Multivariate displays – Coplots

5. Trellis graphics

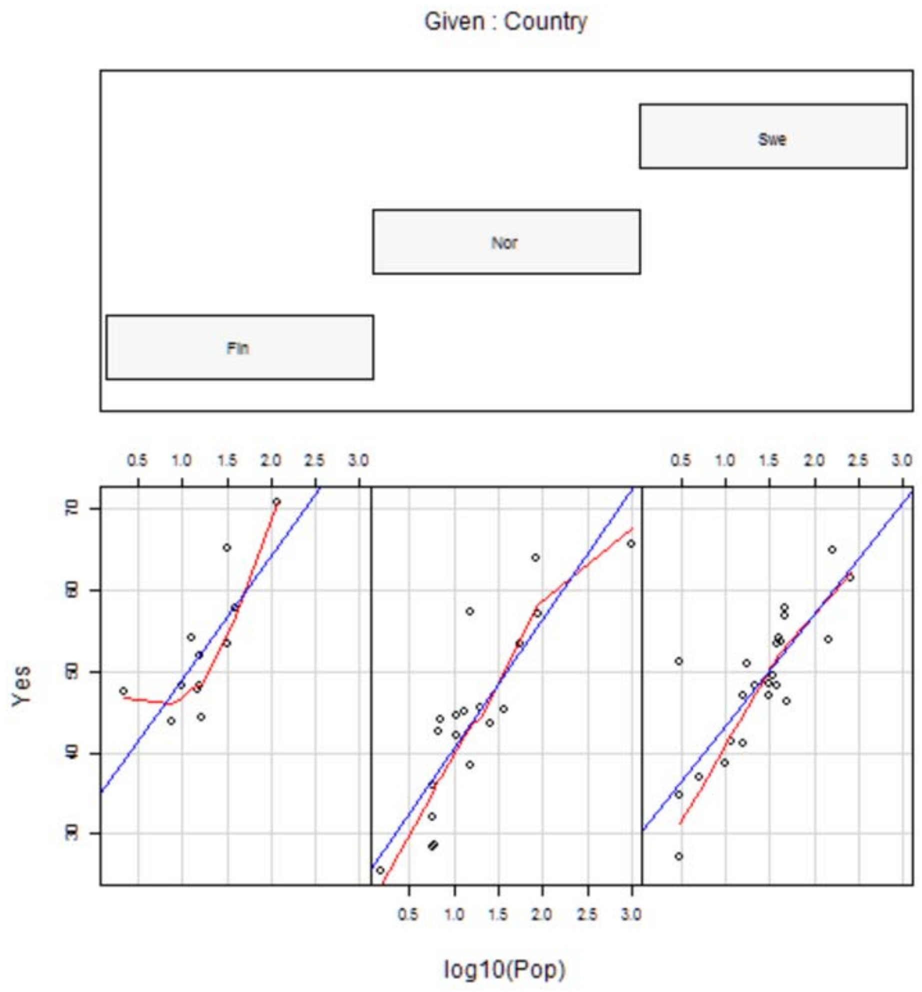
Many data sets include a mixture of both “continuous” (ordinal-, interval- or ratio-scale variables) and “discrete” (nominal-scale variables). Often, the issue might arise of how a particular relationship between variables might differ among groups. Information of that nature can be gained using conditioning plots (or coplots). Such plots are part of a general scheme of visual data analysis, known as Trellis Graphics that has been created by the developers of the S language. Trellis Graphics are implemented in R using the 'lattice package.

Coplots (conditioning scatter plots)

Conditioning scatter plots involves creating a multipanel display, where each panel contains a subset of the data. This subset can be either a) those observations that fall in a particular group, or b) they may represent a the values that fall within a particular range of the values of a variable. The idea is that the individual panels should illustrate the relationship between a pair of variables, over part of the range of the two marginal “conditioning” variables (i.e. the relationship “conditional on one marginal variable lying in one particular interval, and the other lying in a different interval.”)

This coplot contains scatter diagrams for Yes as a function of the log(10) of Population, conditioned by country (i.e. one “response” variable and one “conditioning” varible.)

```
library(lattice)
attach(scanvote)
coplot(Yes ~ log10(Pop) | Country, columns = 3, panel = function(x, y, ...) {
    panel.smooth(x, y, span = 0.8, iter = 5, ...)
    abline(lm(y ~ x), col = "blue")
})
```

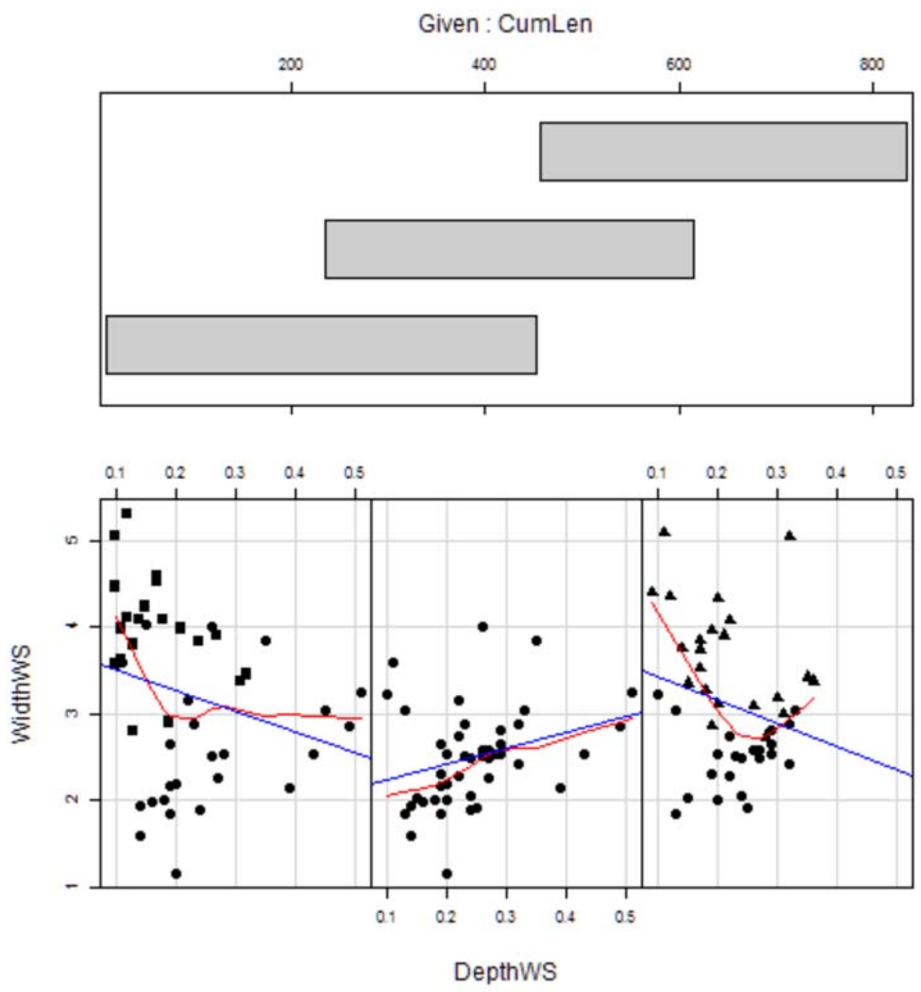


Note the use of the “panel” function here. Basically, what's going on is that the **`coplot()`** function is determining which subset of observations should appear in each panel, while the two function calls within the **panel()** function (**panel.smooth()** and **abline()**) perform their tasks on that subset of observations. In other words, **coplot()** selects the observations of **Yes** and **log(Pop)** for a particular panel (i.e. **Country**), sends these to the panel function, which passes them on (relabeled as x and y), and plots the points, and then **panel.smooth()** and **albline()** draw a lowess curve and least-squares line for those observations on each panel (more about those later). The general idea is to compare the panels (countries) seeing where in the panel the points lie and what the relationship looks like. The general relationship between population and percent of **Yes** votes is apparent, as well as country-to-country differences, like the generally greater proportion of **Yes** votes in Finland.

Coplot, conditioning by one continuous numeric variable

Most of the time, the conditioning variables are continuous numeric variables. Here's a coplot for **WidthWS** as a function of **DepthWS** in the Summit Cr. data set, conditioned by **CumLen** (or distance downsteam):

```
attach(sumcr)
coplot(WidthWS ~ DepthWS | CumLen, pch = 14 + as.integer(Reach), cex = 1.5,
    number = 3, columns = 3, panel = function(x, y, ...) {
        panel.smooth(x, y, span = 0.8, iter = 5, ...)
        abline(lm(y ~ x), col = "blue")
    })
```



We know the arrangement of the reaches, and so the resulting plot should be no surprise. The plotting characters are determined by Reach, to reveal the extent of overlap in the conditioning "shingles." The plot could be regenerated using Reach as the conditioning variable, which would result in no overlap between the individual panels.

Its easy to see that the two grazed reaches (A upstream and C downstream) have generally wider channels, which would be expected. Something that is not apparent in ordinary plots of the data is that the "normal" or expected inverse relationship between width and depth (as one gets bigger the other gets smaller) does not apply in the middle (exclosure) reach.

The main documentation for Trellis graphics includes:

- Trellis Graphics User Manual, and
- A Tour of Trellis Graphics

two .pdf documents published by the developers of the S language and Trellis Graphics, Lucent Technology.