

7.1: How to draw the multivariate data

The most simple operation with multidimensional data is to draw it.

Pictographs

Pictograph is a plot where each element represents one of objects, and every feature of the element corresponds with one character of the primary object. *If the every row of data is unique*, pictographs might be useful. Here is the *star plot* (Figure 7.1.1) example:

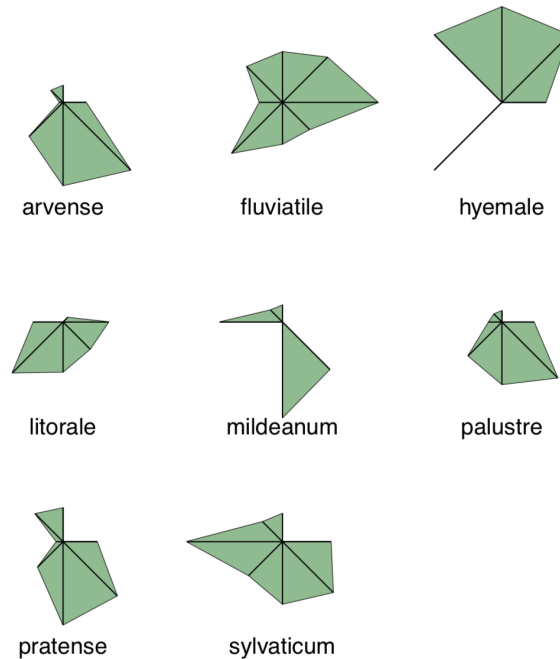


Figure 7.1.1 Stars show different horsetail species.

(We made every element to represent the species of horsetails, and length of the particular ray corresponds with some morphological characters. It is easy to see, as an example, similarities between *Equisetum × litorale* and *E. fluvatile*.)

Slightly more exotic pictograph is *Chernoff's faces* where features of elements are shown as human face characters (Figure 7.1.1): (Original Chernoff's faces have been implemented in the [faces2\(\)](#) function, there is also another variant in [symbols\(\)](#) package.)

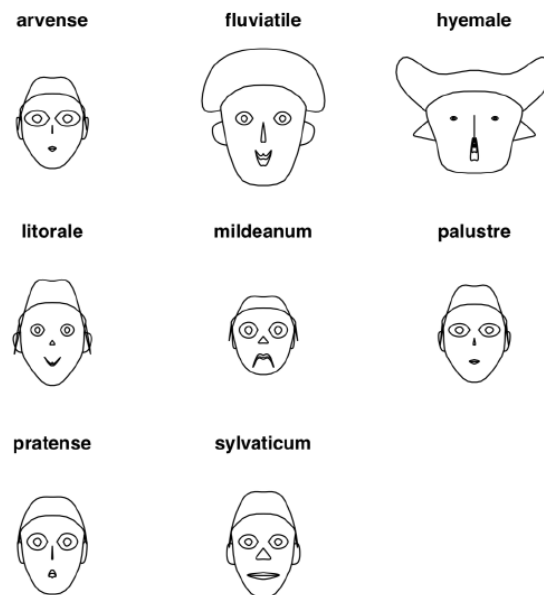


Figure 7.1.2 Chernoff's faces show different horsetail species.

Related to pictographs are ways to overview the *whole* numeric dataset, matrix or data frame. First, command `image()` allows for plots like on Figure 7.1.3:

(This is a “portrait” or iris matrix, not extremely informative but useful in many ways. For example, it is well visible that highest, most red, values of **Pt.L** (abbreviated from **Petal.Length**) correspond with lowest values of **Sp.W** (**Sepal.Width**). It is possible even to spot 3-species structure of this data.)

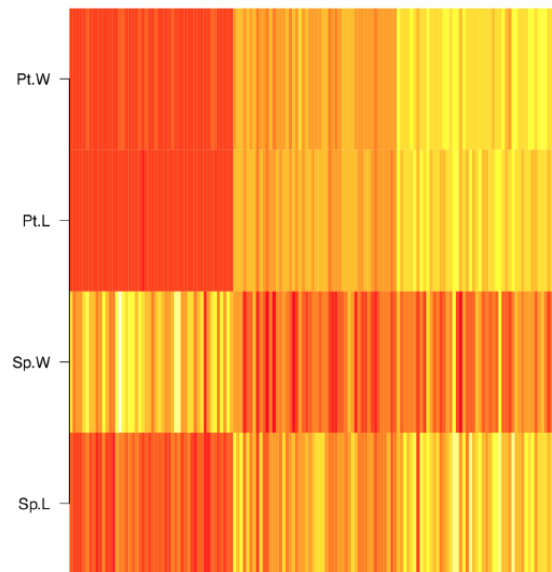


Figure 7.1.3 Results of plotting `iris` data with the `image()` command. Redder colors correspond with higher values of scaled characters.

More advanced is the *parallel coordinates plot* (Figure 7.1.4):

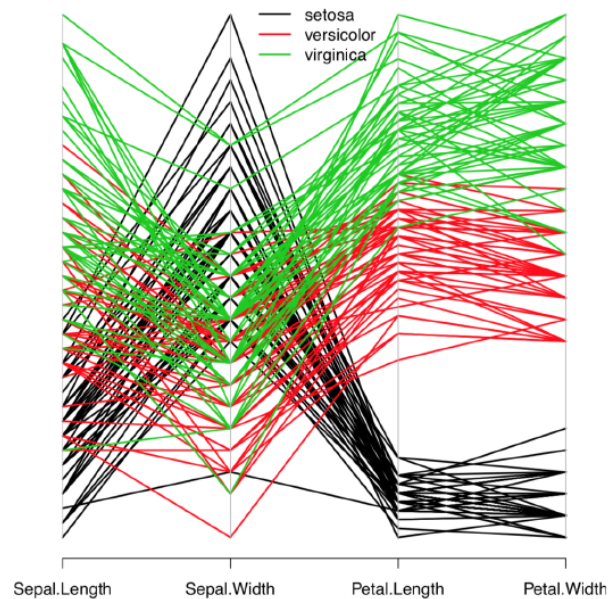


Figure 7.1.4 Parallel coordinates plot.

This is somewhat like the multidimensional stripchart. Every character is represented with one axis which has its values from all plants. Then, for every plant, these values were connected with lines. There are many interesting things which could be spotted from this plot. For example, it is clear that petal characters are more distinguishing than sepal. It is also visible that *Iris setosa* is more distinct from two other species, and so on.

Grouped plots

Even boxplots and dotcharts could represent multiple characters of multiple groups, but you will need to scale them first and then manually control positions of plotted elements, or use `Boxplots()` and `Linechart()` described in the previous chapter:

(Please try these plots yourself.)

Function `matplot()` allows to place multiple scatterplots in one frame, `symbols()` allows to place multiple smaller plots in desired locations, and function `pairs()` allows to show multiple scatterplots as a matrix (Figure 7.1.5).

(This matrix plot shows dependencies between each possible pair of five variables simultaneously.)

Matrix plot is just one of the big variety of R trellis plots. Many of them are in the `lattice` package (Figure 7.1.6):

(Note how to use `make.groups()` and `do.call()` to stack all columns into the long variable (it is also possible to use `stack()`, see above). When `LOC` was added to temporary dataset, it was recycled five times—exactly what we need.)

Library `lattice` offers multiple trellis variants of common R plots. For example, one could make the trellis dotchart which will show differences between horsetail species (Figure 7.1.7)

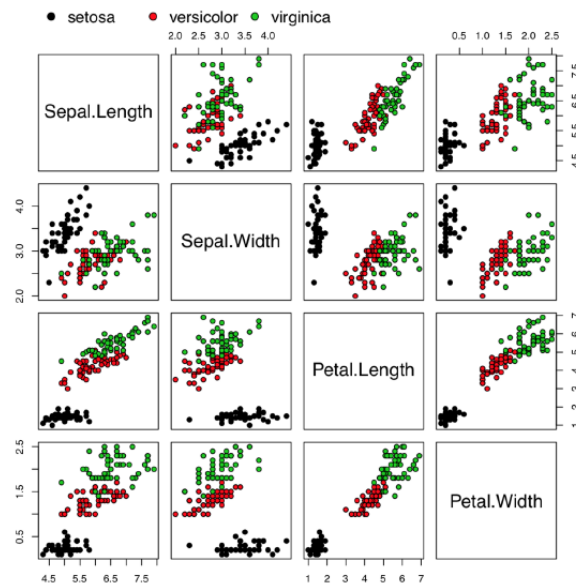


Figure 7.1.5 Matrix plot.

(Here we stacked all numerical columns into one with `stack()`.)

Few trellis plots are available in the core R. This is our election data from previous chapter (Figure 7.1.8):

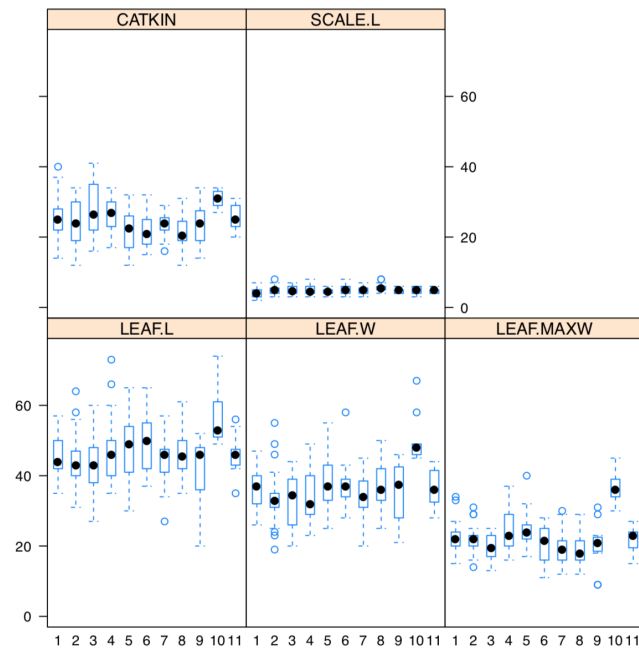


Figure 7.1.6 The example of trellis plot: for each measurement character, boxplots represent differences between locations.

3D plots

If there just three numerical variables, we can try to plot all of them with 3-axis plots. Frequently seen in geology, metallurgy and some other fields are *ternary plots*. They implemented, for example, in the [vcd](#) package. They use triangle coordinate system which allows to reflect simultaneously three measurement variables and some more categorical characters (via colors, point types *etc.*):

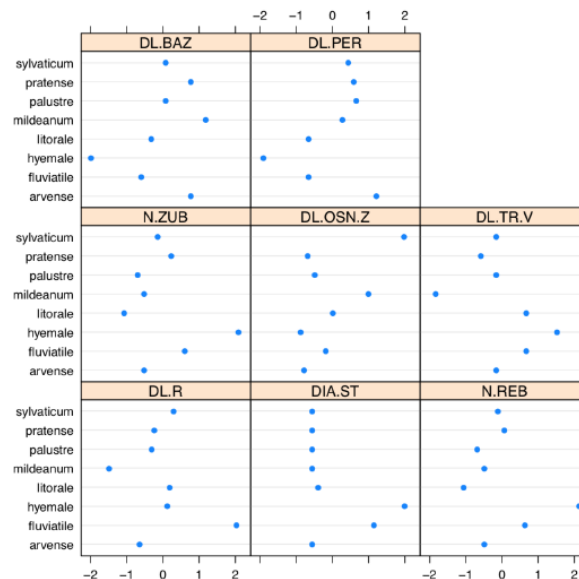


Figure 7.1.7 Trellis dotchart of the horsetail species (character values are scaled). These plots are typically read from the bottom. The “brick” 3D plot could be done, for example, with the package [scatterplot3d](#) (Figure 7.1.10):

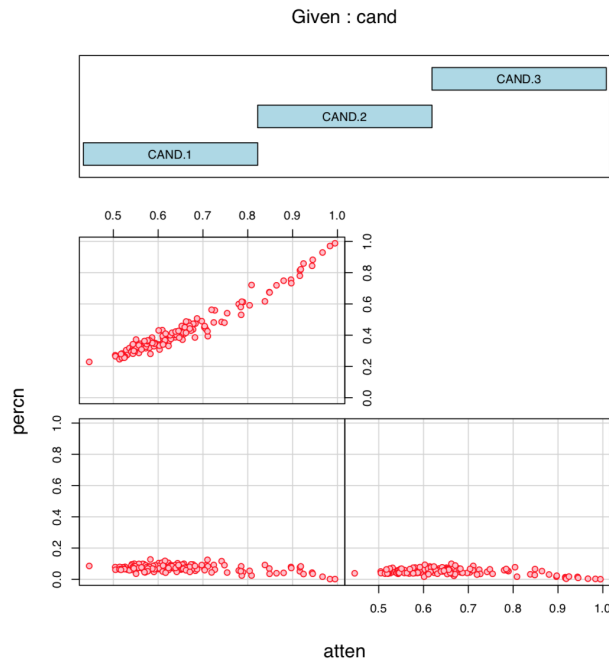


Figure 7.1.8 Voting data from previous chapter represented with `coplot()` function.

(Here some additional efforts were used to make y-axis label slanted.)

These 3D scatterplots look attractive, but what if some points were hidden from the view? How to rotate and find the best projection? Library `RGL` will help to create the *dynamic* 3D plot:

Please run these commands yourself. The size of window and projection in RGL plots are controlled with mouse. That will help to understand better the position of every point. In case of `iris` data, it is visible clearly that one of the species (*Iris setosa*) is more distinct than two others, and the most “splitting” character is the length of petals (`Petal.Length`). There are *four* characters on the plot, because color was used to distinguish species. To save current RGL plot, you will need to run `rgl.snapshot()` or `rgl.postscript()` function. Please also note that RGL package depends on the external OpenGL library and therefore on some systems, additional installations might be required.

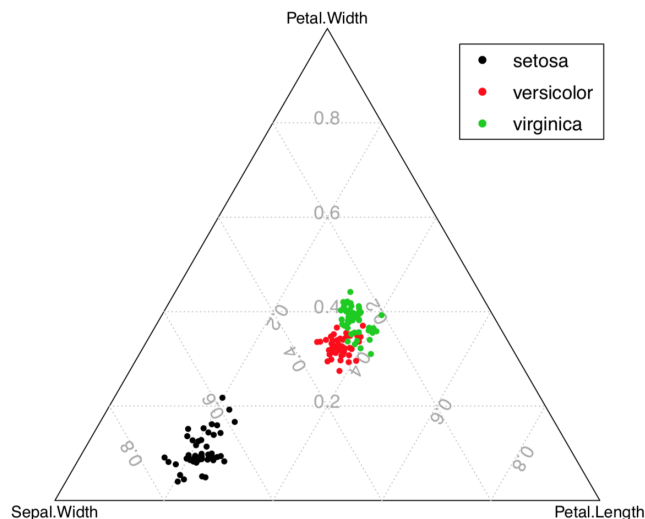


Figure 7.1.9 Ternary plot for `iris` data.

Another 3D possibility is `cloud()` from `lattice` package. It is a static plot with the relatively heavy code but important is that user can use different rotations (Figure 7.2.1):

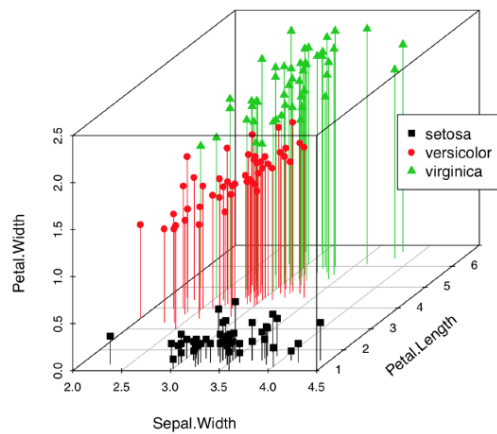


Figure 7.1.10 Static 3D scatterplot of *iris* data.

This page titled 7.1: How to draw the multivariate data is shared under a [Public Domain](#) license and was authored, remixed, and/or curated by Alexey Shipunov via source content that was edited to the style and standards of the LibreTexts platform.