

Spacing in the Strucplot Framework

by David Meyer, Achim Zeileis, and Kurt Hornik

Spacing of strucplot displays is customizable in a similar way than shading. The `spacing` argument of the `strucplot()` function takes a list of unit vectors, one for each dimension, specifying the space between the tiles corresponding to the levels. As an example, consider the ‘`Arthritis`’ data resulting from a double-blind clinical trial investigating a new treatment for rheumatoid arthritis. Since we are interested in the effect of the medicament in the placebo and treatment groups, we could use a mosaic plot to visualize the three levels of ‘Improved’ in the two ‘Treatment’ strata (see Figure 1):

```
> (art <- structable(~Treatment + Improved, data = Arthritis,
+   split_vertical = TRUE))
```

| | Treatment | Placebo | Treated |
|----------|-----------|---------|---------|
| Improved | | | |
| None | | 29 | 13 |
| Some | | 7 | 7 |
| Marked | | 7 | 21 |

```
> mosaic(art)
```

Another conceptual approach, however, is to use spine plots with highlighting (Hummel, 1996). A spine plot is a variation of a bar plot where the heights of the bars are held constant, whereas the widths are used to represent the number of cases in each category. This is equivalent to a mosaic plot for a one-way table. If a second (indicator) variable is highlighted in a spine plot, we obtain a display equivalent to a simple mosaic display for a two-way table, except that no space between the levels of the highlighted variable is used. In the ‘`Arthritis`’ example, we will highlight patients with ‘Marked’ improvement in both groups. To obtain such a display within the strucplot framework¹, it suffices to set the space between the ‘Improved’ tiles to 0 (see Figure 2):

```
> (my_spacing <- list(unit(0.5, "lines"), unit(c(0, 0),
+   "lines")))
```

```
[[1]]
[1] 0.5lines
```

```
[[2]]
[1] 0lines 0lines
```

```
> my_colors <- gpar(fill = c("lightgray", "lightgray",
+   "black"))
> mosaic(art, spacing = my_spacing, gp = my_colors)
```

The strucplot framework also provides a set of spacing grapcon generators that compute suitable spacing objects for typical applications. The simplest spacing is `spacing_equal()` that uses the same space between all tiles (see Figure 3):

```
> mosaic(art, spacing = spacing_equal(unit(2, "lines")))
```

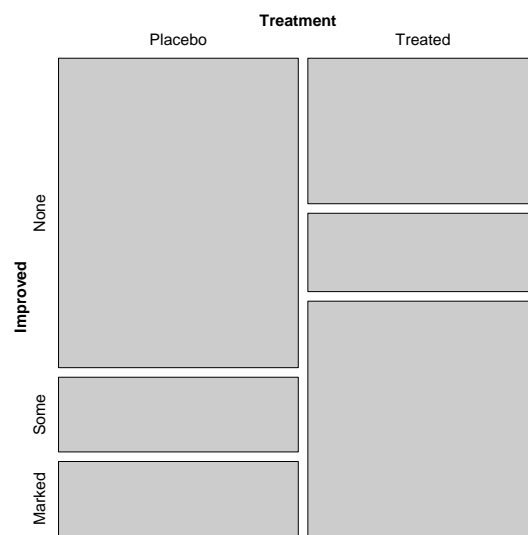


Figure 1: Mosaic plot visualizing the 'Arthritis' data.

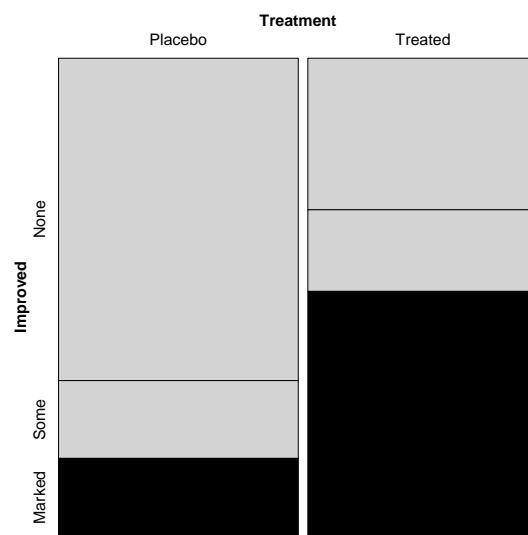


Figure 2: Spine plot for the 'Arthritis' data using the strucplot framework.



Figure 3: Fixed spacing for all dimensions.

`spacing_equal()` is the default `grapcon` generator for two-dimensional tables. Slightly more flexible is `spacing_dimequal()` that allows an individual setting for each dimension (see Figure 4):

```
> mosaic(art, spacing = spacing_dimequal(unit(1:2, "lines")))
```

The default for multi-way contingency tables is `spacing_increase()` that uses increasing spaces for the dimensions. The user can specify a start value and the increase factor (see Figure 5):

```
> mosaic(art, spacing = spacing_increase(start = unit(1,
+ "lines"), rate = 1.5))
```

For the arthritis example above, we could as well have used `spacing_highlighting()` that is similar to `spacing_increase()` but sets the spacing in the last splitting dimension to 0 (see Figure 2):

```
> mosaic(art, spacing = spacing_highlighting, gp = my_colors)
```

Finally, `spacing_conditional()` can be used for visualizing conditional independence: it combines `spacing_equal()` (for the conditioned dimensions) and `spacing_increase()` (for the conditioning dimensions). As an example, consider the ‘PreSex’ data. The conditional independence of premarital and extramarital sex, given marital status and gender, can be visualized using the following (see Figure 6):

```
> mosaic(~PremaritalSex + ExtramaritalSex | MaritalStatus +
+ Gender, data = PreSex)
```

The conditional spacing clearly allows to better distinguish the conditioning variables (‘Gender’ and ‘MaritalStatus’) from the conditioned variables (‘PremaritalSex’ and ‘ExtramaritalSex’). This spacing is the default when conditional variables are specified for a `strucplot` display.

References

Hummel J (1996). “Linked bar charts: Analysing categorical data graphically.” *Computational Statistics*, **11**, 23–33.

¹The `vcd` package also provides a specialized `spine()` function for spine plots and spinograms.



Figure 4: Fixed, but individual spacing for each dimension.

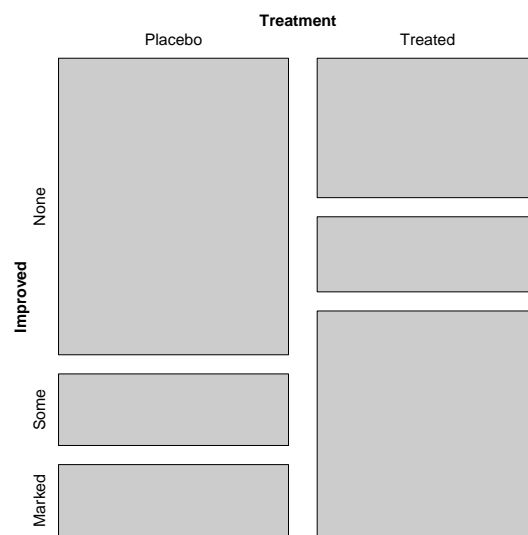


Figure 5: Increasing spacing.



Figure 6: Spacing appropriate for conditional independence.