

Trellis Displays vs. Interactive Graphics II.

Martin Theus,
Lehrstuhl für Rechnerorientierte Statistik und Datenanalyse
Institut für Mathematik, Universität Augsburg, 86135 Augsburg, Germany

Summary

The curse of dimensionality as described in (HUBER 1985) is not restricted to mathematical statistical problems, but can be found in graphic based data analysis as well. Most plots like histograms or boxplots can only handle one single variable. Scatterplots can cope with two continuous variables, rotating plots with three. Mosaic plots (UNWIN 1995) can deal with a lot of categorical variables – although the interpretation may be hard.

We find two common possibilities to bypass this restriction. On the one hand there are interactive statistical graphics, which allow the user to link plots together, and thus by highlighting achieving a multidimensional insight into the data. On the other hand (BECKER et. al. 1994a-c) have suggested Trellis Displays, which offer the possibility of combining up to eight variables in one plot panel.

The main differences between those dynamic and static visualisation techniques as well as the particular power and disadvantages shall be discussed here.

1. Trellis Displays

1.1 Definition:

Historically Trellis Displays are based upon the so-called *Co-Plots*. They were first mentioned in (CHAMBERS 1992). "Co" stands for "conditioning", what means that a specific plot is drawn for different subsets of a conditioning variable.

The generalisation to Trellis Displays is presented by William S. Cleveland in (CLEVELAND 1993). In (BECKER 1994a) to (BECKER 1994c) Becker, Cleveland and Shyu describe the setup of Trellis Displays by means of an example.

Trellis Displays are now available as an S-Plus version 3.2 library (see MATH-SOFT 1994), and will be part of S-Plus version 3.3.

The core of such a display is the one-, two- or three-dimensional graphics of the so-called *axis variables*. Those plots are designated as *panel functions*. The kind of plot is not limited any further; they can be dot-plots, scatter-plots, box-plots, surface-plots etc.

Besides the *axis variables*, there can be up to three *conditioning variables* chosen, to build the trellis. These are either categorical, or they have to be subdivided into several overlapping intervals. Those variables, called *shingles*, are then interpreted as categorical. For each category (or interval) or each combination of categories the *panel function* is called to draw a panelplot.

The plots are arranged in a vector (one conditioning variable) a matrix (two conditioning variables) or a set of parallel matrices (three conditioning variables).

Finally two more categorical or *shingle* variables, called *adjunct variables*, can be displayed by using different colors and markers.

The number of categories of the *conditioning variables* is limited to about eight to ten. The limitation of categories with the *adjunct variables* is much tighter. Using more than three or four colors or markers would confuse the viewer of the plots too much.

A critical point in the design of Trellis Displays is the scaling of each *panel function*. Trellis Displays guarantee that all plots have the same scale, thus facilitating comparison of the plots.

1.2 An Example: The Car Data

To explain the specific elements of a Trellis Display I show the example of the Cars dataset, taken from the DataDesk datasets (VELLEMAN 1992). Figure 1 shows a Trellis Display of scatterplots for Miles per Gallon (MPG) vs. Weight. The conditioning variables are chosen as Continent (derived from Country) and Zylinder (the only car with 5 cylinders has been added to a group of less than 6 cylinders).

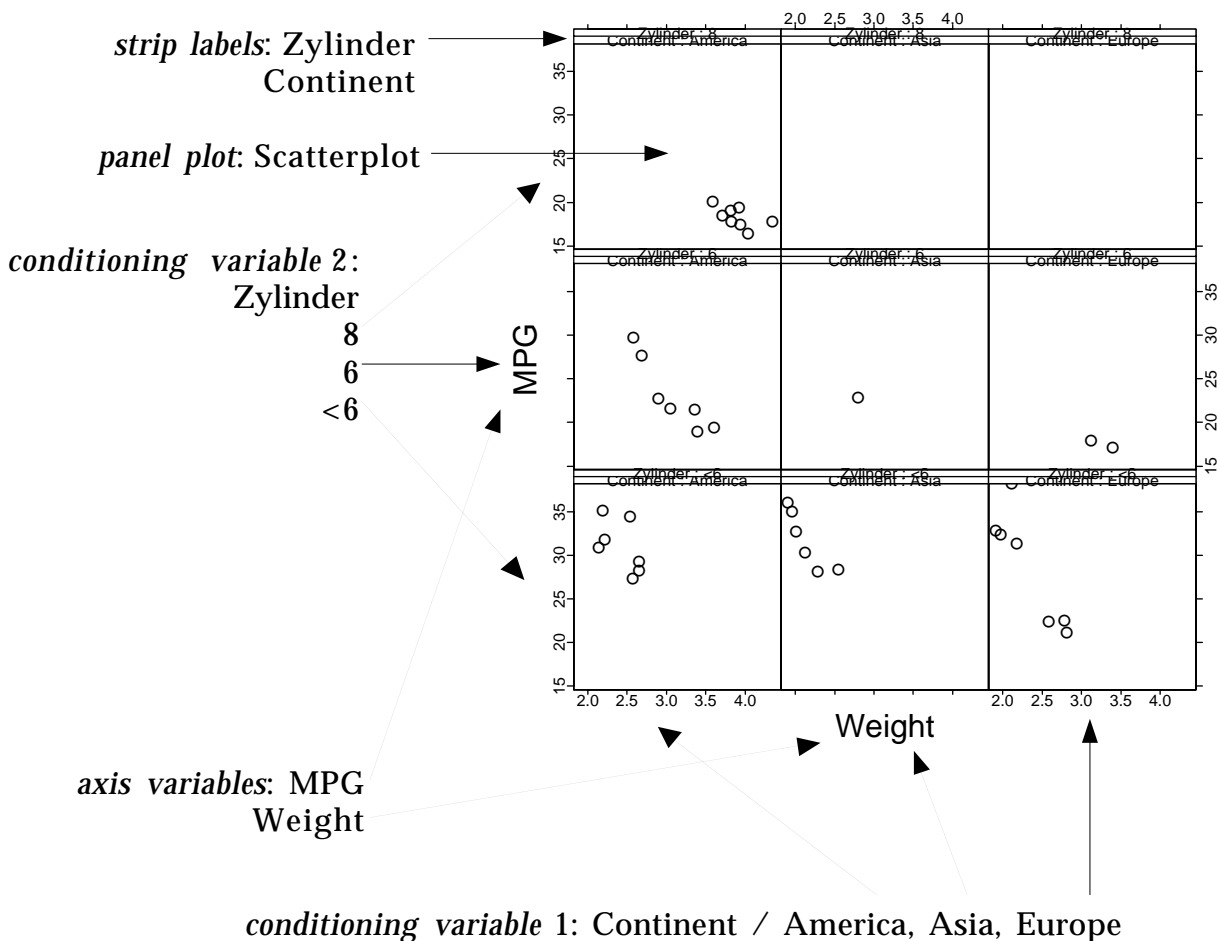


Figure 1: A sample Trellis Display with all elements

The underlying text-based input was:

```
> xyplot( MPG ~ Weight | Continent * Zylinder, data = Cars )
```

This input is easy to understand and intuitive. MPG is plotted in a scatterplot by Weight. The conditioning variables are Continent and Zylinder which form rows and columns of the display. The data is taken from the dataframe Cars. (To achieve a meaningful order of the variable Zylinder it is useful to order the levels by Weight:

```
> Cars$Zylinder <- reorder.factor( Cars$Zylinder, Cars$Weight ) )
```

The display can be extended with a regression line in each panel-plot. This can theoretically be done in an easy way, too. There has to be one argument added, specifying the new panel function:

```
... panel = function(x,y)
  {
    panel.xyplot(x,y)
    panel.lmline(x,y)
  } ...
```

Unfortunately in our example this modification ends up in an error message, because of empty panels or too few observations in a panel. This leads to a further modification, which needs a closer knowledge of the underlying programming language of S-Plus (BECKER 1988).

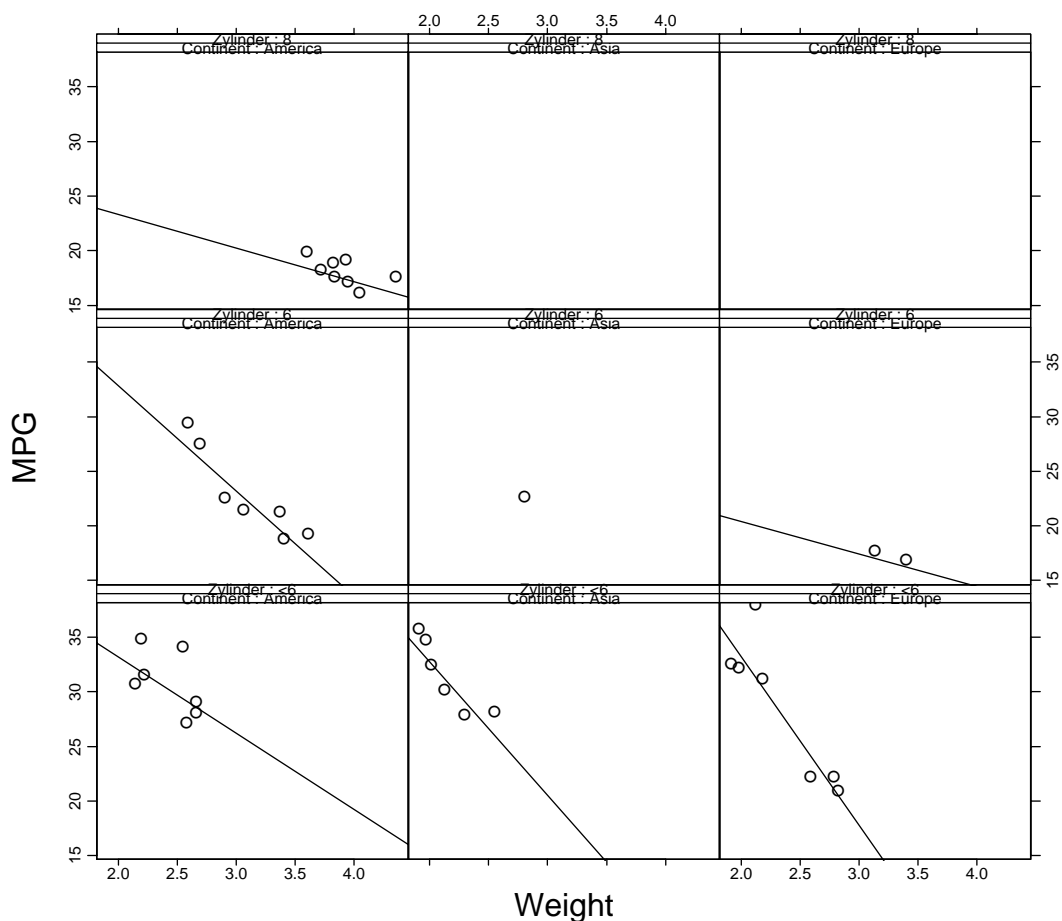


Figure 2: A regression line superposed on all panels

```

... panel = function(x,y)
  {
    panel.xy(x,y)
    if( length(x) > 1 )
      lmline(x,y)
  } ...

```

The resulting plot is shown in figure 2. The implementation in the S-Plus environment seems to yield unnecessary limitations on the theoretically flexible setup of Trellis Displays.

1.3 Trellising: Which plot is best?

Trellising means the assignment of the different variables of a dataset to the different components of a Trellis Display. In the above example it was easy to assign the roles of the four variables. The results are not very surprising even if the roles of Continent and Zylinder were exchanged. But this correct assignment was supported by a preceding knowledge of the data, and an insight into the underlying technical background.

A complete examination of the dataset would include all 7 variables. Two of them are categorical the rest – 5 – are continuous. Not including shingle variables, this would lead us to ten scatterplots (two chosen out of five). Permuting the x- and y-role as well as the assignment of the conditioning variables yields four times as many plots, giving 40.

Using shingling, there is no natural assignment of the variables any longer. This means, that we have $n!$ potential arrangements. For the Car dataset we get $7!=5040$ permutation resp. plots – which is more than a data analyst would like to produce.

Figure 3 depicts the possible permutations of the variables, which are abbreviated by the numbers 1 to 7.

Variable Typ:	Variable Assignment:								
Axis x	1	1	1	1	1	1		7	7
Axis y	2	2	2	2	2	2		6	6
Conditioning 1 (rows)	3	3	3	3	3	3		5	5
Conditioning 2 (columns)	4	4	4	4	4	4	...	4	4
Conditioning 3 (pages)	5	5	6	6	7	7		3	3
Adjunct 1 (markers)	6	7	5	7	5	6		1	2
Adjunct 2 (colours)	7	6	7	5	6	5		2	1

Figure 3: A schematic listing of all permutations

Typically only a few assignments offer a relevant view to the data. But changing and understanding a specific assignment is hard. Here, Interactive Statistical Graphics seem to have a great advantage, because for most variables there is just one suitable plot, from which we can derive all relevant views, using linking and the appropriate selection techniques. This leads to no more than $k*n$ plots, where k is a small integer.

A brief description of this situation will be given in section 2.

1.4 Limitations:

Although a lot of plot-types are offered in the Trellis Display library, there are **no** plots for categorical data. This is because S-Plus itself only offers a piechart as a plot for categorical data. Barcharts as well as mosaic plots are missing. Therefore it is not possible to analyse categorical data with Trellis Displays, although they can be found very often in surveys.

On the other hand all variables besides the axis variables must be categorical. To achieve this, the shingling mechanism is built into the trellis library. As mentioned before, shingling means to subdivide a continuous variable into several overlapping intervals. This is done automatically with some control options. Shingling is equivalent to the slicing process in interactive graphics. But in contrast to interactive graphics, the user has no visual control about what is going on. This can be very misleading, because the individual structure of the shingled variable – e.g. gaps, ties etc. – are not considered at all by the automatic process.

In close connection to shingling, we find another problem. How many observations are inside each plotpanel. In scatterplots e.g., there is the possibility of comparing the number of observations falling into the different intersections visually. But all plots that summarise the data, e.g. boxplots, hide the actual amount of data being used to form the plot. This can be dangerous when judging optional model-fits in a Trellis Display. Comparing figure 2, a strange regression line would be only interesting, if it is based upon a major amount of datapoints. For this it is necessary to have a clear knowledge about the underlying number of observations in each plotpanel.

2. Interactive Statistical Graphics

2.1 Overview:

In order to fight the curse of dimensionality Interactive Statistical Graphics takes a nearly opposite approach to Trellis Displays. While Trellis Displays try to incorporate all variables at a time, Interactive Statistical Graphics use lower dimensional plots, that suit the variables best. To achieve a multidimensional insight to the data, selecting, highlighting and linking inside the different plots is used.

Selecting data can be done in various ways. From single point selection over brushing and slicing to lassoing, where irregular areas in a plot can be captured, we find a lot of flexible ways to highlight data in the linked plots. Linking denotes the mechanism to propagate the highlighting action in one plot to all other active plots. A brief description of interactive software and techniques can be found in (WILHELM et. al. 1995) (same book).

2.2 Where Trellis Displays and Interactive Graphics meet:

Since Trellis Displays can be seen as a general concept of systematic plotting, I like to point out, where we can find Trellis Displays beside the implementation as a S-Plus library. The boxplot y by x and dotplot y by x inside DataDesk is a simple form of the

function `boxplot` in the Trellis Display library. Selecting MPG as y-variable and country as x-variable and choosing `boxplot y by x` will lead to figure 4 – which is a Trellis Display.

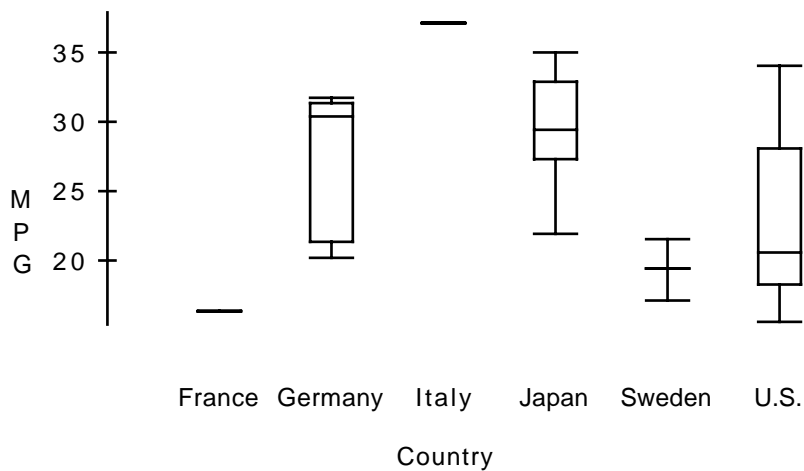
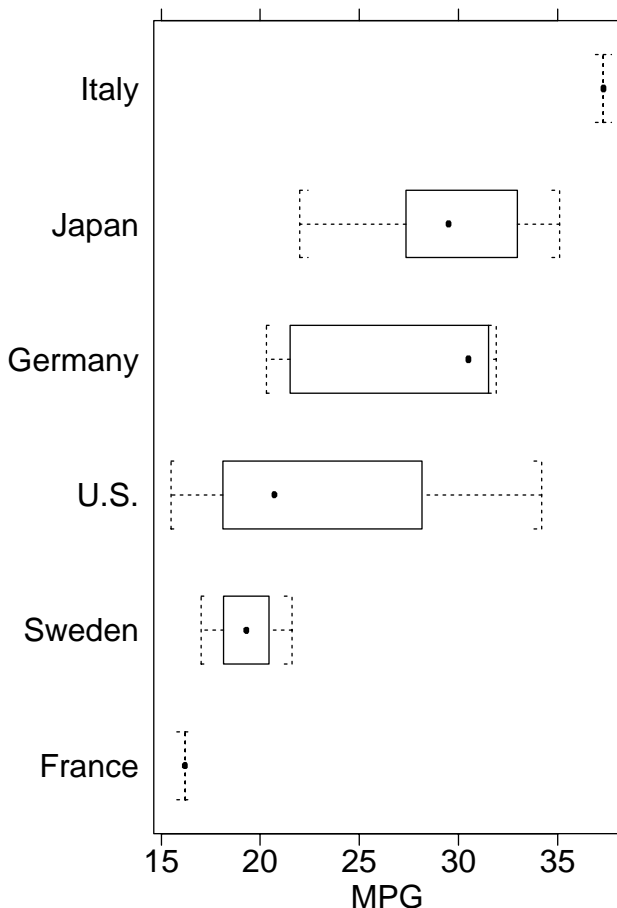


Figure 4: A x by y boxplot set up with DataDesk



This plot could also be constructed with the S-Plus call:

```
> boxplot(MPG~Country, data=Cars)
```

which is shown in figure 5. The main difference between the two plots is that on the one hand the DataDesk plot is still interactive i.e. linked to all other plots but on the other hand no further conditioning variable could be added to the plot, as it would be possible with Trellis Displays.

Furthermore we find an advantage of Trellis Displays here, because it is relatively easy to order the levels of country by the size of MPG, which was done in figure 5. This offers a good possibility for comparing the different

Figure 5: The corresponding boxplot to figure 4 as Trellis Displays in S-Plus

countries.

2.3 Working with Interactive Graphics: Returning to the Example

Remember figure 2. To achieve the same plots with DataDesk, it is suitable to set up a barchart for Continent and Zylinder each, together with a scatterplot for MPG vs. Weight. If we set the scatterplot to automatic update and hot selection mode and freeze



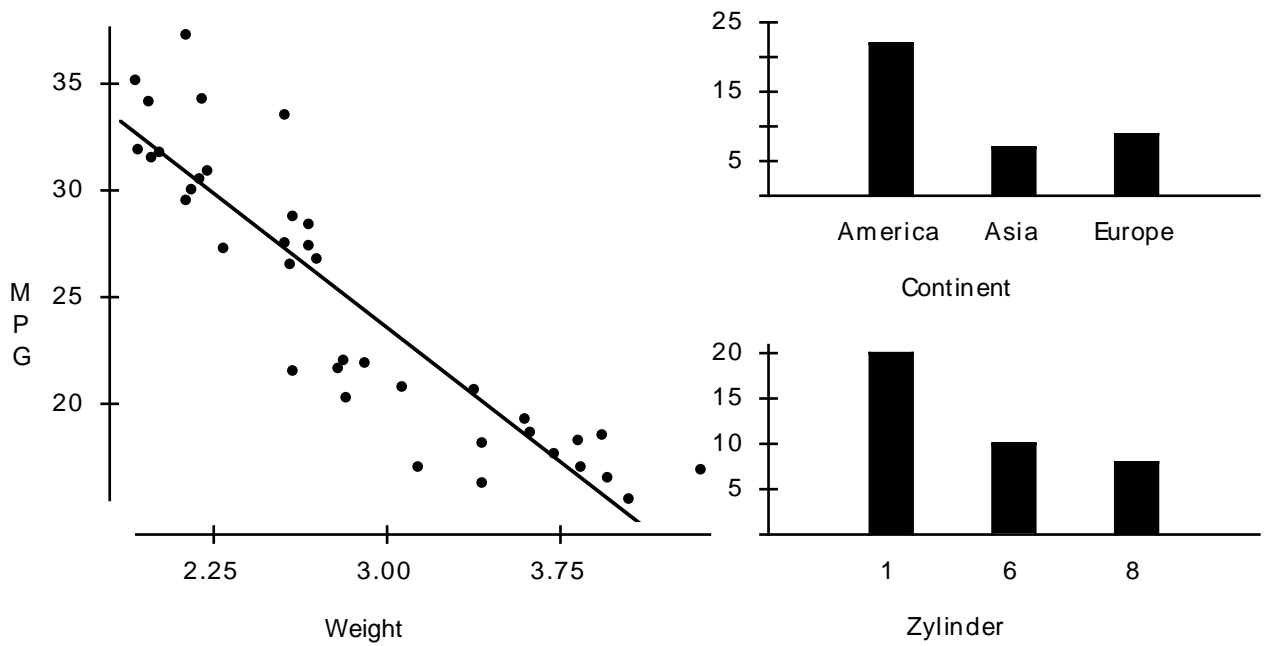


Figure 6: The three plots from which all other view can be derived

scale, we can derive all plots we find in figure 1. Adding the regression line will deliver the corresponding plots from figure 2. This is shown in figure 6 with all points selected. To get the single plots in figure 2 one has to switch to the intersection selection mode and click the individual bars in the barcharts. Figure 7 shows the

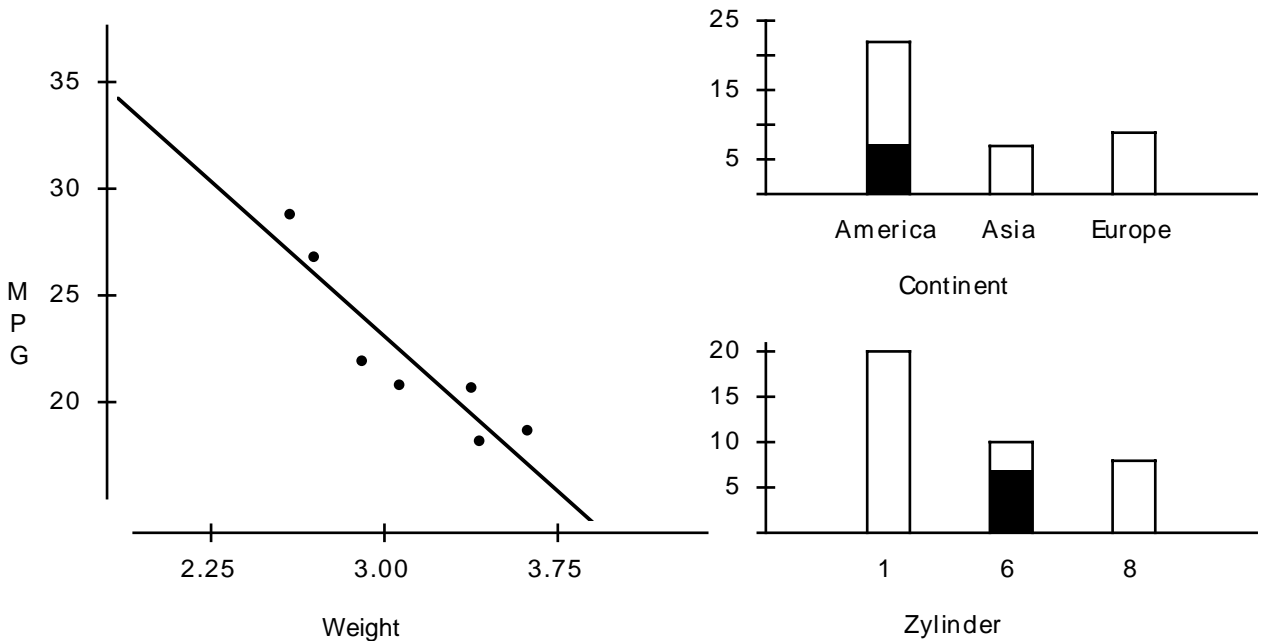


Figure 7: The panelplot in row two, column one of figure 2 setup with DataDesk

scatterplot for all 6 cylinder cars made in the U.S. It can be seen, that the regression line for this group is close to the line for the whole dataset. All other plotpanels from figure 2 can be accessed in this easy way. In addition the row- and columnsums can be plotted, too. The main advantage of Interactive Statistical Graphics is the fast and hierarchical way we get to the different views of the data.

2.4 The Curse of Static Plots: Ghostplotting

As could be seen in the previous sections, interactions to the plots offer a much faster and efficient insight into statistical data than static graphics could do. But often only the toggling between groups, rotating of a pointcloud or dynamic change of a parameter offers the specific visualisation impact. It is obvious, that a static representation of such plots loses a lot of that visualisation impact. Although WorldWideWeb offers new possibilities for multimedia presentations, static plots in black and white printing will dominate publications in the fields of statistics in the near future. Ghostplotting partially seems to solve this problem.

Ghostplotting is the technique of overlaying plots of different subgroups of data. To guarantee the distinction of the different groups, different shades of grey are used.

To illustrate Ghostplotting we use the Cars dataset again. Figure 8 shows a boxplot MPG by Continent for two groups of the variable Zylinder. Whilst the group with less

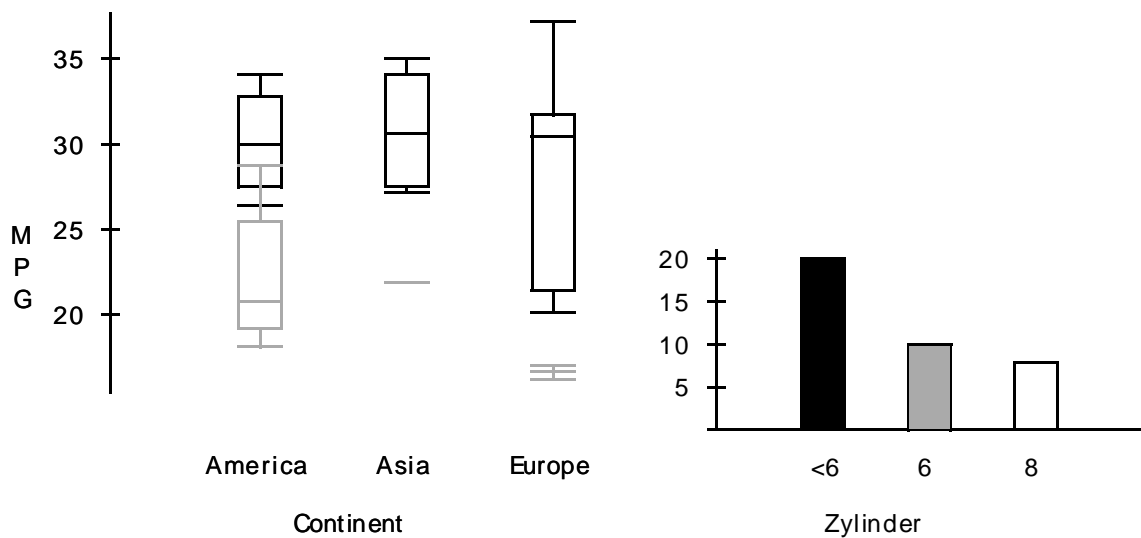


Figure 8: A sample Ghostplot for MPG by Continent by Zylinder

than six cylinders is marked in black, the group for six cylinders is plotted in grey. It can be found from this plot, that the difference between those two groups is the largest in Europe. (Be aware of the small sample size for demonstration purposes.)

Figure 8 is a trial to imitate the effect, when toggling between the two groups by clicking the two bars in the bar chart.

The use of Ghostplotting in boxplots is only one aspect. Ghostplotting can be generalised to nearly all other statistical plots. Figure 9 shows an application of Ghostplotting in time series analysis. The plot shows the estimated seasonal pattern of German unemployment in the years 1960 to 1995. In order to compare the development of the patterns, all 25 years are overlayed in one plot. The recent years are plotted in black, whereas the earlier years decrease in the shade of grey. The plot shows clearly, that the extra peak in July occurred in the last 10 to 15 years. The overall magnitude increased in the middle of the period, but is decreasing to the level of the 60s again.

No statistics software tool offers Ghostplots. Thus they have to be produced manually with the help of graphics-software.

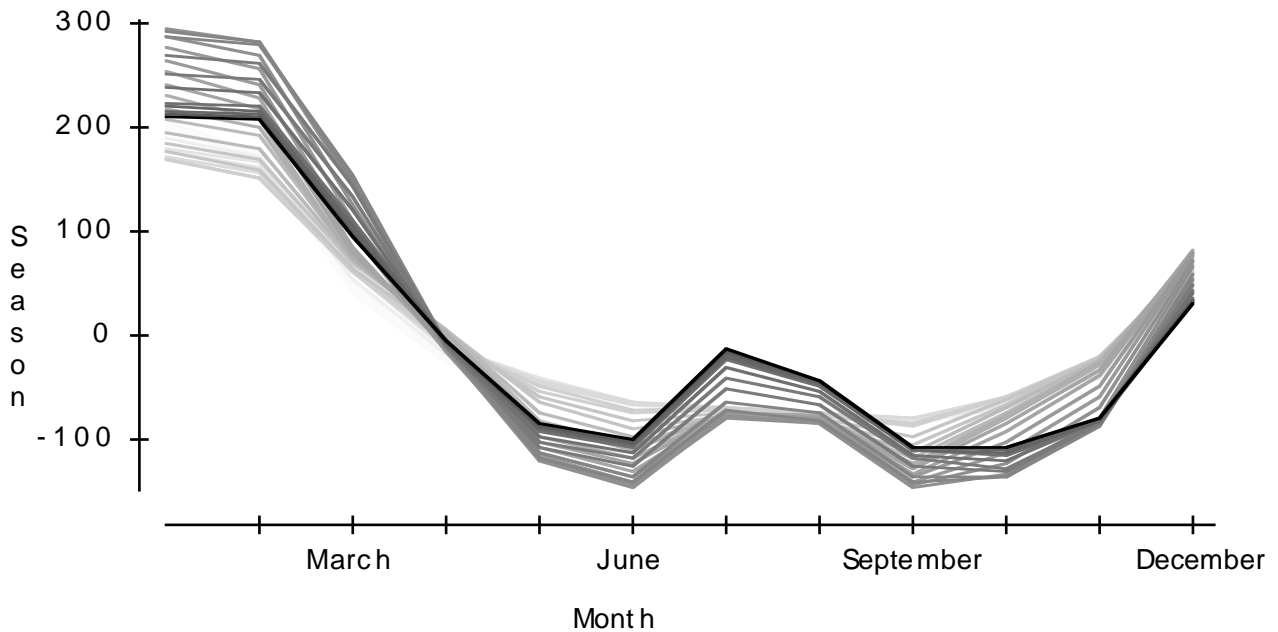


Figure 9: A Ghostplot to analyse the seasonal pattern of the German unemployment

3. Conclusions

As already pointed out in (THEUS 1995), Trellis Displays offer a lot of useful extensions to standard static graphics. No other tool offers the possibility to set up plots in a systematic way like the Trellis-library inside S-Plus does. These possibilities are provided through the underlying programming environment of S-Plus. Non experienced S-Plus users have to be content with the high level functions of the library, which are often not very satisfying.

The main difference between Trellis Displays and Interactive Statistical Graphics can be found in the contexts of data analysis. The flexibility of interactive tools and techniques are very useful for the exploration of datasets. Here Trellis Displays appear to be too clumsy to achieve a deep insight into the data. On the other hand, once one has found an appropriate model that suits the data, Trellis Displays can help to visualise this model very well. Thus Interactive Graphics and Trellis Displays can complement data analysis in the following way:

1. Analyse the data and look for a suitable model with interactive techniques
2. Present the model graphically with Trellis Displays.

As shown in section 1.4, there is still room for improvement for Trellis Displays. Rumours say, that the authors of Trellis Displays are working on an improved version of Trellis Displays, to cover the weakness of the current implementation and to reach the full power of the underlying theoretical concept.

A possible extension to Trellis Displays is further more the embedding of Trellis Displays into an interactive environment. This would lead to a more powerful use of Interactive Statistical Graphics as well as of Trellis Displays. A first implementation is described in (THEUS 1996).

References

- BECKER, Richard A., CHAMBERS, John M., WILKS, Allan R. (1988)
The New S Language, A Programming Env. for Data Analysis and Graphics
Wadsworth & Brooks/Cole, Pacific Grove CA.
- BECKER, R. A., CLEVELAND, W. S., SHYU, Ming-Jen, KALUZNY, St. P. (1994a)
Trellis Display: A Framework for Visualizing 2D and 3D Data
AT &T Bell Laboratories Statistics Research Report No. 8
- BECKER, R. A., CLEVELAND, W. S., SHYU, Ming-Jen, KALUZNY, St. P. (1994b)
Trellis Display: Questions and Answers
AT &T Bell Laboratories Statistics Research Report No. 9
- BECKER, R. A., CLEVELAND, W. S., SHYU, Ming-Jen, KALUZNY, St. P. (1994c)
Trellis Display: User's Guide
AT &T Bell Laboratories Statistics Research Report No. 10
- CHAMBERS, John M., HASTIE, Trevor J. eds. (1992),
Statistical Models in S
Wadsworth & Brooks/Cole, Pacific Grove CA.
- CLEVELAND, William S. (1993)
Visualizing Data
Hobart Press, Summit NJ.
- MATHSOFT (1994)
S-Plus Trellis Displays User's Manual, Version 1.0
MathSoft Inc., Seattle.
- HUBER, Peter J. (1985)
Projection Pursuit
The Annals of Statistics, Vol. 13, pp. 435–475
- THEUS, Martin (1995)
Trellis Displays vs. Interactive Statistical Graphics.
Computational Statistics, Vol. 10 Issue 2, pp. 113–127
- THEUS, Martin (1996)
MANET – Extensions to Interactive Statistical Graphics for Missing Values
in: Proceedings of the NTTS '95
- UNWIN, Antony R. (1995)
Interactive Graphics for Data Sets with missing values – MANET
submitted to the Journal of Computational and Graphical Statistics
- VELLEMAN, Paul F. (1992)
Data Desk
Data Description, Ithaca, New York.
- WILHELM, Adalbert, UNWIN Antony R. & THEUS, Martin (1995)
Software for Interactive Statistical Graphics – A Review
SoftStat '95, Advances in Statistical Software 5, eds. Faulbaum, F., Bandilla, W.,
Lucius & Lucius, Stuttgart