Visualization of large multivariate datasets with the `tabplot` package

Martijn Tennekes and Edwin de Jonge

June 1, 2011

Abstract

The tableplot is a powerful visualization method to explore and analyse large multivariate datasets. In this vignette, the implementation of tableplots in R is described.

1 Introduction

The tableplot is a visualization method that is used to explore and analyse large datasets. By using tableplots, data analysts are able to observe the relationships between the variables, discover strange data patterns, and check the occurrence and selectivity of missing values.

An example of a tableplot applied to the diamonds dataset of the ggplot2 package is illustrated in Figure **??**. Each column represents a variable. The whole data set is ordered according to one or more columns (in this case, carat), and then grouped into row bins. Algorithm 1 describes the creation of a tableplot into detail.

2 The `tableplot` function

To illustrate the `tabplot` package, we will use the diamonds dataset that is provided in the `ggplot2` package. This dataset contains information about 53,940 diamonds. There are 7 continuous variables and 3 categorical. In order to illustrate the visualization of missing values, we add several NA's.

```
> require(ggplot2)
> data(diamonds)
> is.na(diamonds$price) <- diamonds$cut == "Ideal"
> is.na(diamonds$cut) <- (runif(nrow(diamonds)) > 0.8)
```

A tableplot is simply created by the function `tableplot`:

```
> tableplot(diamonds)
```

Algorithm 1 Create tableplot

Input: Tabular dataset t, column i_s of which the distribution is of interest.

$t' \leftarrow$ sort t according to the values of column i_s.
Divide t' into n row bins according to the order of t'.
for each column i **do**
    if i is numeric **then**
        $m_{ib} \leftarrow$ mean value per bin b
        $c_{ib} \leftarrow$ fraction of missing values per bin b
    end if
    if i is categorical **then**
        $f_{ijb} \leftarrow$ frequency of each category j (including missing values) per bin b
    end if
end for
for each column i **do**
    if i is numeric **then**
        Plot a bar chart of the mean values $\{m_{i1}, m_{i2}, \ldots, m_{in}\}$. A logarithmic scale can be used. The fraction of missing values $\{c_{i1}, c_{i2}, \ldots, c_{in}\}$ determines the lightness of the bar colour. The lighter the colour, the more missing values occur in bin b. If all values are missing, a light red bar of full length is drawn.
    end if
    if i is categorical **then**
        Plot a stacked bar chart according to the frequencies $\{f_{i1b}, f_{i2b}, \ldots\}$ for each bin b. Each category is shown is a distinct colour. If there are missing values, they are depicted by a red colour.
    end if
end for

The result is depicted in Figure **??**. By default, all variables of the dataset are depicted. With the argument `colNames`, we can specify which variables are plotted.

Further, the dataset is by default ordered according to the values of the first variable. With the argument `sortCol`, we can specify on which variables the data is ordered.

```
> tableplot(diamonds, colNames = c("carat", "price", "cut", "color",
+     "clarity"), sortCol = "price")
```

Setting an appropriate number of row bins (argument `nbins`) is important, like in a histogram. A good number of row bins is a trade of between good polished but meaningless data, and detaild, but noisy data. In practice, we found out that the default number of 100 usually is a good starting point.

2.1 Continuous variables

For each bin of a continuous variable, the mean value is calculated (see Algorithm 1). When the distribution of these mean values is exponential, it is useful to apply a logarithmic transformation (see **?**). The argument `scales` is default set to the auto-detection mode `"auto"`, and can also set to linear mode `"lin"` or logarithmic mode `"log"`.

In Figure **??**, the x-ases of the variables depth and table are broken. The x-axis of a variable i is broken if either

$$0 < max(m_{i1}, m_{i2}, \ldots, m_{in}) \qquad \text{AND}$$
$$\texttt{bias\_brokenX} \cdot max(m_{i1}, m_{i2}, \ldots, m_{in}) < min(m_{i1}, m_{i2}, \ldots, m_{in})$$

OR

$$0 > min(m_{i1}, m_{i2}, \ldots, m_{in}) \qquad \text{AND}$$
$$\texttt{bias\_brokenX} \cdot min(m_{i1}, m_{i2}, \ldots, m_{in}) > max(m_{i1}, m_{i2}, \ldots, m_{in}),$$

where `bias_brokenX` is a bias parameter that should be a number between 0 and 1. If `bias_brokenX=1` then the above conditions are always false, which implies that the x-axes are never broken. On the other hand, if `bias_brokenX=0` then the x-axes are always broken. By default, `bias_brokenX=0.`, which mean that an x-axis is broken if (in case of a variable with positive values) the minimum value is at least 0.8 times the maximum value. In the diamonds dataset, this applies to the variables depth and table.

2.2 Categorical variables

3 Graphical User Interface `tableGUI`

4 Final remarks

Conclusions

A First appendix on the `tabplot` package

B Second appendix