Journal of Statistical Software

May 2013, Volume 53, Issue 7. *http://www.jstatsoft.org/*

New Approaches in Visualization of Categorical Data: R Package extracat

Alexander Pilhöfer
Universität Augsburg

Antony Unwin
Universität Augsburg

Abstract

The R package **extracat** provides two new graphical methods for displaying categorical data extending the concepts of multiple barcharts and parallel coordinates plots. The first method called `rmb` plot uses a crossover of mosaicplots and multiple barcharts to display the frequencies of a data table split up into conditional relative frequencies of one target variable and the absolute frequencies of the corresponding combinations of the remaining explanatory variables. It provides a well-structured representation of the data which is easy to interpret and allows precise comparisons. The graphic can additionally be used as a generalization of spineplots or with barcharts for the conditional relative frequencies. Several options, including ceiling censored zooming, residual shadings and a choice of color palettes, are provided. An interactive version based on the R package **iWidgets** is also presented. The second graphic `cpcp` uses the interactive parallel coordinates plots in the **iplots** package to visualize categorical data. Sequences of points are used to represent each of the variable categories, while ordering algorithms are applied to represent a hierarchical structure in the data and keep the arrangement clear. This interactive graphic is well-suited for exploratory analysis and allows a visual interpretation even for a higher number of variables and a mixture of categorical and numeric scales.

Keywords: categorical data, multiple barcharts, parallel coordinates, R.

1. Introduction

This paper introduces two new graphical approaches in visualization of categorical data and their implementation in the package **extracat** for the R system for statistical computing (R Core Team 2013). The package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=extracat`.

The first graphical display is the `rmb` plot which stands for "relative multiple barcharts". It is a new attempt to enrich the family of mosaicplots by combining the most important

advantages of multiple barcharts (see Hofmann 2000) and classical mosaicplots (see Friendly 1994; Hartigan and Kleiner 1981) in one display. The R package **vcd** (Meyer, Zeileis, and Hornik 2006) provides an implementation of classical mosaicplots and interactive graphics are available through the **iplots** package (Urbanek and Theus 2003). The main intention of `rmb` plots is to precisely display relative frequencies of a target variable for each combination of explanatory variables divided over a grid-like graphical display and, simultaneously, their corresponding weights. The breakup of absolute frequencies into conditional distributions and weights is a common procedure in many methodologies for categorical data analysis, such as generalized linear models or correspondence analysis, but there seems to be a lack of graphical solutions for exploratory as well as illustrative purposes.

The second graphical display provided by the `cpcp` function applies point sequences to the variable categories and uses interactive parallel coordinates plots from the package **iplots** for visualization. These point sequences are ordered to represent a hierarchical structure in the data.

Whilst `rmb` plots aim at a structured and precise graphical representation of categorical data with a target variable, `cpcp` plots are better for exploratory analysis of several variables at the same time. In a graphical data analysis a possible way to make the graphics work together is to explore the data using a `cpcp` plot and to display any findings in an `rmb` plot for a more precise and better structured view.

Section 2 introduces the basic buildup and options of both graphics. Section 3 presents the R package itself and uses real data examples to illustrate the basic usage and options of the plots like the *generalized spineplot*, model visualization (Theus and Lauer 1999), and color palettes based on the R package **colorspace** (Zeileis, Hornik, and Murrell 2009; Ihaka, Murrell, Hornik, Fisher, and Zeileis 2013). An interactive version of the `rmb` plot is also presented.

2. Basic buildup

2.1. `rmb` plots

The mosaicplot family could be described as a collection of graphics which visualize a flat contingency table (see e.g. Meyer *et al.* 2006). Each entry of the table is represented by a rectangle of a size proportional to the corresponding number of observations. The graphics in this family all inherit hierarchical splitting orders in horizontal as well as in vertical directions.

`rmb` plots are a mixture of two members of this family, namely multiple barcharts and classical mosaicplots. W.l.o.g. we will use the case with three variables V_1, V_2 and V_3 for the following explanations: The absolute frequencies n_{ijk} of the frequency table are the number of observations in the i-th, j-th and k-th category of the first, second and third variable respectively. The frequencies are split into conditional relative frequencies $p_{i|jk}$ of one variable and weights corresponding to the other variables according to:

$$n_{ijk} = p_{i|jk} \cdot n_{+jk} = p_{i|jk} \cdot p_{+jk} \cdot n$$

where n is the total number of observations, $n_{+jk} = \sum_i n_{ijk}$ and $p_{+jk} = n_{+jk}/n$. The variable which is represented by the conditional relative frequencies $p_{i|jk}$ will be referred to as the *target variable* in this section. The other variables will be called *explanatory variables* and their combinations are represented by $n_{+jk} = p_{+jk} \cdot n$.

Variable	Description	Levels
`Cont`	Contact to other residents	`"Low"`, `"High"`
`Infl`	Influence on housing conditions	`"Low"`, `"Medium"`, `"High"`
`Type`	Type of residence	`"Tower"`, `"Atrium"`, `"Apartment"`, `"Terrace"`
`Sat`	Satisfaction	`"Low"`, `"Medium"`, `"High"`

Table 1: The Copenhagen `housing` dataset.

In principle, classical mosaicplots (see Friendly 1994; Hartigan and Kleiner 1981) also show both $p_{i|jk}$ and n_{+jk} but while the space is efficiently used, it becomes harder to establish the relation between the rectangles and the corresponding variable combinations with every additional variable. Comparing the proportions of a target category in different combinations of explanatory variables is only possible in a qualitative manner, because the corresponding rectangles neither share a common axis nor have a common scale.

By contrast multiple barcharts and fluctuation diagrams display only the total number of observations n_{ijk} but allocate the information in equal-sized rectangles in a hierarchical grid layout (see Hofmann 2000). The allocation along the grid makes it easier to read the plot and also allows better comparisons especially within the rows or columns because all combinations now share the same x- and y-axis scales. In multiple barcharts the y-axis is set to $[0, \max(n_{ijk})]$ and the x-axis is cut into equal segments for the target categories (or vice versa). Unfortunately comparisons of the conditional distributions of a target variable are quite hard: Comparing absolute frequencies $n_{i|s}$ and $n_{i|t}$ of target category i in two explanatory combinations s and t is obviously not equivalent to the comparison of the relative frequencies $p_{i|s}$ and $p_{i|t}$ and hence it is necessary to use ratios of the form $\frac{n_{i|s}}{n_{j|s}} = \frac{p_{i|s}}{p_{j|s}}$ and $\frac{n_{i|t}}{n_{j|t}} = \frac{p_{i|t}}{p_{j|t}}$ instead.

The basic version of `rmb` plots is constructed as follows: Consider a set of m categorical variables including one target variable. The basis of the plot is a multiple barchart of the $m-1$ explanatory variables displaying the observed frequencies n_{+jk} of their combinations. The plot uses horizontal bars which means that all bars have an equal height and their widths are proportional to the ratios $\frac{n_{+jk}}{\max(n_{+jk})}$.

The conditional distributions of the target categories defined by the probabilities $p_{i|jk}$ are displayed inside these bars. The basic type of visualization is again a barchart with vertical bars. An alternative which is discussed in Section 3 is the *generalized spineplot* version which splits each bar from the basis plot vertically into segments according to their relative frequencies, just as in classical mosaicplots or spineplots. In both versions the x- and y-axis scales are the same, namely $[0, \max(n_{+jk})]$ and $[0, 1]$ respectively.

A first introductory example using the well-known *Copenhagen housing* dataset (c.f. Venables and Ripley 2002) is shown in Figure 1. In R the dataset is available from the **MASS** package and the variables are listed in Table 1.

Figure 1 shows the variables `Cont` and `Infl` on the x-axis, `Type` on the y-axis and `Sat` as the target variable which is by convention on the x-axis. The graphic reveals the weak influence of the `Cont` variable and the strong positive correlation between `Infl` and `Sat`: The differences between the distributions on the left side (low contact) and the corresponding counterparts on the right side (high contact) are quite small and hence the influence of the `Cont` variable on the satisfaction of the respondents is weak. In contrast the variable `Infl` shows a strong positive correlation with the target variable: The people who judged their influence to be low

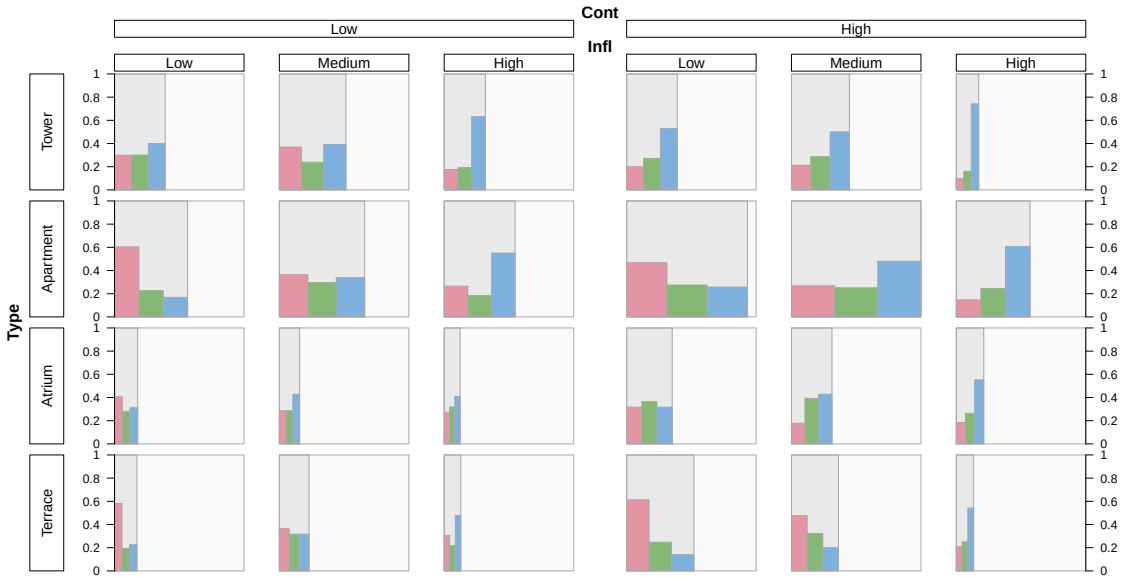

Figure 1: `rmb` plot of `Cont` (x), `Type` (y), `Infl` (x) and `Sat` (x) using the default parameters.

(first and fourth column) tend to be very dissatisfied (left bar) whereas those with a high influence (third and sixth column) are far more content (right bar). A fact which indicates an interaction between the variables `Infl` and `Type` is that this association is weaker for people who live in a tower building (first row) than for others. Due to the equal y-axes scales in each cell a comparison of relative frequencies is possible even for combinations which are not in the same row of the plot. At the same time it is easy to be aware of the cell weights (e.g. the majority of the respondents live in apartments) and the colors make a visual assignment of the target categories simpler. The graphic was created with the following command:

```
R> rmb(formula = ~ Cont + Type + Infl + Sat, data = housing)
```

Further examples can be found in Section 3 and this section will now conclude with a short comparison of `rmb` plots to classical mosaicplots and multiple barcharts.

Figure 2 shows the same example as in Figure 1 displayed as a mosaicplot which was generated using the package **vcd**:

```
R> mosaic(xtabs(Freq ~ Cont + Type + Infl + Sat, data = housing),
+    split_vertical = c(TRUE, FALSE, TRUE, FALSE),
+    labeling = labeling_border(labels = c(TRUE, TRUE, TRUE, FALSE),
+    gp_labels = gpar(cex = 1.5), gp_varnames = gpar(cex = 1.5)),
+    gp = gpar(fill = rainbow_hcl(3)), margins = c(5, 5, 5, 5))
```

It is again possible to see the results we obtained before using the `rmb` plot: The variable `Cont` has hardly any influence and there is a strong relationship between `Infl` and `Sat`. But even though the dataset is well-suited for presentation purposes having no sparse or empty combinations and variables with only a few categories, comparisons between the different proportions are much harder. The conditional relative frequencies which have to be compared in order to judge the differences between people with low contact and those with high contact

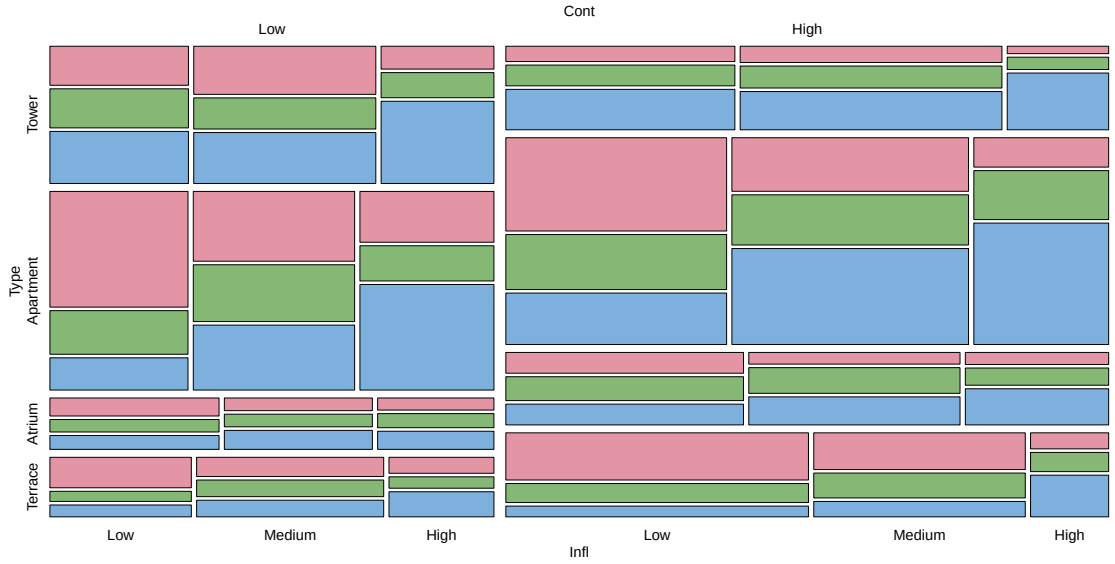

Figure 2: Mosaicplot of `Cont` (x), `Type` (y), `Infl` (x) and `Sat` (y) generated with the package **vcd**.

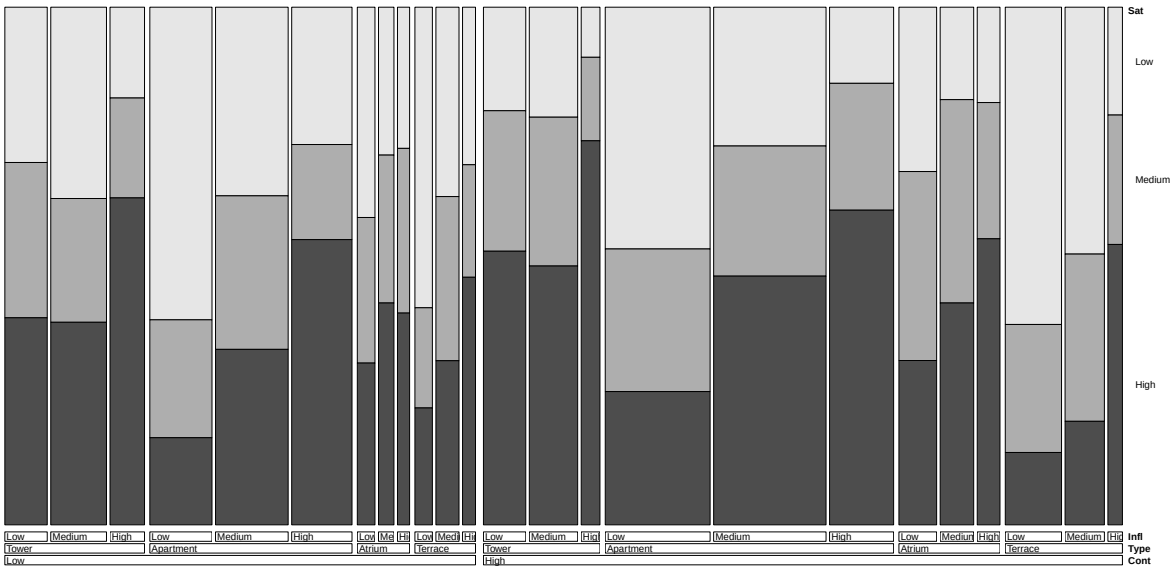

Figure 3: Doubledeckerplot of `Cont` (x), `Type` (y), `Infl` (x) and `Sat` (x) generated with **vcd**.

are proportions of rectangles with different heights and axes. It is even more difficult to evaluate the proportion of any category which is neither the first nor the last one, e.g., the `"Medium"` category of the satisfaction variable in the current example. Although it is possible to change the order and the axes of the variables in order to optimize the display for a specific comparison these problems become harder to deal with when the number of variables and categories increases.

One of the best classical mosaicplot variations for precise comparisons of the conditional

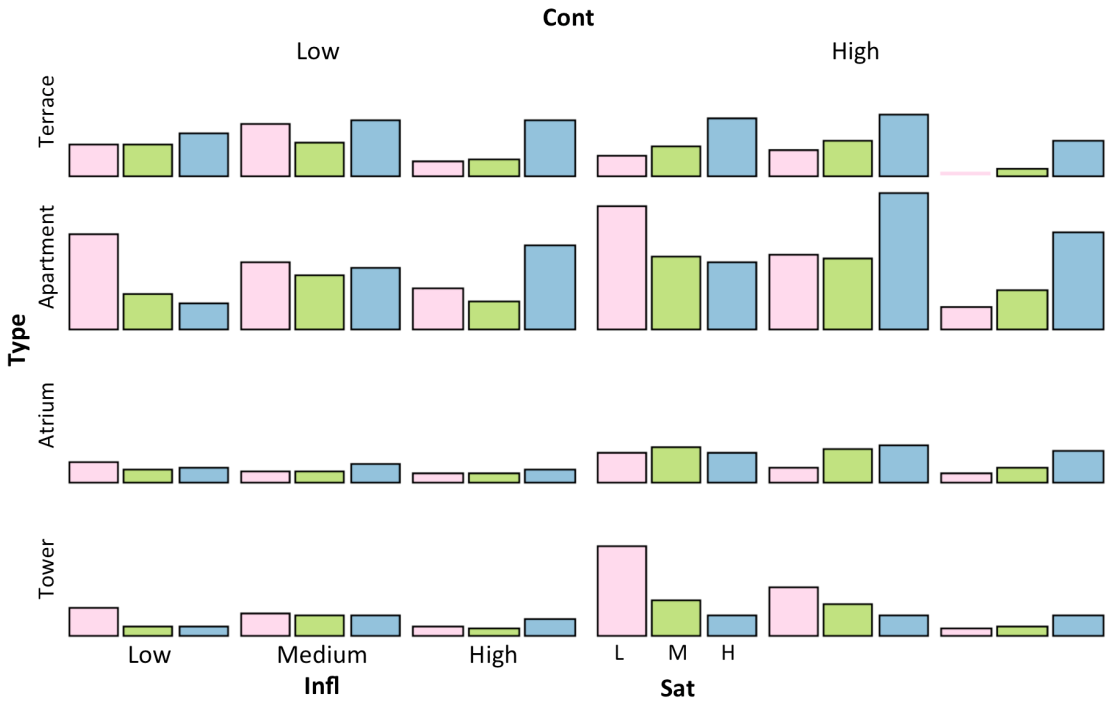

Figure 4: Multiple barchart of `Cont` (x), `Type` (y), `Infl` (x) and `Sat` (x) generated with the **Mondrian** software.

frequencies is called *doubledeckerplots* (Hofmann 2001), which has all explanatory variables on the x-axis. Its interpretability decreases with the number of displayed combinations. For relatively small examples such as the Copenhagen housing data, the graphic is among the best visual representations as Figure 3

```
R> doubledecker(xtabs(Freq ~ Cont + Type + Infl + Sat, data = housing))
```

illustrates: It is easy to compare combinations with different levels of influence or different types of residence but less easy to compare the two different levels of contact.

Figure 4 shows the multiple barchart visualisation of the example discussed in Figures 1 and 2 created with the interactive software **Mondrian** (Theus and Urbanek 2008). The main difference between the multiple barchart and the `rmb` plot is that the multiple barchart displays absolute frequencies whereas the `rmb` plot shows their factorization into conditional relative frequencies and weights. The advantage of this becomes apparent in the last two rows (`"Atrium"` and `"Terrace"`) of the left side of the plot (low contact) where the bars are very small and hardly comparable. Within each combination of `Type`, `Infl` and `Cont` the ratio of any two absolute frequencies is obviously the same as that of the corresponding conditional relative frequencies. Unfortunately this does not hold for two different combinations of these three variables and thus only the ratios of the bars can be compared.

Figure 2 and 4 show that it is at least possible to judge strong differences in the shape of the distributions of a target variable in both classical mosaicplots and multiple barcharts. E.g., the strong positive relationship of `Infl` and `Sat` is apparent in both graphics. Nevertheless in many examples the `rmb` plot provides a better overview and allows for more precise com-

parisons than the other two graphics. `rmb` plots are the preferred choice in two situations: Firstly when the frequencies of combinations of explanatory variables vary a lot. The visual connection between the rectangles and their category labels is hard to make, and the scales for the relative frequencies are very different . Secondly when the conditional relative frequencies $p_{i|s}$ of the target categories, or differences between these values, are small, precise comparisons are only possible with common scales. The `rmb` plot is generally relatively easy to understand and is well-suited for presentation purposes.

The results which have been presented for this example might also be achieved using residual shadings from a logistic regression model. The `rmb` plot also provides this feature, which is illustrated in Section 3.1.

2.2. `cpcp` plots

Although `rmb` plots may provide several advantages in categorical data visualization they are still a member of the mosaicplot family and thus not capable of displaying a large number of variables and categories.

The concept of parallel coordinates plot (PCP, see Unwin, Volinsky, and Winkler 2003) is amongst the most useful graphical solutions with which a relatively high number of variables can be visualized in one display. It was discovered in the late 19th century by Maurice d'Ocagne (see d'Ocagne 1885) and, independently, by Al Inselberg in 1959 (see Inselberg 2009). It is a powerful tool for visualizing multivariate data in one display without dropping information on the raw data values. For exploratory data analysis the adaption of the graphic in an interactive environment (Theus and Urbanek 2008) was another important step in development. Interactive highlighting and the interactive rearrangement and rescaling of variable axes are among the most important features. α-blending can be used to minimize overplotting in larger datasets.

The original concept does not allow for categorical variables, which is a serious disadvantage. Bendix, Kosara, and Hauser (2005) developed an application for categorical variables which has been implemented in the **Parallel Sets** (version 2.1) software (Kosara and Ziemkiewicz 2009). The `cpcp` plot (*categorical parallel coordinates plot*) is a different approach which displays both numeric and categorical variables in the same plot. Is is based on the R package **iplots** and takes advantage of the interactive capabilities of the package.

In order to apply the PCP idea to categorical data it is not sufficient to simply convert the categories into integer values, as this would lead to overplotting hiding most of the important information. To avoid this, within every variable, each category is assigned a sequence of equidistant points with one point for each case and a range proportional to each category's relative frequency. The fact that for any one of these point sequences the corresponding cases are indistinguishable regarding the corresponding variable can be used to make the display clearer and to display additional information. For this purpose the dataset is recursively sorted starting with the last variable and ending with the first one before assigning points to the cases. This procedure leads to a display which shows a hierarchical splitting structure from left to right. The polylines of cases which are identical in the first m variables are drawn together on the corresponding axes and within each such group they will not cross each other. In R for a `data.frame V` with `m factor` variables the sorting process works as follows:

```
R> V <- V[do.call(order, c(V, decreasing = FALSE), ]
```

Variable	Description	Levels
`survived`	Did the passenger survive?	`"No"`, `"Yes"`
`class`	The passenger class	`"1st"`, `"2nd"`, `"3rd"`, `"Crew"`
`gender`	The passenger's gender	`"male"`, `"female"`
`age`	A binary age variable	`"Adult"`, `"Child"`

Table 2: The `titanic` dataset.

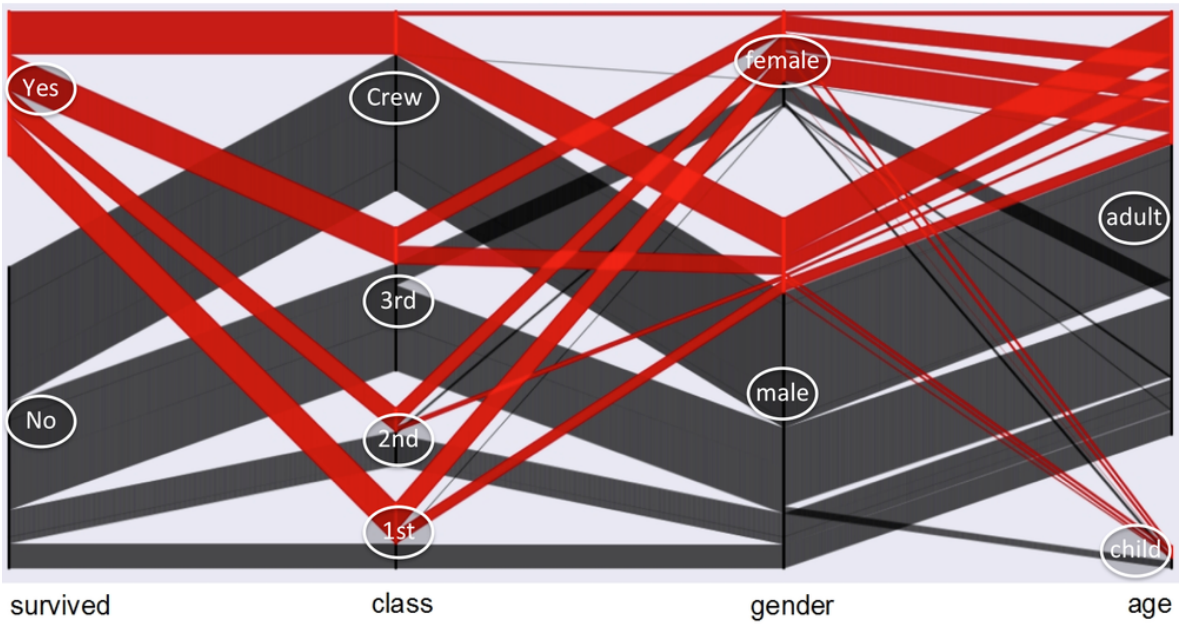

Figure 5: `cpcp` plot of `survived`, `class`, `gender` and `age` using the default parameters.

Figure 5 which was created with

```
R> cpcp(Titanic, ord = c(4, 1, 2, 3))
```

shows the resulting graphic for the well-known `titanic` dataset obtainable from the standard R package **datasets**. It serves as base data for examples in many publications such as Friendly (2000) or Unwin, Theus, and Hofmann (2006). The variables of the dataset are listed in Table 2. In all `cpcp` plots throughout this paper the category labels have been added manually. Graphically it is currently possible to use interactive highlighting and linking to other plots in order to work out the labels for the displayed categories. In future the graphic will also provide interactive queries and automatic label creation for this purpose. A static plot version with more options will also be offered. More information on the interactive implementation of the graphic and use of the underlying**iplots** package can be found in Section 3.3. Before returning to the example it should be mentioned that it is necessary to run **iplots** and therefore `cpcp` from the **JGR** console (Helbig, Theus, and Urbanek 2005) on Unix systems.

In Figure 5 the top category (1) of the first binary variable `survived` is highlighted and α-blending has been used. α-blending is another word for the transparency of the lines. If $\alpha = 0.01$, only a point at which at least 100 lines intersect will be fully saturated. This modification increases the interpretability of the plot and reveals the hierarchical splitting

from left to right better. It will be applied to every `cpcp` example in this paper.

One result from Figure 5 is that in the first two classes (bottom categories) almost every woman survived the catastrophe whereas in the third class about half of the females did not survive. The survival rate for the men is lower by far, especially in the second and third classes. These results have been obtained by a comparison of the widths of the corresponding branches, which represent the frequencies of those groups. For instance in the third class the female branches for survivors and victims have about the same width whereas the victim branch of the males in this class has more than four times the width of the survivor branch. This means that the survival rate is around 50% for the women and below 20% for the men. For a further and more precise analysis of the results obtained from a `cpcp` plot, mosaicplots or `rmb` plots are good choices.

The described basic version of the plot is similar to Parallel Sets (Bendix *et al.* 2005), where lines with identical starting and ending categories are replaced by one polygon of appropriate width. This is a reasonable approach, but is does not allow including continuous data easily, which is a serious disadvantage. The benefit of combining lines in the aforementioned manner dwindles with an increasing number of possible combinations (i.e. variables and/or categories). The advantage for interactive highlighting is that each case or group of cases is selectable without further computations. It is also possible to compute the ribbons from the `cpcp` coordinates easily.

As the small example from Figure 5 reveals, visual clarity decreases with every additional variable and further modifications are necessary to keep plot useful. Additional ordering concepts can be applied, which minimize the number of lines crossing. The first one resorts each point sequence by the rank of the left neighboring variable. Let `ind[[i]][[j]]` denote the indices of the j-th category of the i-th variable, `V[, i]` and `S[ , i]` denote the complete numeric representation of variable `V[ , i]` and `m` be the total number of variables. Then in R the reassignment of the numeric values works as follows:

```
for(i in 2:m) {
  for(j in 1:nlevels(V[ ,i])) {
    ri <- rank(S[ind[[i - 1]][[j]], i])
    S[ind[[i]][[j]], i] <- S[ind[[i]][[j]], i][ri]
  }
}
```

Applying this to the `titanic` example from Figure 5 results in the graphic shown in Figure 6 (top) where the arrangement of the lines between `gender` and `age` is obviously improved. In principle the same sorting can be applied using the ranks of the next variable to the right instead of the left neighboring variable, but as the basic ordering splits up from left to right the suggested procedure is a reasonable choice. The paper will henceforth use sorting by the left neighboring variable. Using the right variable can have advantages, because between each pair of variables the information of the next variable is then anticipated by the ordering, which enhances the view of multidimensional interactions in the graphic.

Between each neighboring pair of variables the number of lines crossing with different categories in the right variable can also be optimized by changing the category orders themselves. Because this procedure is neither applicable to ordinal data nor is it part of the graphical concept itself this paper will not go into this idea any further but ideas in this direction will

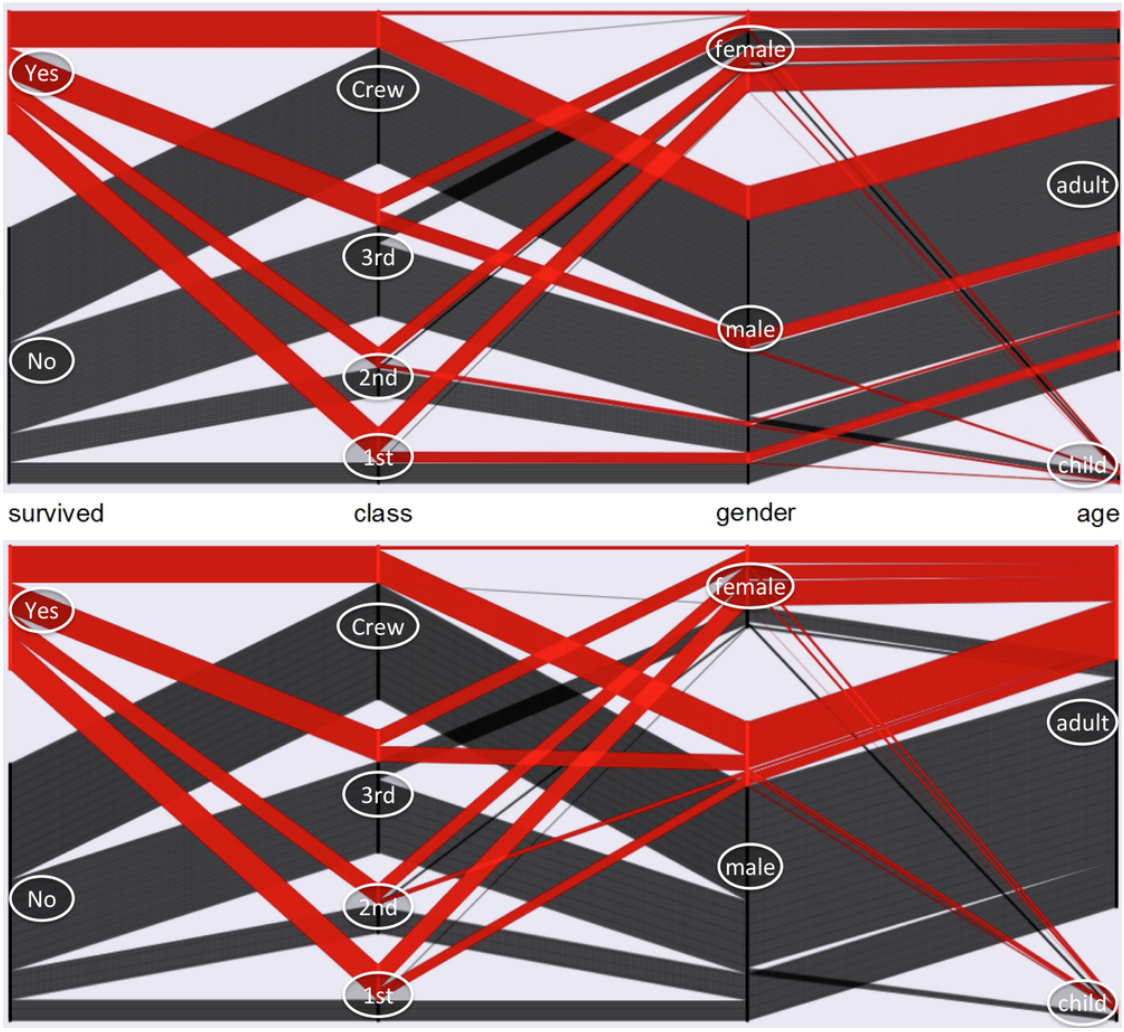

Figure 6: `cpcp` plot of `survived`, `class`, `gender` and `age` using the `sort.individual` option without additional reordering (top) and after reordering by the selection (bottom).

be presented in a future publication. In addition to the minimization of crossings a second ordering procedure has been implemented which is closely related to interactive highlighting. The graphic has the disadvantage that for any highlighting selection the corresponding lines are possibly not drawn together even within the same point sequence (see Figure 6, top). Hence it is not possible to read the corresponding proportions of the selected group within the categories from the graphic. The aim of bringing such data points together can again be achieved by reordering the numeric sequences by the binary variable derived from the selection. Figure 6 (bottom) shows the result of this approach applied to the example at the top of the figure. The graphic was created via

```
R> cpcp(Titanic, ord = c(4, 1, 2, 3), sort.individual = TRUE)
```

and an additional call to `resort()` for the bottom example. Note that if the selection changes again the procedure has to be applied to the original values. More real data examples for this feature and other options can be found in Section 3.3.

3. Implementation in R: Examples, usage and interactivity

This section presents the implementation of the `rmb` plot and the `cpcp` plot in R and introduces further options and variations of the graphics. Every example used in this section is based on a dataset available in **extracat** or another R package and for every variable contained in the examples a short description is given in form of a small table as has been done for the datasets Copenhagen `housing` and `titanic` in Section 2. In `cpcp` plots the categories from bottom to top accord with the category order in these tables. Descriptions for variables contained in one of the datasets but not in the example can be found on the corresponding R-help page.

3.1. The `rmb` function

The first basic example was already shown in Figure 1 in Section 2.1 using the default command

```
R> rmb(formula = ~ Infl + Type + Cont + Sat, data = housing)
```

The `formula` argument controls the order in which the variables will join the plot and `data` is the `data.frame` containing the variables. If `data` contains a frequency variable it should either be called `"Freq"` or be defined as the left hand side of `formula`. The axes of the variables in their given order can be defined by the argument `col.vars` which is either a logical vector where `TRUE` means the variable is split horizontally or an integer vector specifying the indices of the column variables. The default is alternating behavior `col.vars = c(TRUE, FALSE, TRUE, FALSE, ..., TRUE)` and the last (target) variable entry will automatically be set to `TRUE`. Instead of the arguments `formula` and `data` it is also possible to pass a contingency table of class `table` or a frequency table of class `ftable` to the `rmb` function. `ftable` defines the axes for the variables and in this case the `col.vars` argument has no effect. The second example (Figure 7) uses the `col.vars` argument to exchange the variables `Type` and `Infl` in

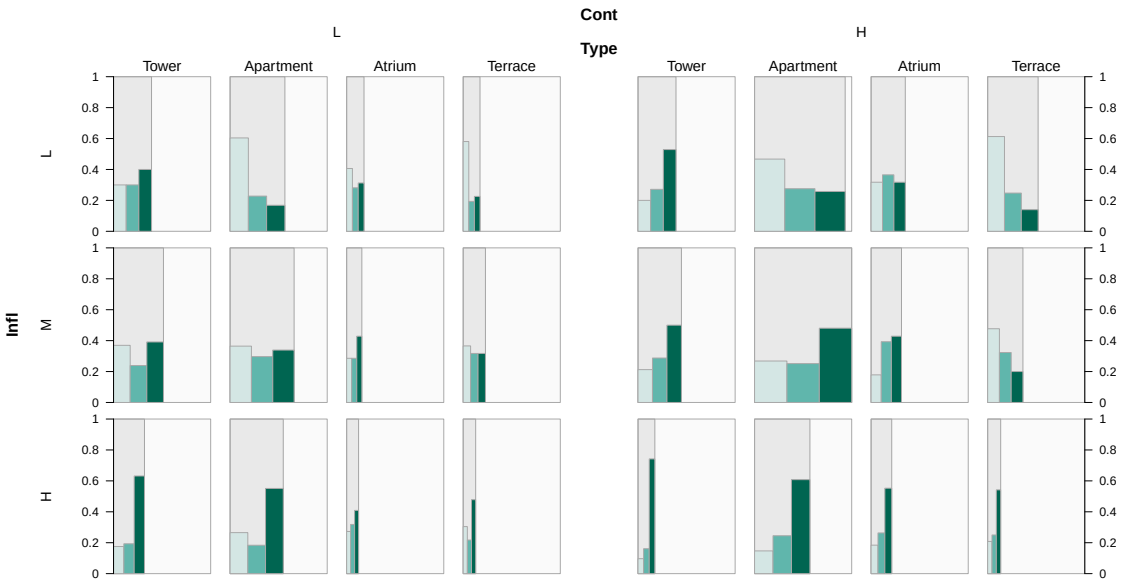

Figure 7: `rmb` plot of `Cont` (x), `Type` (x), `Infl` (y) and `Sat` (x) using custom layout options.

order to examine the differences between the four types of flats. The layout has been changed by setting the following arguments.

`gap.prop`: Total space of the gaps between cells as a proportion of the total width/height.

`gap.mult`: The multiplier for the gap space of different dimensions.

`label`: Whether or not to draw labels.

`label.opt`: An optional list with parameters for the labels.

- `lab.cex`: The font size multiplier for the labels.
- `boxes`: A logical defining whether or not to draw boxes around the labels.
- `abbrev`: An integer vector specifying the number of characters for the abbreviation of the labels.
- `yaxis`: A logical defining whether or not to draw an axis for the proportions.
- `varnames`: A logical defining whether or not to draw the variable names.

`col`: A vector with colors or a key word specifying a color palette.

`col.opt`: An optional list with parameters for the color palette.

The logical arguments `yaxis` and `varnames` can be set to `FALSE` to disable the probability axes and the variable names respectively. `label = FALSE` excludes all labeling in the plot, which is better if space is limited.

The argument `col` is either a vector of custom colors or a keyword which specifies a palette: The default value `"hcl"` stands for hcl-based rainbow colors, `"hsv"` and `"rgb"` stand for hsv-based rainbow colors, `"div"` or `"diverge"` for hcl-based diverging colors and finally `"seq"` or `"sequential"` for hcl-based sequential colors. `"hsv"` and `"rgb"` use the `rainbow` function in the **grDevices** package and the other color vectors are computed using functions from the **colorspace** (Zeileis *et al.* 2009; Ihaka *et al.* 2013) package. Additional arguments can be specified in the `col.opt` argument according to the underlying functions in the **colorspace** package.

The function call which was used to create the graphic in Figure 7 is:

```
R> rmb(formula = ~ Cont + Type + Infl + Sat, data = housing,
+    col.vars = c(1, 2, 4), gap.prop = 0.2, gap.mult = 4,
+    col = "seq", col.opt = list(h = 180, c = c(90, 10)),
+    label.opt = list(lab.cex = 1.5, boxes = FALSE, abbrev = c(1, 18, 1, 1)))
```

One result from the graphic is that the satisfaction of people who judged their influence on the housing conditions to be low (first row) is very different for the `"Tower"` and `"Apartment"` types of residence.

The next example shows the first fundamental option set by the `eqwidth` parameter using the `carcustomers` dataset from 1983 (Department of Statistics, University of Munich 1983) which is available in the R package **extracat**. The variables used here are listed in Table 3.

Figure 8 shows the target variable `sat` given the combinations of `premod` and `model` in the *equal-width-mode* of an `rmb` plot and was created with

Variable	Description	Levels
`model`	The car model the customer purchased	`"A"`, ..., `"D"`
`premod`	The origin/model the customer had before	`"Audi"`, ..., `"Volkswagen"`
`sat`	Satisfaction with the new car	`1` (fully satisf.), ..., `5` (not satisf.)

Table 3: Three variables from the `carcustomers` dataset.

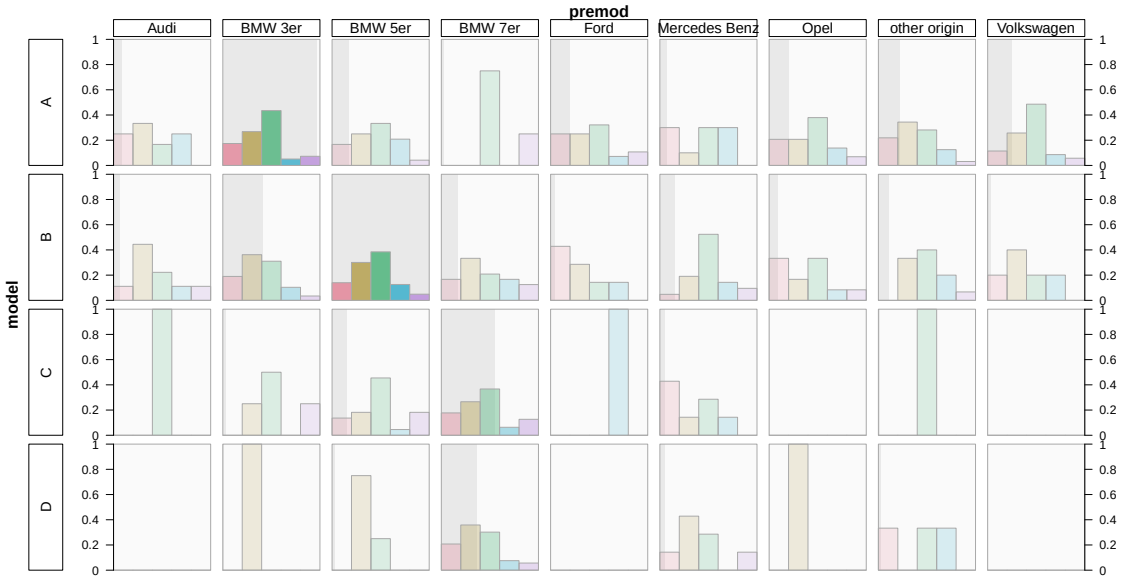

Figure 8: `rmb` plot of `model` (x), `premod` (y) and `sat` (x) using the `eqwidth` option: the barcharts use the full cell width and the shaded rectangles in the background represent the observed frequencies.

```
R> rmb(formula = ~ premod + model + sat, data = carcustomers,
+     eqwidth = TRUE, gap.prop = 0.1, gap.mult = 4,
+     label.opt = list(lab.cex = 1.5))
```

The equal width option uncouples the width of the barcharts for the relative frequencies from the corresponding horizontal bars for the weights so that the full cell width is used, which increases the space efficiency. α-blending corresponding to the cell weights is applied in order to allow a quick visual judgement. The multiple barchart for the weights (shaded rectangles) remains in the background. This option is especially recommended when the display contains a large number of cells. The graphic shows clearly that many of the customers had a BMW before they bought a new model. That and the fact that previous BMW ownership is split up into three particular models makes it reasonable to suppose that the new models are also BMW cars in ascending classes. Assuming that the models `"A"`,`"B"` and `"C"` are new versions of the `"BMW 3,5,7"` series the graphic shows that people who changed from a `"BMW 3"` to a `"BMW 5"` series car are a bit more satisfied than those who kept faith with the `"BMW 3"` series. This shows up in less weight on the third category and more weight on the second one. Remember that German ratings range from good (one) to bad (five). Although the small size of the dataset makes it hard to obtain reliable results beyond these main categories, the graphic provides a good overview of the whole situation. The corresponding classical

Variable	Description	Levels
`Europe`	Scale for respondents' attitudes toward European integration	`1` (positive), ..., `11` (eurosceptic)
`political.knowledge`	Knowledge of parties' positions on European integration	`0` (low), ..., `3` (high)
`vote`	Party choice	`"Conservative"`, `"Labour"`, `"Liberal Democrat"`

Table 4: The `BEPS` dataset from the R package **effects**.

mosaicplot which can be found in Figure 15 in the appendix is less easy to interpret.

Another important option can be activated by setting the `spine` argument to `TRUE`: Instead of barcharts a spineplot will be drawn in each cell of the plot. This version of the plot is called a *generalized spineplot* and is recommended when the number of cells increases and the target variable has few categories. In addition it is possible to choose which target categories will be shown and their order by setting the `cat.ord` parameter. For instance `cat.ord = c(2,1,4)` will arrange the second target category at the bottom of each cell then stack the first and fourth category and leave out the third. Setting the `cat.ord` argument in the barchart version works similarly: All cases with a target category that is not included in the `cat.ord` argument will be left out of the plot and the graphic is conditioned on the remaining cases.

Figure 9 shows the `rmb` plot of the *British Election Panel Study* dataset from 1997–2001 in the generalized spineplot version. This dataset is called `BEPS` and can be found in the R package **effects** (Fox 2003). The variables used in the plot are listed in Table 4. The parameter `freq.trans = "sqrt"` causes the plot in Figure 9 to use the square-root transformed absolute frequencies of the combinations of the explanatory variables so that the visual interpretability of sparse combinations is improved. Setting it to `c("sqrt",k)` or `"log"` will lead to k-th root and log transformations respectively.

For the purpose of judging any kind of model it is reasonable to compare it to the actual data whenever possible. In this spirit it is interesting to compare Figure 9 which was created using

```
R> rmb(formula = ~ political.knowledge + Europe  + vote, data = BEPS,
+    col.vars = c(1, 2, 3), spine = TRUE, yaxis = FALSE, gap.mult = 100,
+    gap.prop = 0.1, freq.trans = "sqrt", col = c("blue", "red", "orange"),
+    label.opt = list(lab.cex = 1.5))
```

with the (stacked) effects plot taken from Fox and Hong (2009) which is shown in Figure 13 in the appendix. This plot is a model visualization rather than a data visualization and it uses a B-spline with three degrees of freedom instead of the original `Europe` variable. Both plots use the x-axis for both variables `political.knowledge` and `Europe`.

Two observations are that the second value (category 1) for `political.knowledge` is very sparse and the `0`-respondents seem to have voted a bit randomly, both facts the effects plot does not reveal to the user. Focussing on the third (2) category the uppermost liberal democrat party seems overvalued by the model compared to the real data. Taking into account that in the model the variable `Europe` is just a B-spline with three degrees of freedom it is not surprising that the local maximum of the conservative party at `Europe == 8` has been smoothed out in the last category (3) of `political.knowledge`. These observations are not

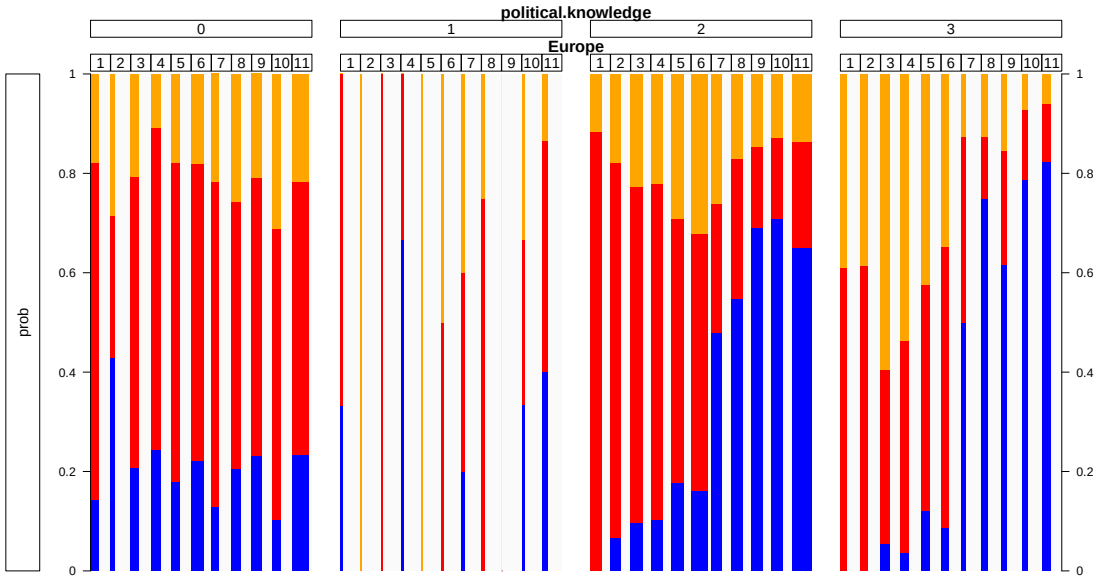

Figure 9: `rmb` plot of `political.knowledge` (x), `Europe` (x) and `vote` (x) using the options `spine` and `freq.trans`.

a criticism of effects plots but a suggestion for comparing real data plots with model based plots in order to obtain additional information about the data, the model fit, and where they differ.

Another type of model which is often linked to plots from the mosaicplot family is *log-linear models* (Theus and Lauer 1999). The usual way of doing so is through residual shadings which are also available for `rmb` plots. The argument `expected` is either `NULL` or a list containing index vectors defining the interaction terms of a model the same way as in the **vcd** package. The interaction terms depend on the model chosen by `mod.type` which currently accepts the values `"polr"` and `"poisson"` for *proportional odds logistic regression* (see e.g. Tutz 2011, p.243–246) and *log-linear Poisson models* respectively. Every multinomial logistic regression model has a Poisson equivalent and hence Poisson models are the more general solution here. For the Poisson model the function `residuals` computes the residuals from the model object. Possible types of residuals are `"deviance"`, `"pearson"`, `"working"`, `"response"` as well as `"partial"` and can be selected by setting the argument `resid.type` to one of those values.

Variable	Description	Levels
`poverty`	Government commitment in poverty reduction	`"Too Little"`, `"About Right"`, `"Too Much"`
`religion`	Member of a religion	`"no"`, `"yes"`
`degree`	University degree	`"no"`,`"yes"`
`country`	Where the respondent comes from	`"Australia"`, `"Norway"`, `"Sweden"`, `"USA"`
`age`	Respondent's age in years	`[18,25)`, ..., `[65,93)`
`gender`	Respondent's gender	`"female"`, `"male"`

Table 5: The `WVS` dataset available in the R package **effects**.

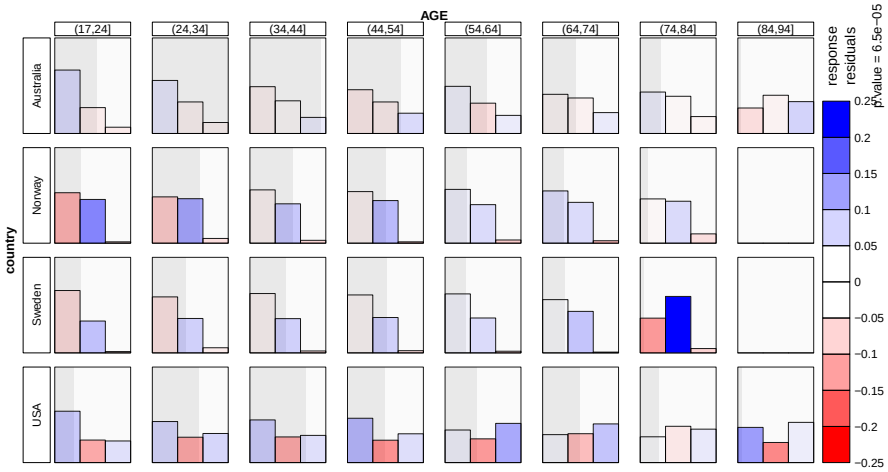

Figure 10: `rmb` plot of `age` (x), `country` (y) and `poverty` (x) using residual shadings corresponding to the proportional odds model `poverty`∼`age + country` in the `eqwidth` mode.

For the proportional odds model only `"response"` is implemented so far but other options will be added in the future.

The next example is again taken from Fox and Hong (2009) and also related to a model. The dataset is from the World Values Surveys 1995–1997 for Australia, Norway, Sweden and the United States and contains an ordinal target variable `poverty`. Both ordinal logistic regression models like the proportional odds logistic regression which is available through the R function `polr` in the **MASS** package or the more general multinomial logistic regression can be applied to the data. The variables of the dataset are listed in Table 5. The `rmb` plot with residual shadings corresponding to the response residuals of the proportional odds model `poverty ~ age + country` can be obtained with the command

```
R> rmb(formula = ~ AGE + country + poverty, data = WVS,
+    col.vars = c(1, 3), eqwidth = TRUE, expected = list(1, 2),
+    label.opt = list(lab.cex = 1.5, yaxis = FALSE),
+    model.opt = list(mod.type = "polr", resid.type = "response"))
```

and is shown in Figure 10.

The `eqwidth` option has been enabled to improve the interpretability of the display in absence of the helpful target category coloring. The residual shading shows for instance that the midlevel target category `"About right"` is overestimated in the USA (red shading) and underestimated in Norway and Sweden (blue shading). The comparative graphic taken from Fox and Hong (2009) can be found in Figure 14 in the appendix.

3.2. Interactive `rmb` plots

In the previous section a variety of options for `rmb` plots were presented. Some of them have a huge influence on the resulting graphic and its usefulness. The main choice is between the standard barchart and the generalized spineplot versions, but horizontal and vertical zooming are important features too. As always with a member of the mosaicplot family, the selection

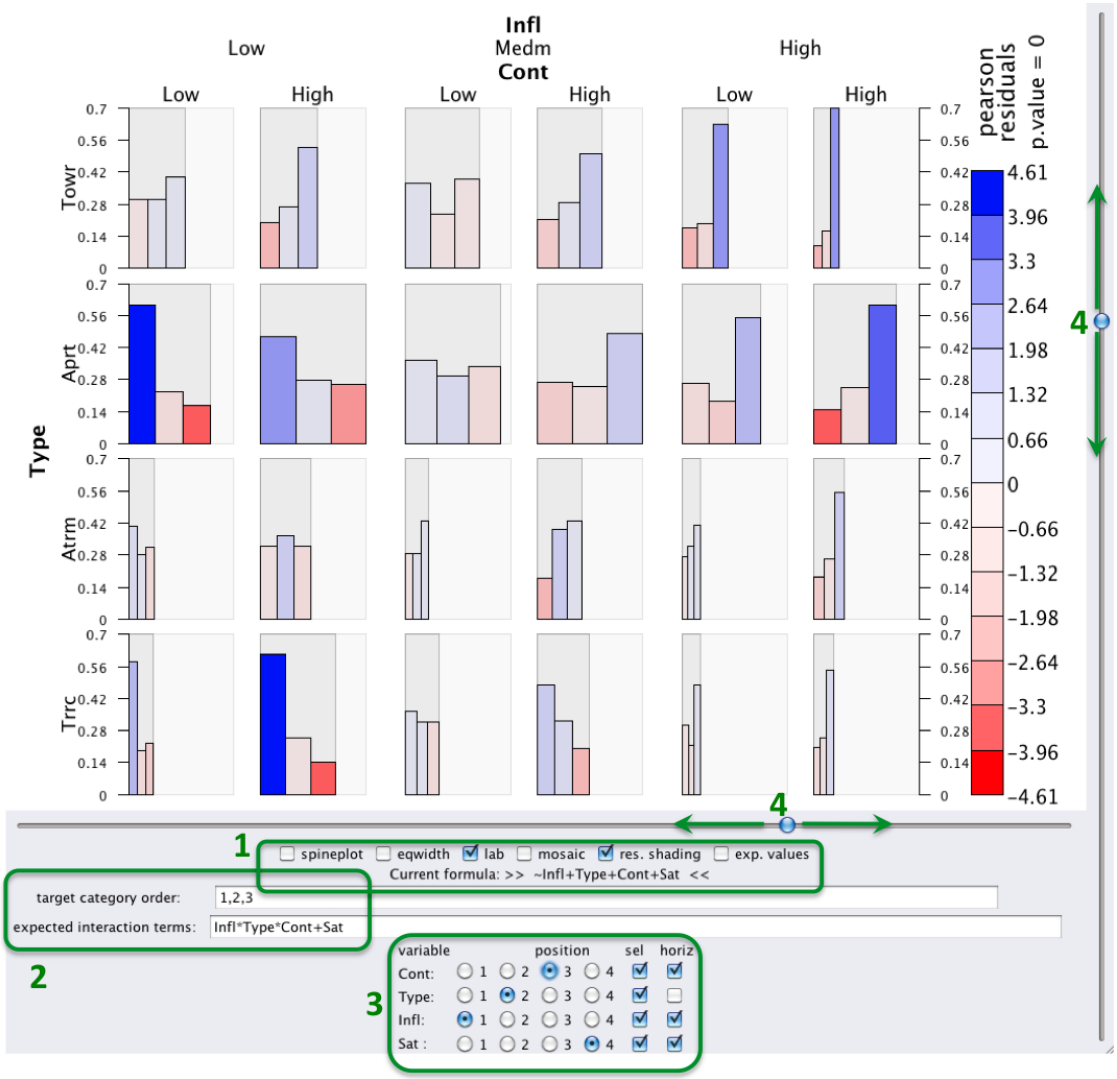

Figure 11: `irmb` plot of the Copenhagen housing dataset as in Figure 7. Zooming has been applied to both axes and a residual shading replaces the colors. The green marked sets of controls are explained in Table 6.

of variables, the order of variables and finally the axes to which the variables are assigned matter a lot. Additionally it is possible to integrate information from statistical models in the graphic. There is no uniformly ideal default combination of settings and the user has to find his or her way through the options in order to determine what suits their particular problem best. This section presents an interactive version of `rmb` plots which facilitates working with this type of graphic. The function is called `irmb` and was developed using the R packages **iWidgets** (Urbanek 2007) and **JGR** (Helbig *et al.* 2005). It is currently not publicly available in the **extracat** package.

The `irmb` function provides a variety of interactive features which are listed in Table 6. The first column (`*` = `1`, ..., `4`) corresponds to the marker numbers of the controls in Figure 11 which was created with

*	Label	Feature
1	`Spineplot`	Changes between barcharts and spineplots
1	`Eqwidth`	Toggles the `eqwidth` mode
1	`Lab`	Switches the labeling on and off
1	`Mosaic`	Changes between `rmb` and mosaicplot drawn via **vcd**
1	`Res.shading`	Enables/disables residual shading
1	`Exp.values`	Uses expected values instead of the observed ones
2	`Expected interaction terms`	Enters a model formula for the residual shadings
2	`Target category order`	Chooses the target category order
3	`Position`	Changes the variable order via the radio button field
3	`Sel`	Includes or excludes variables
3	`Horiz`	Changes between horizontal and vertical axis
4		Sliders for ceiling censored zooming on x- and y-axes

Table 6: The interactive features of the `irmb` plot through the corresponding controls marked in Figure 11.

```
R> irmb(formula = ~ Cont + Type + Infl + Sat, data = housing,
+    abbrev = 4, lab.cex = 1.5, boxes = FALSE)
```

The example shown in Figure 11 is basically the same as the static example from Figure 7. It uses residual shadings according to the logit independence model

```
Freq ~ Sat + Type * Infl * Cont
```

instead of the colors, and zooming on the x- and y-axis has been applied. The order of the variables `Infl` and `Cont` was changed using the radio button field. Except for the mosaicplot option which is based on the R package **vcd**, all the interactive options are also available in the basic `rmb` function. The most significant (pearson) residuals occur in the second and fourth row and there is a clear difference between the first two columns (low influence) and the last ones (high influence) for all types of residence but the first. This again indicates an interaction between the variables `Infl` and `Type`. With the interactive controls it is easy to exclude the less important variable `Cont` by clicking on the corresponding checkbox and to add model parameters like `Infl:Sat` or `Type:Sat` to the model. Exchanging the variable orders, plotting the expected values or zooming in can then yield further insights.

3.3. The `cpcp` function

For the `cpcp` plot two basic examples have already been given in Figure 5 and Figure 6 in Section 2.2 but without an explicit explanation of the corresponding parameters. These are:

`V`: The dataset in form of a `matrix` or a `data.frame`.

`ord`: An integer vector containing the ordered indices of the variables to plot.

`freqvar`: The (optional) name of the frequency variable.

`numerics`: An integer vector containing the indices of variables which are to be handled as numeric variables.

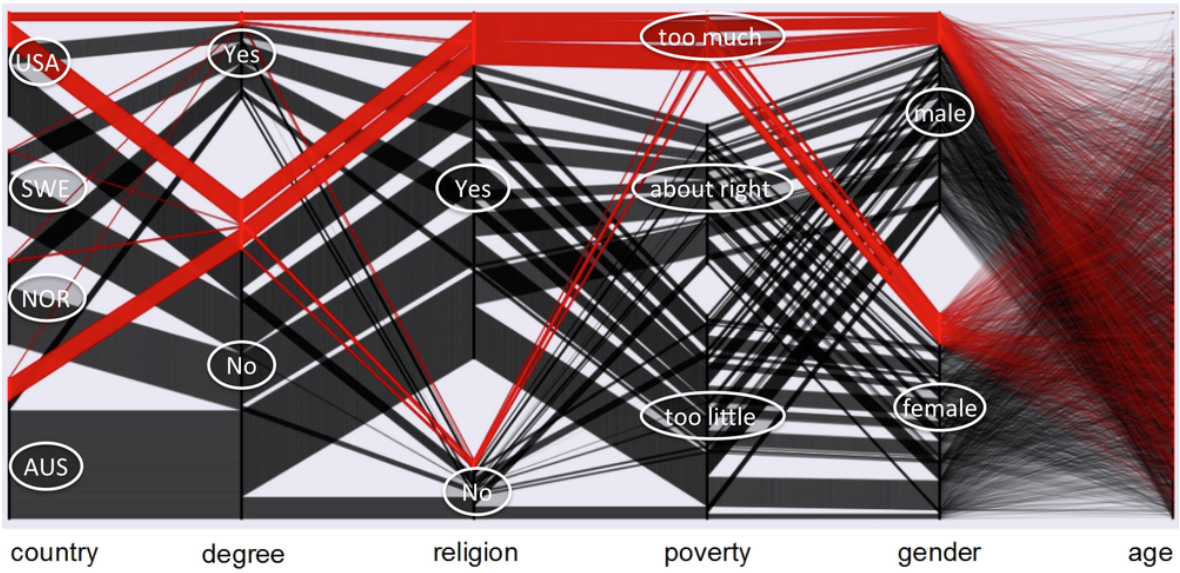

Figure 12: `cpcp` plot of the `WVS` dataset with highlighting of all respondents who judged their government's commitment in poverty reduction as `"too much"`.

In the first example in Figure 5 the variable order was changed using the `ord` parameter whereas `freqvar` and `numerics` remained unchanged. The latter choice is clear because there are simply no numeric variables in the dataset, but a look at its summary in R reveals the fact that it contains a frequency variable. The secret lies in the labeling of the variable: `"Freq"` is the labeling which is used by `ftable` and will be indicated automatically if `freqvar` is undefined. Note that it is not necessary to specify the `numerics` argument for variables containing real numbers because they will also be handled automatically. In contrast variables of class integer are treated as factors.

The second plot in Figure 6 varies from the first one only in the choice of the additional argument `sort.individual = TRUE` which is described in Section 2.2. We return to the `WVS` dataset from Figure 10 and proceed with the additional sorting algorithm for highlighted groups. Figure 12 shows the `cpcp` plot of the `WVS` dataset after applying the second resort-algorithm for all untransformed variables in the modified order `country`, `degree`, `religion`, `poverty`, `gender` and `age` with a highlighting of all respondents who judged their government's commitment in poverty reduction as `"too much"`. The graphic was created using the commands

```
R> cpcp(WVS, ord = c(4, 3, 2, 1, 6, 5), numerics = 5, jitter = TRUE)
R> s <- iset()
R> iset.select(what = which(ivar.data(s$poverty) == "Too Much"))
R> resort()
```

Regarding the first two variables `country` and `degree` we obtain two interesting results. First there are very few respondents in our selected group who came from Scandinavia (second and third level) and second nearly all group members who hold a university degree come from the USA (top) whereas those without a degree come from USA and Australia (bottom) in approximately equal parts. The same result could have been seen in the plot with the

default variable order without any additional sorting. Then the selection would have been a connected group in the first variable (as is the case in the first `cpcp` example in Figure 5) and hence the hierarchical structure of the plot would have kept the selected cases together automatically. As the graphic shows the second resort algorithm provides interpretable results for any variable order.

The underlying `ipcp` function as well as other software like the Parallel Sets software (Kosara and Ziemkiewicz 2009) offer an interactive change of the variable order via drag-and-drop. In `cpcp` this option is not yet available but will be added in the future.

After a selection has been made it is always possible to call the function `resort` in order to arrange the highlighted cases at the top of each point sequence. Selecting either all cases or none will restore the original order. An example for this feature is given in Figure 6 (bottom). This functionality is also available through the function `listen` which waits for the selection to change and then automatically calls `resort`. The underlying **iplots** functionalities are `ievent.wait` and `iset.sel.changed`.

One of the most important aspects of interactive highlighting is the parallel use of several plots at the same time. The `cpcp` function constructs a `data.frame` of class `iset` which contains the original variables, the numeric variables derived from the procedure as well as some auxiliary variables. Thus it is possible to create other `iplots` based on this `iset` which are linked to the `cpcp` plot. `iplots` which have been created before drawing the `cpcp` plot are not linked to it because the `cpcp` plot creates its own new `iset` with the observations in a changed order.

While the original variables bear their original name the labels of the auxiliary variables start with one of the capital letters `"S."`,`"I."` or `"C."` and contain the centered point **S**equences, the **I**nteger values and the final numeric **C**oordinates which are used for the plot. The original variables can be addressed via `s <- iset()` and referred to using their name or index. For example consider the following commands:

```
R> cpcp(WVS, ord = c(1, 2, 3))
R> s <- iset()
R> names(s)
```

```
 [1] "poverty"    "religion"   "degree"     "country"    "age"
 [6] "gender"     "C.poverty"  "C.religion" "C.degree"   "S.poverty"
[11] "S.religion" "S.degree"   "I.poverty"  "I.religion" "I.degree"
```

```
R> ibar(s[[4]])
R> ibox(s$age)
```

The input commands first produce a `cpcp` display for the first three variables. After selecting the corresponding `iset s` and printing its name vector a barchart for `country` as well as a boxplot for the integer variable `age` are plotted. Additional options are:

`gap.type`**:** The rule for the gaps between categories.

`gap.space`**:** The (maximum) total proportion of the gaps.

`spread`**:** The spread multiplier if `gap.type == "spread"`.

`jitter`**:** If `TRUE` integer variables defined by `numerics` will be jittered using `runif`.

The first option `gap.type` allows the choice of one of three different rules for the gaps between the categories. The default value is `"equal.tot"` which makes the gaps add up to a proportion of the total height defined by the argument `gap.space`. This is in most cases the preferred choice because it keeps proportions on different axes comparable. In this case the numeric sequences for each variable `V[, j]` are computed as follows:

```
p <- table(V[, j])
N <- sum(p)
p <- p/N
cp <- c(0, cumsum(p)) * (1 - gap.prop)
k <- length(p)
gap <- gap.prop/(k - 1)
seqs <- list()
for(i in 1:k) {
  seqs[[i]] <- seq(p[i], p[i + 1], (p[i + 1] - p[i])
  seqs[[i]] <- seqs[[i]]/(p[i] * (N - 1)) + (i - 1) * gap
}
```

If `gap.type` is set to `"equal.gaps"` then the gaps will be equal in all variables and add up to a total proportion `gap.space` in the categorical variables with the highest number of categories. The last choice for this parameter is `"spread"` which arranges the categories in a way that their central points have equal distances. The parameter `spread` then defines how widely the cases are spread around these central points. That means for each variable the width for category `j` is proportional to `spread * p[j] / max(p)` where `p` is the vector of relative frequencies for this variable as above.

Last but not least the logical `jitter` can be set to `TRUE` in order to add random numbers from a uniform distribution to integer variables which have been defined as `numerics`. This can reduce overplotting and hence improve the interpretability of the affected variables.

The `cpcp` plot and especially the resort algorithm will work for datasets of up to several thousand cases depending on the system capabilities. The most timeconsuming part of the procedure is the updating process of the `ipcp` plot itself which takes much longer than the computation itself. In a static version of the plot it is more efficient to combine lines to ribbons.

4. Conclusion

This paper has introduced two extensions of well-known graphics for the visualization of categorical data. The `rmb` plot is a member of the mosaicplot family which displays the natural factorization of absolute frequencies into conditional relative frequencies and their weights. This makes it especially useful for the analysis of target variables. Zooming and the equal-width option are key features for displaying small frequencies. Residual shadings are used with log-linear and logistic models and the option to use `rmb` plots as a generalization of spineplots further increases the flexibility of the graphic. Several layout options complete the implementation in R. The interactive version described in this work offers controls for the most important options and alternatives such as the equal width mode or residual shadings as well as an additional classical mosaicplot based on the **vcd** package.

In contrast the `cpcp` plot is an attempt to increase the number of displayable categorical variables using the well-established parallel coordinates plot as its basis. Its strength lies in interactive features like highlighting and the resort-algorithms which make it a powerful tool for exploratory data analysis. Its capability of displaying a mixture of categorical and continuous variables gives it an advantage over alternative plots.

One possible way of combining the graphics in a graphical analysis of categorical data is the following: A `cpcp` plot is used for interactive exploration of the dataset and `rmb` plots are then used to display any specific findings in the data more precisely.

In future it is intended to add more methods and tools for the analysis of categorical data to the package and more options for the plots will be offered.

References

Bendix F, Kosara R, Hauser H (2005). "**Parallel Sets**: Visual Analysis of Categorical Data." In *Proceedings of the 2005 IEEE Symposium on Information Visualization (InfoVis)*, pp. 133–140.

Department of Statistics, University of Munich (1983). "Consumer Satisfaction of Car Owners." Accessed 2010-07-19, URL http://www.stat.uni-muenchen.de/service/datenarchiv/auto/auto_e.html.

d'Ocagne M (1885). *Coordonnées Parallèles et Axiales: Méthode de Transformation Géométrique et Procédé Nouveau de Calcul Graphique dÉduits de la Considération des Coordonnées Parallèlles*. Gauthier-Villars, Paris.

Fox J (2003). "Effect Displays in R for Generalised Linear Models." *Journal of Statistical Software*, **8**(15), 1–27. URL http://www.jstatsoft.org/v08/i15/.

Fox J, Hong J (2009). "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the **effects** Package." *Journal of Statistical Software*, **32**(1), 1–24. URL http://www.jstatsoft.org/v32/i01/.

Friendly M (1994). "Mosaic Displays for Multi-Way Contingency Tables." *Journal of the American Statistical Association*, **89**, 190–200.

Friendly M (2000). *Visualizing Categorical Data*. SAS Insitute, Carey.

Hartigan JA, Kleiner B (1981). "Mosaics for Contingency Tables." In WF Eddy (ed.), *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, pp. 268–273. Springer-Verlag, New York.

Helbig M, Theus M, Urbanek S (2005). "**JGR**: A Java GUI for R." *The Computing and Graphics Newsletter*, **16**(2), 9–12.

Hofmann H (2000). "Exploring Categorical Data: Interactive Mosaic Plots." *Metrika*, **51**(1), 11–26.

Hofmann H (2001). "Generalized Odds Ratios for Visual Modeling." *Journal of Computational and Graphical Statistics*, **10**(4), 628–640.

Ihaka R, Murrell P, Hornik K, Fisher JC, Zeileis A (2013). ***colorspace**: Color Space Manipulation*. R package version 1.2-2, URL http://CRAN.R-project.org/package=colorspace.

Inselberg A (2009). *Parallel Coordinates*. Springer-Verlag, New York.

Kosara R, Ziemkiewicz C (2009). "**Parallel Sets** v2.1: Categorical Data Visualization." URL http://eagereyes.org/parallel-sets.

Meyer D, Zeileis A, Hornik K (2006). "The Strucplot Framework: Visualizing Multi-Way Contingency Tables with **vcd**." *Journal of Statistical Software*, **17**(3), 1–48. URL http://www.jstatsoft.org/v17/i03/.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Theus M, Lauer SRW (1999). "Visualizing Loglinear Models." *Journal of Computational and Graphical Statistics*, **8**(3), 396–412.

Theus M, Urbanek S (2008). *Interactive Graphics for Data Analysis: Principles and Examples*. Chapman & Hall/CRC.

Tutz G (2011). *Regression for Categorical Data*. Cambridge University Press, Cambridge.

Unwin A, Theus M, Hofmann H (2006). *Graphics of Large Datasets: Visualizing a Million*. Springer-Verlag, New York.

Unwin A, Volinsky C, Winkler S (2003). "Parallel Coordinates for Exploratory Modelling Analysis." *Computational Statistics & Data Analysis*, **43**(4), 553–564.

Urbanek S (2007). "**iWidgets** – Basic GUI Widgets for R." Accessed 2010-07-21, URL http://www.rforge.net/iWidgets/.

Urbanek S, Theus M (2003). "**iPlots** – High Interaction Graphics for R." In K Hornik, F Leisch, A Zeileis (eds.), *Proceedings of the 3rd International Workshop on Distributed Statistical Computing 2003*. Technische Universität Wien, Vienna, Austria. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Zeileis A, Hornik K, Murrell P (2009). "Escaping RGBland: Selecting Colors for Statistical Graphics." *Computational Statistics & Data Analysis*, **53**, 3259–3270.

A. Additional graphics

Here, some alternative visualizations are provided. Figures 13 and 14 are created by:

```
R> plot(europe.knowledge, style = "stacked",
+    colors = c("blue", "red", "orange"), rug = FALSE)
R> plot(effect("country*bs(age,4)", wvs.2, xlevels = list(age = 18:83),
+    given.values = c(gendermale = 0.5)), rug = FALSE, style = "stacked")
```

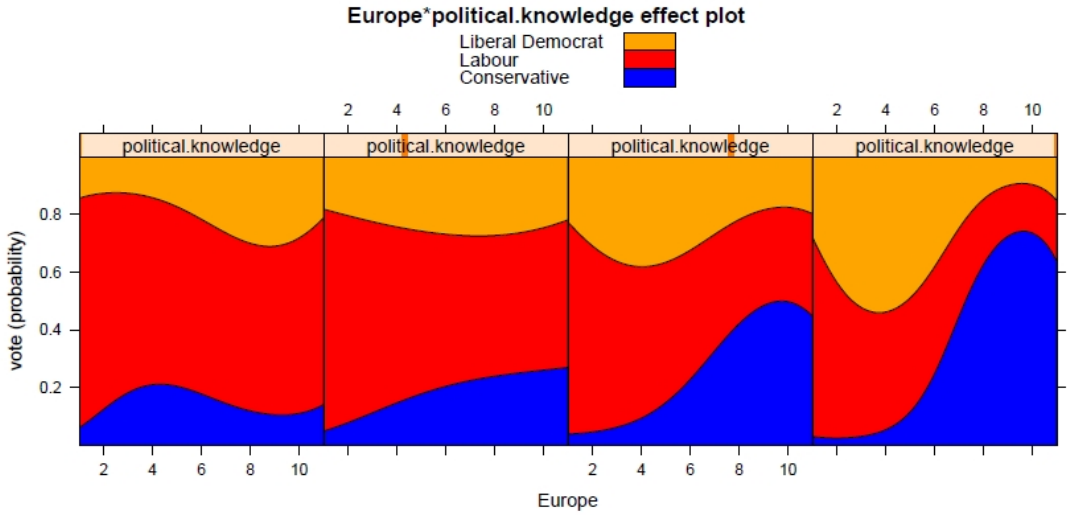

Figure 13: Alternative "stacked-area" effect display for the Europe × political.knowledge interaction. Taken from Fox and Hong (2009, p. 14).

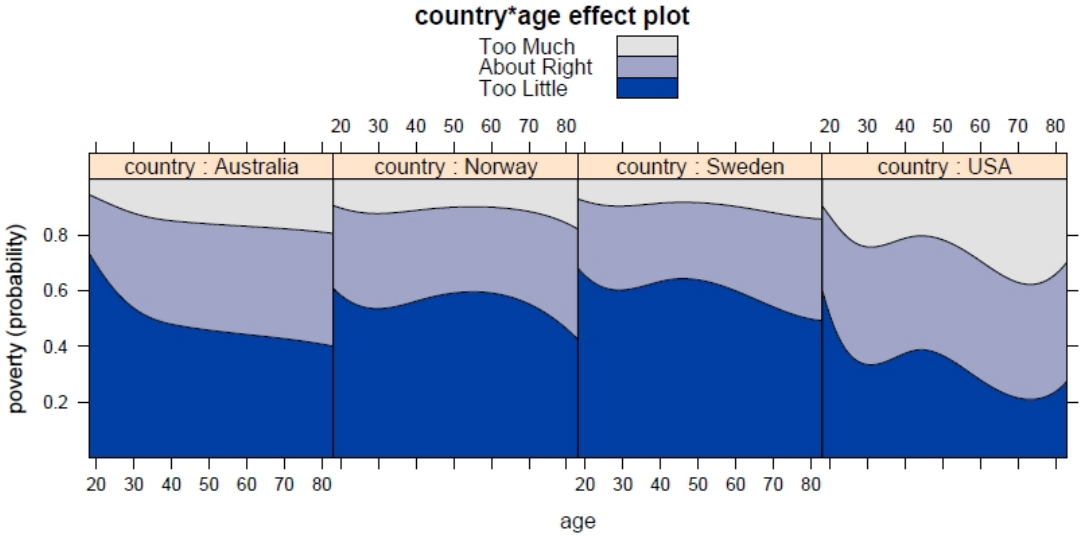

Figure 14: Alternative "stacked-area" effect display for the country × age interaction taken from Fox and Hong (2009, p. 19).

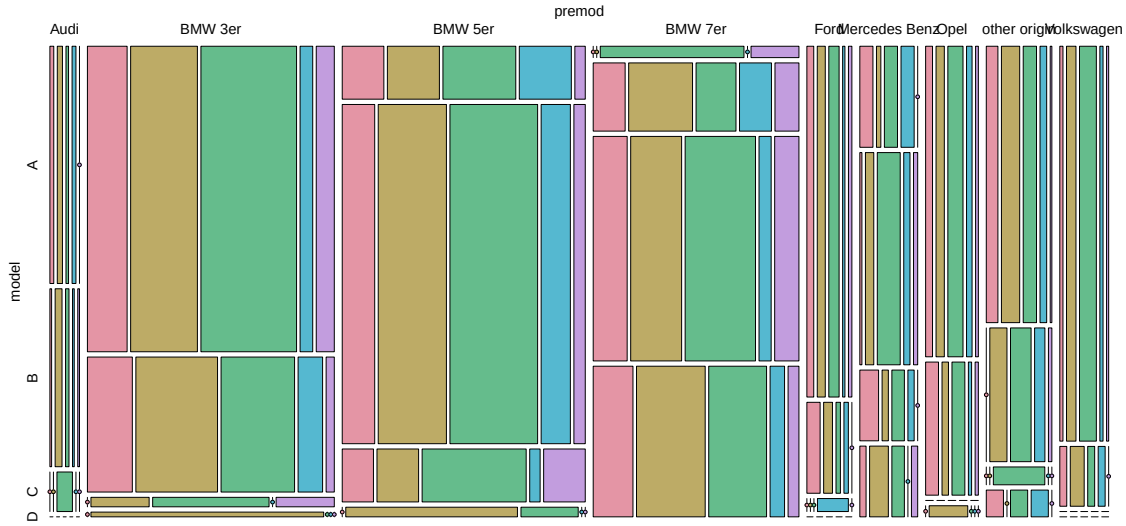

Figure 15: Classical mosaicplot of `model` (x), `premod` (y) and `sat` (x) from the `carcustomers` dataset.

Affiliation:

Alexander Pilhöfer, Antony Unwin
Department of Computer Oriented Statistics and Data Analysis
Institute of Mathematics
Universität Augsburg
86135 Augsburg, Germany
E-mail: alexander.pilhoefer@math.uni-augsburg.de,
antony.unwin@math.uni-augsburg.de
URL: http://www.rosuda.org/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 53, Issue 7 *Submitted:* 2011-02-22
May 2013 *Accepted:* 2013-05-19