, there are always borderline-cases where it is not obvious whether an extreme value is a faulty record or a statistical outlier. In such cases communication and subject-specific knowledge are required.

4. CONCLUSIONS

ScattR may be part of the official statistician's toolkit. First, it can be applied to a broad range of problems starting from visual data exploration preceding empirical analyses over detecting faulty micro-data to the visualization of imputed data. Second, it is free, easy to maintain and flexible as additional components for new problems can be integrated at short notice. Third, it can improve our data literacy. In the future, we want to automate the decomposition of hierarchical structures such that no user input in advance is required. The shiny app will be distributed in an R package with additional helper functions to increase the usability among less experienced R users.

REFERENCES

[1] R Core Team, R: A language and environment for statistical computing. (2018), URL: https://www.R-project.org.

[2] Unwin A., Theus M. and Hofman H. (2006) Graphics of Large Datasets: Visualizing a Million (Statistics and Computing).

[3] Hadley, W. (2011) ggplot2. *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 3, No. 2, pp. 180–185.

[4] Templ M. and Alfons, A. (2013) Estimation of social exclusion indicators from complex surveys: The R package laeken. *The Journal of Statistical Software*, Vol. 54, No. 15, pp. 1–25.

[5] Templ M., Alfons, A. and Filzmoser, P. (2012) Exploring incomplete data using visualization tools. *Journal of Advances in Data Analysis and Classification*, Vol. 6, No. 1, pp. 29–47.

[6] Barnett, V. (1978) Outliers in Statistical Data.

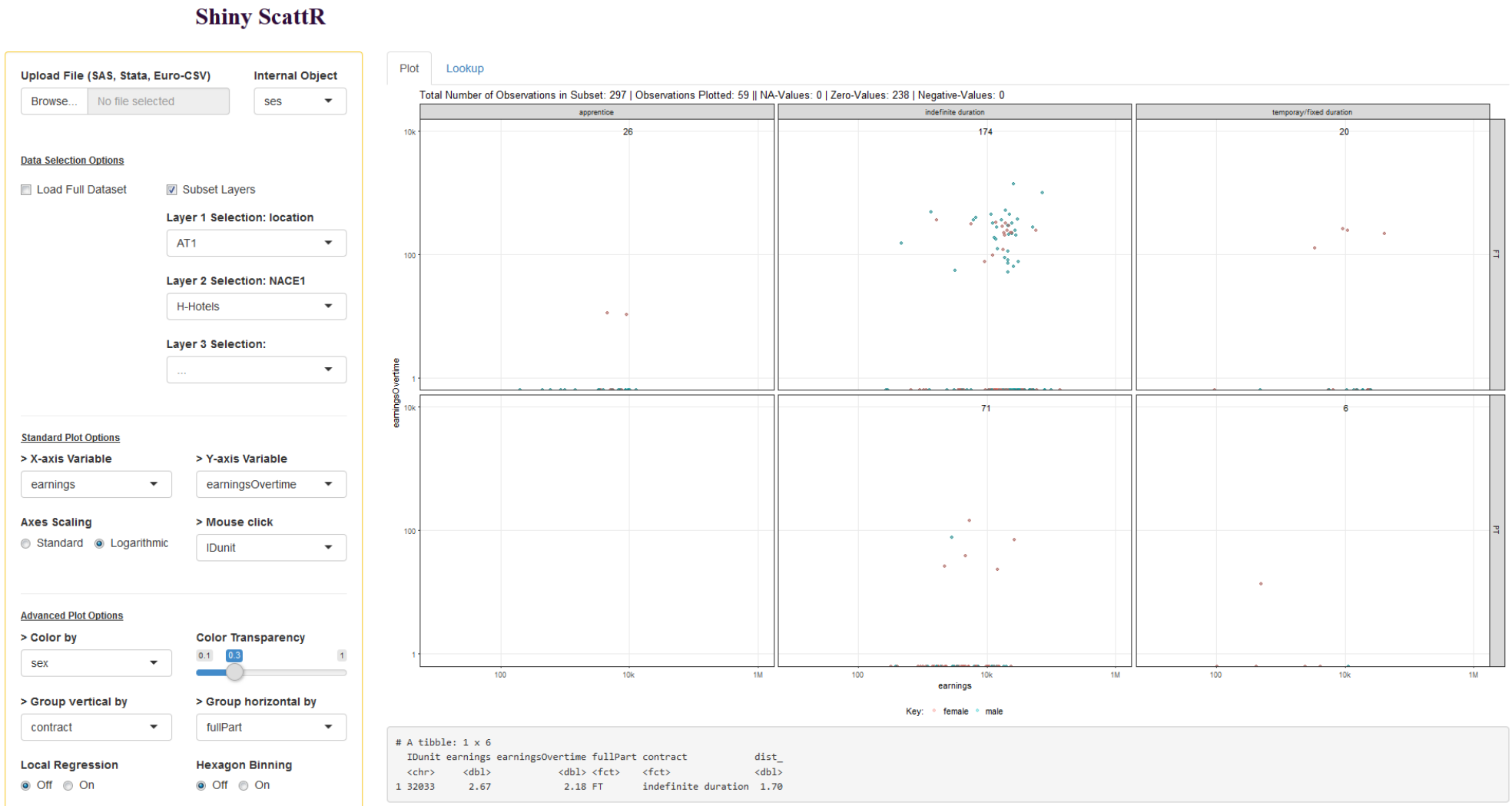

Figure 1. Sidebar and plot panel of the shiny app.