

# Multivariate analysis - Introduction

*Gilles San Martin*

*24 September 2018 - 00h38*

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Organize your datasets and statistical methods . . . . .	2
1.1.1	Variables and observations . . . . .	2
1.1.2	Multivariate vs univariate methods . . . . .	3
1.1.3	Supervised vs unsupervised methods . . . . .	3
1.1.4	Distances - clustering - ordination . . . . .	3
1.1.5	Overview of the methods . . . . .	6
1.2	Simple graphical visualisation of complex datasets . . . . .	8
1.2.1	Descriptive statistics as summary of the data . . . . .	8
1.2.2	Visualise the distribution of the variables . . . . .	10
1.2.3	Faceting . . . . .	13
1.2.4	SPLoMs : Scatterplot Matrices . . . . .	15
1.2.5	Heatmap of the correlation matrix . . . . .	19
1.2.6	Parallel Coordinate plots . . . . .	23
1.2.7	SPLoMs between 2 matrices . . . . .	25

---

```
source("/home/gilles/stats/mytoolbox.R")
setwd("/home/gilles/stats/Formation_R_stats/Formation_Stats_4_Multivariate/")

# load the packages used for this chapter
library(vegan)
library(mlbench)
library(GGally)
library(skimr)
library(Hmisc)
library(corrplot)
library(RColorBrewer)
library(MASS)

# load ggplot, change the default theme and change the locale (language = English)
library(ggplot2)

Sys.setlocale("LC_ALL", 'en_GB.UTF-8')
mytheme <- theme_bw(10) + theme(axis.text.x=element_text(size=8),
                                legend.key = element_rect(color = NA))
theme_set(mytheme)
```

---

# 1 Introduction

## 1.1 Organize your datasets and statistical methods

### 1.1.1 Variables and observations

Thinking about your data analysis in terms of data matrices, variables and observations, will not only help you to choose the right statistical methods but it will also help you to have a better understanding of your research and to refine the scientific questions at hand from general questions (like : “what factors are associated with honey bee mortality ?”) to more specific, actionable questions (like “do we observe more honey bee winter mortalities when we observe more pesticides in the beehive ?”).

Almost all datasets can be organized into one or several data matrices characterized by their :

1. **variables / columns / descriptors / species**
2. **observations / lines / objects / sites**

NB : these terms are used interchangeably in text books or softwares. For example in the **vegan** package the columns are called “species” and the lines are called “sites” even if **vegan** can be used for other type of data.

The variables are the characteristics that are measured (weight, abundance of a species, expression of a given gene, sex, ...). The observations are the sampling units on which the sevariables are measured. Sometimes each sampling unit is subdivided into sub-sampling units (eg triplicates in mollecular biology, vegetation plots within sites,...) and/or is measured several times. Each of these replicates in time or space are a different observation and can be added as a separate line in the dataset with additional variables to identify for example the sampling date, the sub-sampling unit etc. . .

The same variable should never be placed on different columns !! Groups of varaibles about different topics might be stored into different matrices (for example one matrix for species abundance, one matrix for environmental characteristics).

The variables are generally on the columns and the observations on the lines but in some disciplines the data matrix is often transposed. For example in genomics and other \*omics datasets it is frequent to have the genes (peptides, metabolites,...) - that correspond to the variables - on the lines and the samples (= the observations) on the columns. In many cases you will not use the same methods when you work on the variables or the observations so it is particularly important to make the distinction. We will always consider here that your variables are in the columns and the observations are on the columns. . . You need also to check what is the expectation of the pice of software you use.

The collumns (variables) are said to be homogenous if they are measured with the same units. Site x species matrices are typically homogenous (all columns are abundances or presence/ absences) while sites x environmental matrices are typically not homogenous (e.g. mix of temperature, humidity, ...).

The collumns can contain quantitative, presence/absence (binary), ordinal or qualitative variables or a mix of these. This will have an influence on the choice of the methods.

In Ecology we typically have one site x species matrix (Y) and sometimes one corresponding sites x environmental measures matrix (X). You can also typically have additionnal spatial or temporal information that are often stored in other matrices (or in cubic arrays). You can view such a matrix as a multidimensional space. For example in a matrix with 3 species, each species represent an axis of your 3 dimentional space and you can place the sites in this space depending on the abundance of each species in each site.

### 1.1.2 Multivariate vs univariate methods

See also figure 2.

A first usefull distinction can be made between univariate methods where you look at one main variable of interest at a time vs multivariate methods where you look at several variables at the same time. This rather artificial distinction is usefull in practice because most of the time it involves the use of rather different statistical methods.

I will consider here that the frequent case of a multiple regression  $y \sim x_1 + x_2 + x_3 + \dots$  is NOT a multivariate method because we study one variable of interest ( $y$ , the response) at a time even if we have multiple explanatory variables  $x_1, x_2, \dots$ . Note however that this definition is not universal. You will often read about “bivariate models” for simple regression  $y \sim x$  and multivariate models for multiple regressions  $y \sim x_1 + x_2 + x_3$ . In addition some methods like PLS and LDA that are clearly univariate supervised methods in the sense we use it here (their aim is to predict a quantitative or a qualitative variable) are in fact strongly connected to classical multivariate unsupervised methods (eg PCA). The supervised/unsupervised distinction (below) is on the other hand quite universal and broadly accepted (but there are also some methods considered as semi-supervised, ...).

### 1.1.3 Supervised vs unsupervised methods

Typically an other distinction is done between two main statistical approaches (See figure 1):

1. “unsupervised” methods : you want to understand the structure of one matrix at a time ( $Y$ ), i.e. the relationships/similarity/dissimilarity between either the columns or the lines of the matrix. These techniques are mainly clustering (hierarchical clustering, k-means, ...) or ordination methods also called dimensionality reduction methods (PCA, tPCA, CA, PCoA, nMDS, ...).
2. “supervised” methods : you want to understand the relationships between 2 (sometimes more) data matrices ( $Y \sim X$ ). These methods are related to regression analyses (RDA, CCA), correlation (CCorA, Mantel test, ...) or regression trees (MRT). When  $Y$  is a single variable we are in the case of a univariate supervised method with the well known regression like methods : GLMs, & co.

### 1.1.4 Distances - clustering - ordination

The starting point of most of the multivariate techniques is implicitly or explicitly the calculation of a similarity or dissimilarity matrix either between the columns or the rows.

In the explicit case you first calculate the dissimilarity matrix and then you use it as input for another method like hierarchical clustering, Principal Coordinate Analysis (PCoA), non Metric Multidimensional Scaling (nMDS), ...

The implicit methods are using Euclidean distance (PCA, RDA, k-means) or Chi-squared distance (CA, CCA) but you don’t have to calculate them directly.

After carefully choosing (implicitly or explicitly) a distance index, you have to decide if you want to

1. have a direct representations of the distance matrix (e.g. with heatmaps) → rarely done
2. create discrete groups based on these distances (clustering methods)
3. display it as a continuous gradient in reduced space (ordination methods).

It is also frequent to combine clustering methods and ordination methods that are really complementary.

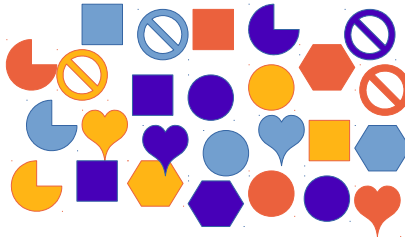
This can be done for both the columns (eg groups of species) and the row (eg groups of sites). One advantage of the ordination approaches is that they often allow you to represent on the same graph both the similarities between the rows and the columns of your dataset so that it is easier to interpret the similarities (eg : these

sites are similar because species 1 and Species 8 are more abundant there). The reduction of the number of variables is also particularly usefull if you want to visualise the changes in an other dimension. For example you often want to visualise how species communities have changed accross time. Ordination methods allow you to represent the whole community at one site by one point on the graph and to show the changes over time. The disadvantage of ordination methods is that 2 dimensions might not be sufficient to properly represent de full space. For example you can have two points close together in one bidimensional plane but that are infact far away from each other in other dimentions. Clustering show you the relationships in all the dimensions at the same time.

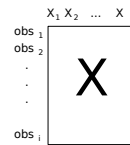
You can also to some extent have the best of both worlds by representing the groups produced by the clustering within the ordination plots and the links between each observation ... -

## Original data set

Complex, messy, many different characteristics (= multivariate) → difficult to "see" patterns, particularly with real dataset that are tables of numbers...



## Raw data



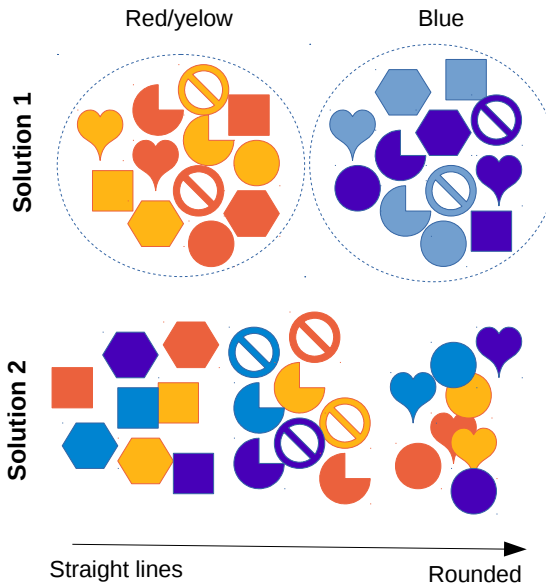
## Unsupervised methods

Reorganize the data to group the objects that are similar in discrete clusters or ordinate them along a gradient.

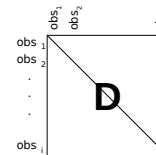
The definition of "similar" might be quite arbitrary and many solutions are possible.

Here are 2 different solutions : one is based on color, the other on shape. None is intrinsically better than the other. But some solutions might be more useful to you depending on what are your questions or your interest in the dataset.

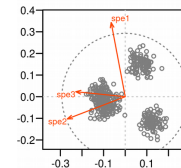
There is no real way to objectively evaluate how good is the result...



## Distance Matrix (on the observations or on the variables)



## Ordination (gradients)



## Clustering (groups)



## Supervised methods

Here you know the "real" answer you want and you try to build a model that will predict as best as possible this real answer.

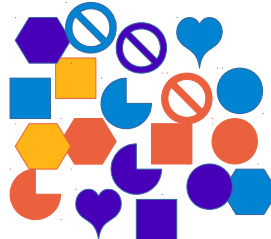
For example here you may want to discriminate the rounded yellow shapes from the others (vector Y called "responses") based on the characteristics you have measured on these objects (matrix X of predictors). You must then first tell the model which are rounded yellow shapes according to you ("labels").

Here you can evaluate the quality of the solution by comparing the values predicted to the "real" values (labels). (ideally on an independent dataset)

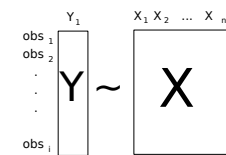
Predicted group of yellow rounded shapes



Predicted group of NOT yellow rounded shapes



1 prediction error/24  
→ 95.8 % of accuracy



Response Predictors/  
Descriptors

Qualitative → "Classification"  
Quantitative → "Regression"

Figure 1:

### 1.1.5 Overview of the methods

Here is a brief overview of the most frequent statistical methods with a tentative classification. See also figure 2

- Univariate unsupervised methods : descriptive statistics of one variable at a time
  - histogram and density distribution
  - average, standard deviation, quantiles
- Univariate supervised methods : relationship between one variable (the response) and one matrix of “predictors” or “explanatory variables”
  - GLMMs and extensions : Generalized Linear Mixed Models that are a generalization of student t test, ANOVA, linear regression, Logistic regression, G test (like Chi-squared test),...
  - Machine learning approaches : related or not to GLMs but heavily based on computer algorithms and resampling techniques (cross validation,...) : GAMMs, MARS, CART (Classification and Regression Trees), Random Forest, BRT, XGboost, SVM, K-NN, Neural Networks,...
  - Methods in the gray zone : univariate supervised methods with connection with unsupervised multivariate methods : PLS (Penalized least Squares), LDA (Linear Discriminant Analysis),...
- Multivariate Unsupervised methods : understand the structure of one matrix at a time
  - Distances used to create discrete groups of similar observations (or variables)
    - \* Implicit Euclidean distance : K-means Partitioning (non hierarchical method)
    - \* Explicit distance matrix :
      - Hierarchical Clustering (with different possible grouping algorithms : Ward, UPGMA,...)
      - Partitioning around medoids (non hierarchical method)
    - \* Many other algorithms with implicit or explicit distance : DBSCAN, fuzzy c-means, model based clustering, SOMs, Hierarchical divisive clustering,...
  - Ordination in reduced space : distances used to represent gradients of differences and/or to simplify the dataset (reduce the number of dimensions)
    - \* Implicit distance matrix
      - Euclidean distance : PCA = Principal Component Analysis
      - Chi squared distance : CA = Correspondance Analysis
      - Chi squared, Chord or Hellinger distance : tbPCA = transformation based Principal Component Analysis
    - \* Explicit distance matrix :
      - MDS = Metric Dimensional Scaling MDS (also called PCoA = Principal Coordinate Analysis)
      - nMDS = non Metric Multidimensional Scaling
- Multivariate Supervised methods : relationship between 2 or more matrices
  - Regression like methods = “canonical” or “constrained” ordinations : one matrix (X) “explains” or “predict” an other one (Y)
    - \* RDA & tbRDA : Redundancy Analysis and transformation based RDA : canonical extension of the PCA and tbPCA
    - \* dbRDA = CAP : (distance based) RDA on the axes of a PCoA (MDS) based on any distance matrix. CAP = Constrained Analysis of Principal Coordinates
    - \* CCA : Canonical Correspondance Analysis = extension of CA
    - \* MRT : Multivariate regression trees (very different approach than ordinations)
  - Multivariate statistical tests
    - \* Mantel test : correlation between two distance matrices
    - \* MANOVA, ANOSIM, ADONIS,... : multivariate comparison between 2 or more groups

## Univariate unsupervised

Y<sub>1</sub>  
obs<sub>1</sub>  
obs<sub>2</sub>  
.  
.  
obs<sub>i</sub>

### One variable considered at a time

--> descriptive statistics/graphs :  
mean, median, standard deviation,  
histogram, density,...

## Multivariate unsupervised

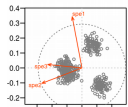
Y<sub>1</sub> Y<sub>2</sub> ... Y<sub>n</sub>  
obs<sub>1</sub>  
obs<sub>2</sub>  
.  
.  
obs<sub>i</sub>

### One matrix of data considered at a time

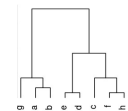
→ how to reduce this complexity and add  
some order to detect patterns ?

obs<sub>1</sub> obs<sub>2</sub> ... obs<sub>i</sub>  
obs<sub>1</sub> obs<sub>2</sub> ... obs<sub>i</sub>  
.  
.  
obs<sub>i</sub>

We are generally interested by the  
similarities/disimilarities between the lines  
or the columns of the matrix  
→ **distance matrix**



**Ordinations/dimensionality reduction :**  
represent a maximum of variability in  
few dimensions  
(tb)PCA - CA - MDS (=PCoA) - nMDS



**Clustering :**  
divide the data in discrete groups of  
similar objects or variables  
*Hierarchical Clustering* (→ dendrograms)  
*non hierarchical* (K-means, PAM,...), ...

## Univariate supervised

Y<sub>1</sub> X<sub>1</sub> X<sub>2</sub> ... X<sub>n</sub>  
obs<sub>1</sub>  
obs<sub>2</sub>  
.  
.  
obs<sub>i</sub>

We want to **predict/explain one unique variable Y** (response) as a function of one or  
several predictors/explanatory variables X

### **Regression et extensions :**

GLMs : including regression, ANOVA, ANCOVA, t-test, G test,  
logistic regression, ...  
+ GLMMs Generalized Linear Mixed Models

### **Machine learning / algorithmic approaches**

related or not to GLMs but heavily based on computer algorithms  
and resampling techniques (cross validation,...)  
*ElasticNet, GAMMs, MARS, CART (regression trees), Random  
Forest, BRT, SVM, Neural Net, K-NN,...*

### **Methods in the gray zone :**

methods connected with unsupervised multivariate methods :  
*PLS , LDA*

## Multivariate supervised

Y<sub>1</sub> Y<sub>2</sub> ... Y<sub>n</sub> X<sub>1</sub> X<sub>2</sub> ... X<sub>n</sub>  
obs<sub>1</sub>  
obs<sub>2</sub>  
.  
.  
obs<sub>i</sub>

### Link between two or more matrices

**"Canonical" or "Constrained" Ordinations**  
(regression like) : CCA, RDA, dbRDA (CAP)

### **MRT : Multivariate Regression Trees**

**Multivariate "tests" :**  
*Mantel, MANOVA, ANOSIM, ADONIS,...*

Figure 2:

## 1.2 Simple graphical vizualisation of complex datasets

Before attacking your data with more complex statistical method (PCA, clustering) an essential and easy step is to first summarize your datasets and to perform quick graphical explorations.

R offers very powerful yet easy to use tools for this purpose.

### 1.2.1 Descriptive statistics as summary of the data

Several functions allows you to compute more or less sofisticated descriptive statistics on your dataset. After importing a dataset you should systematically compute one of these to check that the data have been imported as intended and that there is no obvious encoding errors or abnormal values.

My prefered function is by far `summary` but many people prefer the more compact (but less informative in my opinion) output of `str`. Two other functions from 2 packages provide much more detailed descriptions.

```
library(vegan)
data(mite.env)
summary(mite.env)
```

```
##      SubsDens      WatrCont      Substrate  Shrub      Topo
##  Min.   :21.17  Min.   :134.1  Sphagn1 :25  None:19  Blanket:44
##  1st Qu.:30.01  1st Qu.:314.1  Sphagn2 :11  Few :26  Hummock:26
##  Median :36.38  Median :398.5  Sphagn3 : 1  Many:25
##  Mean   :39.28  Mean   :410.6  Sphagn4 : 2
##  3rd Qu.:46.81  3rd Qu.:492.8  Litter  : 2
##  Max.   :80.59  Max.   :827.0  Barepeat : 2
##                               Interface:27
```

```
str(mite.env)
```

```
## 'data.frame': 70 obs. of 5 variables:
## $ SubsDens : num 39.2 55 46.1 48.2 23.6 ...
## $ WatrCont : num 350 435 372 360 204 ...
## $ Substrate: Factor w/ 7 levels "Sphagn1","Sphagn2",...: 1 5 7 1 1 1 1 7 5 1 ...
## $ Shrub : Ord.factor w/ 3 levels "None"<"Few"<"Many": 2 2 2 2 2 2 2 3 3 3 ...
## $ Topo : Factor w/ 2 levels "Blanket","Hummock": 2 2 2 2 2 2 2 1 1 2 ...
```

```
Hmisc::describe(mite.env)
```

```
## mite.env
##
## 5 Variables      70 Observations
## -----
## SubsDens
##      n missing distinct      Info      Mean      Gmd      .05      .10      .25      .50
##      70      0       69         1    39.28    13.22    24.58    26.58    30.01    36.38
##      .75      .90      .95
##    46.81    56.67    60.75
##
## lowest : 21.17 22.36 22.90 23.55 25.84, highest: 59.93 61.43 62.38 64.75 80.59
## -----
## WatrCont
##      n missing distinct      Info      Mean      Gmd      .05      .10      .25      .50
```



```
##      70      0      70      1    410.6    160.9    184.9    237.6    314.1    398.5
##      .75     .90     .95
##    492.8    592.4    646.6
##
## lowest : 134.13 145.28 145.68 184.04 185.89, highest: 634.75 656.35 691.79 708.16 826.96
## -----
## Substrate
##      n missing distinct
##      70      0        7
##
## Value      Sphagn1  Sphagn2  Sphagn3  Sphagn4  Litter  Barepeat Interface
## Frequency      25      11       1       2       2       2       27
## Proportion    0.357    0.157    0.014    0.029    0.029    0.029    0.386
## -----
## Shrub
##      n missing distinct
##      70      0        3
##
## Value      None  Few  Many
## Frequency      19   26   25
## Proportion 0.271 0.371 0.357
## -----
## Topo
##      n missing distinct
##      70      0        2
##
## Value      Blanket Hummock
## Frequency      44     26
## Proportion    0.629    0.371
## -----
```

```
skimr::skim(mite.env)
```

```
# more complex dataset with factors (ordered or not) and missing value
# (not run here)
library(mlbench)
data(Soybean)
summary(Soybean)
str(Soybean)
skimr::skim(Soybean)
Hmisc::describe(Soybean)
```

The function `str` is particularly useful to explore the results of a statistical analysis and extract exactly the part of the results you need. For example here we perform a Principal Component Analysis on the iris dataset (first 4 columns). Then we look at the structure of the object and we extract the loadings of the original variables from a slot called here “rotation”.

```
pca <- prcomp(iris[,1:4])
str(pca)
```

```
## List of 5
## $ sdev      : num [1:4] 2.056 0.493 0.28 0.154
## $ rotation: num [1:4, 1:4] 0.3614 -0.0845 0.8567 0.3583 -0.6566 ...
```

```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
##   .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
##   $ center   : Named num [1:4] 5.84 3.06 3.76 1.2
##   ..- attr(*, "names")= chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
##   $ scale     : logi FALSE
##   $ x         : num [1:150, 1:4] -2.68 -2.71 -2.89 -2.75 -2.73 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
##   - attr(*, "class")= chr "prcomp"
```

```
pca$rotation
```

```
##               PC1           PC2           PC3           PC4
## Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
## Sepal.Width   -0.08452251 -0.73016143 -0.59791083 -0.3197231
## Petal.Length   0.85667061  0.17337266 -0.07623608 -0.4798390
## Petal.Width    0.35828920  0.07548102 -0.54583143  0.7536574
```

### 1.2.2 Visualise the distribution of the variables

It is often useful to have an idea of how the values of the different variables are distributed. Histogram or density plots are an easy and good tool for this. Note that some functions and graphs presented below (eg SPLOMs) will compute such graphs automatically for you along with other graphs. The manual creation of histograms might however be useful when you have many variables.

Here is a simple example for the mite dataset (abundance of mites)

```
data(mite, package = "vegan")
dim(mite)
```

```
## [1] 70 35
```

```
# reorder the columns of the dataset to have the most abundant species in the
# first columns
```

```
mite <- mite[,order(-colSums(mite))]
```

```
# dev.new(width = 18/2.54, height = 18/2.54)
# Plot an histogram of the 16 most abundant species on the log scale
par(mfrow = c(4,4), mar = c(3.5,3.5,1,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 1)
for (i in 1:16) {
  hist(log10(mite[,i]+1), breaks = 15, freq = TRUE, main = colnames(mite[i]))
}
```

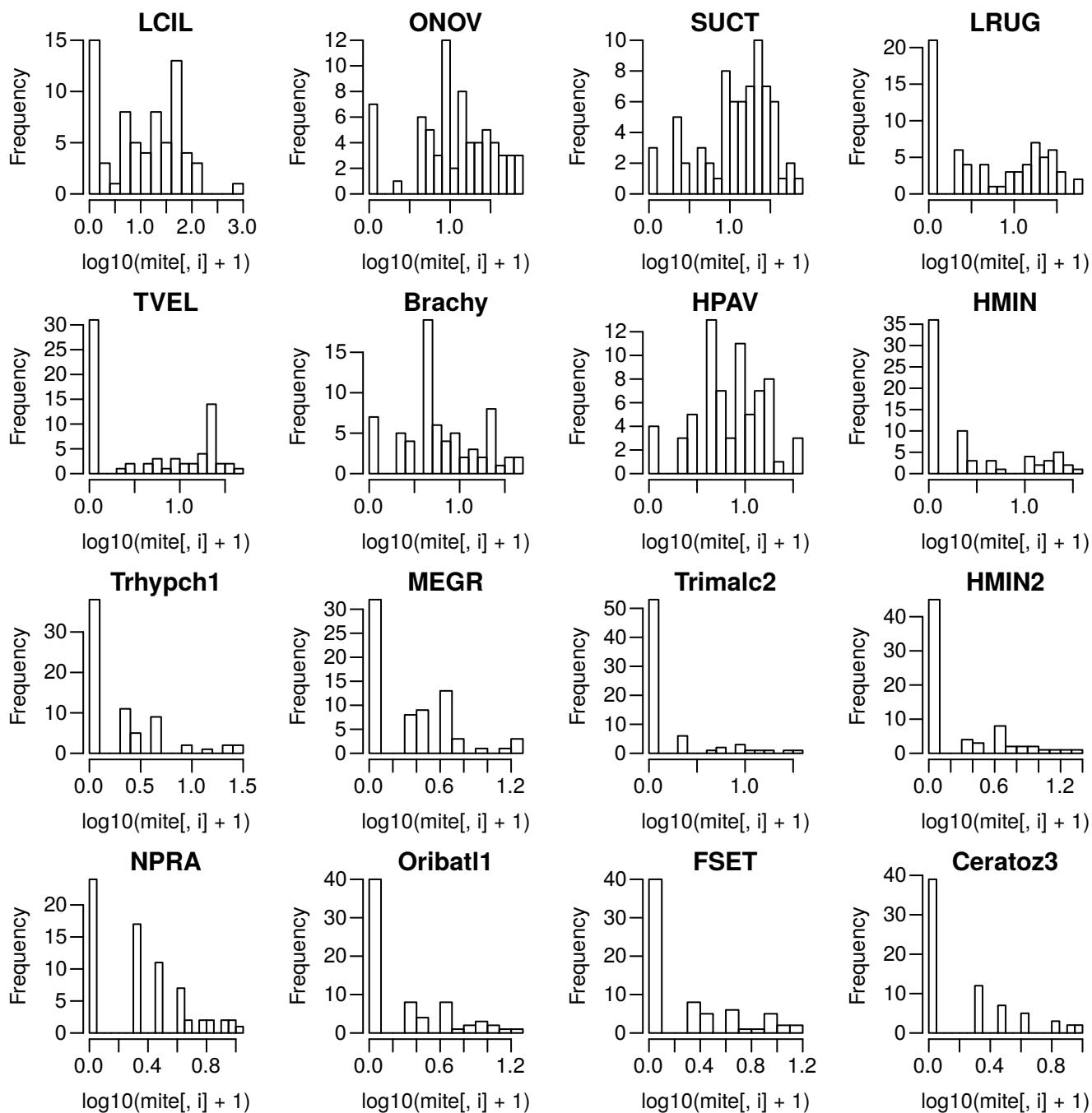


Figure 3:

Here is a more complex example with the iris data. The aim here is to have a joint vision of the distribution of the morphological variables of all iris species together but at the same time of each species separated. Density plots are ideal for this because you can easily superpose them. . . Even if the mathematics behind density plots involves complex probabilistic notions (probability density function), their interpretation is straightforward : the higher the lien, the higher the number of observations in this range of values

```
# dev.new(width = 18/2.54, height = 6/2.54)
par(mfrow = c(1,4), mar = c(3,2,3,0.5), mgp = c(1.8, 0.6, 0))
d <- iris
for (i in 1:4) {
  hist(d[,i], breaks = 20, col = "grey90", border = 'grey60',
       freq = FALSE, las=1, main = "",
       xlab = colnames(d)[i], ylab = "")
  lines(density(d[,i], adjust = 0.5), col = "grey40", lwd = 2, lty = 3)

  dens1 <- density(d[d$Species == "setosa",i], adjust = 1)
  dens2 <- density(d[d$Species == "versicolor",i], adjust = 1)
  dens3 <- density(d[d$Species == "virginica",i], adjust = 1)

  lines(dens1$x, dens1$y/3, col = "dodgerblue", lwd = 2)
  lines(dens2$x, dens2$y/3, col = "orangered", lwd = 2)
  lines(dens3$x, dens3$y/3, col = "gold", lwd = 2)

  if(i == 2) {
    legend("topleft", inset = -0.25, xpd = NA, ncol = 4, bty = "n",
         lwd = 2, lty= c(1,1,1,3),
         col = c("dodgerblue","orangered", "gold", "grey40"),
         legend = c("setosa", "versicolor", "virginica", "All species") )
  }
}
```

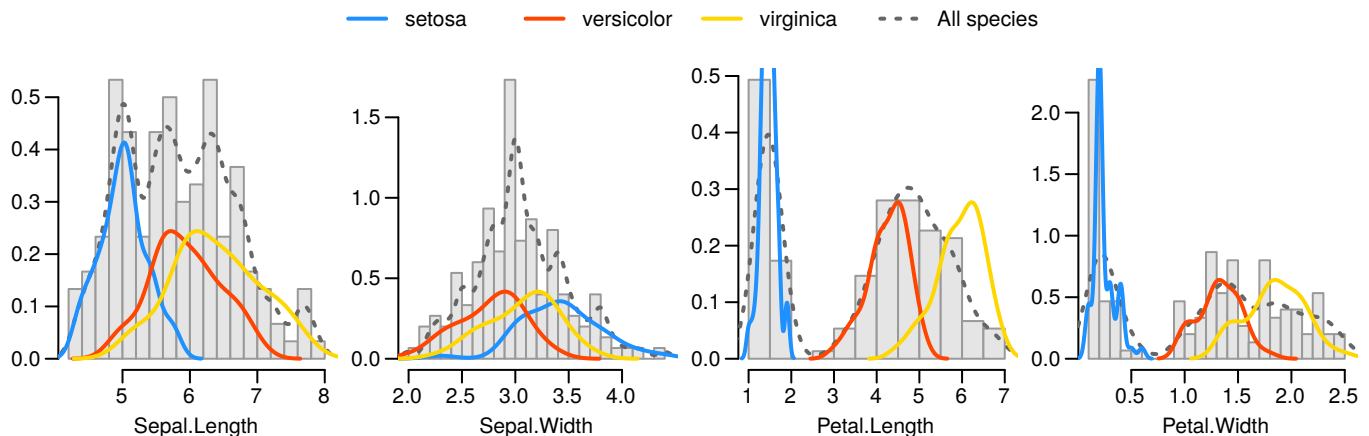


Figure 4:

### 1.2.3 Faceting

Faceting consist in creating sub graphs for subset of the data (categorical variables or quantitative variables). You can do this manually with loops as we did in the previous section but several R graphical packages simplify this process a lot (typically : `ggplot2` or `lattice`).

The big advantage of these packages is that it is much more easy to change completely the visualisation of your dataset

```
# You have to change the organisation of the dataset from a "wide format"
# to a "long format"
d <- iris
d <- reshape2::melt(iris, id = "Species")
head(d)
```

```
## Species    variable value
## 1  setosa Sepal.Length  5.1
## 2  setosa Sepal.Length  4.9
## 3  setosa Sepal.Length  4.7
## 4  setosa Sepal.Length  4.6
## 5  setosa Sepal.Length  5.0
## 6  setosa Sepal.Length  5.4
```

Now you can easily compare the 3 species for each morphological measure :

```
# dev.new(width = 18/2.54, height = 6/2.54)
ggplot(d, aes(y = value, x = Species)) +
  geom_boxplot(color = "dodgerblue") +
  geom_point(color = "gray40", alpha = 0.25, position = position_jitter(width = 0.1)) +
  facet_wrap(~variable, scales = "free", nrow = 1) +
  xlab("")
```

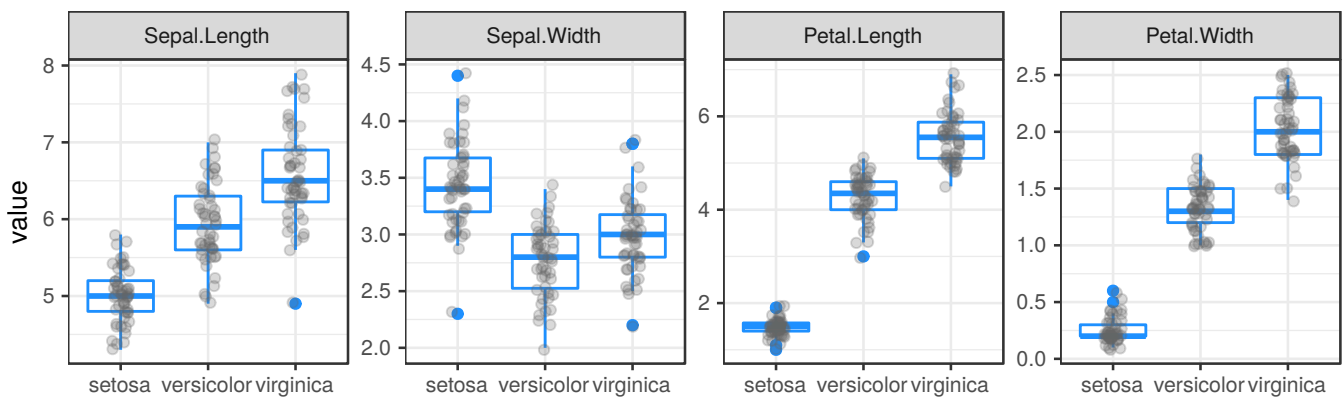


Figure 5:

With a very limited change in the code, you can easily change completely the representation. Here you compare the 4 morphological measures for each species. The interest is limited here but in many situation these faceting facilities are really helpfull to explore graphically the data.

```
# dev.new(width = 16/2.54, height = 8/2.54)
ggplot(d, aes(y = value, x = variable)) +
  geom_boxplot(color = "dodgerblue") +
  geom_point(color = "gray40", alpha = 0.25, position = position_jitter(width = 0.1)) +
  facet_wrap(~Species, scales = "free", nrow = 1) +
  xlab("") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

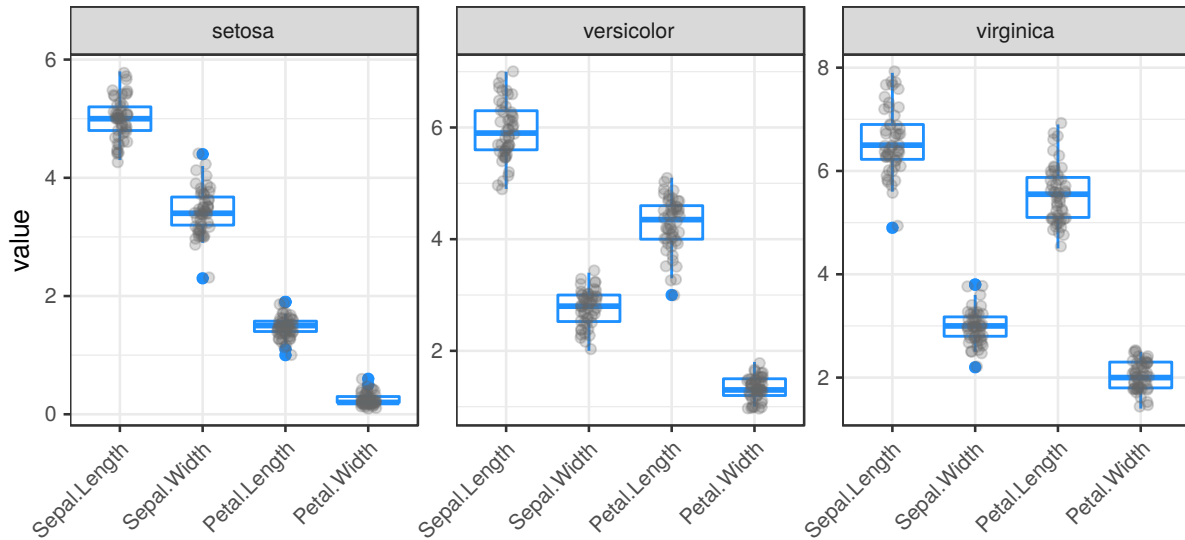


Figure 6:

### 1.2.4 SPLOMs : Scatterplot Matrices

A SPLOM is a graph in which each subgraph is a scatterplot of each pair of the variables of the dataset. The basic function to perform this in R is `pairs`. This function is highly customizable and allows you to add almost everything you want in each subgraph. Here is my own version (there are plenty other available in packages or on the web). The diagonal contains information about the distribution of each variable (histogram + density). The lower panel provides the correlation coefficient with a font size proportional to the correlation value to improve readability. The upper panel contains the scatterplots with a regression line and a smoother (loess) to visualize the trends.

These graphs are extremely useful and should be done almost systematically !! Their main disadvantage is that you are limited on the number of variable that you can plot (~20 variables) to keep your graph readable. Heatmaps of the correlation matrix are probably more adapted for more variable but they are less informative because they show you only a summary statistic (the correlation coefficient) and not a scatter plot of the real relationship. Here we transform the qualitative variables into dummy variables → the correlation coefficient including species can be interpreted.

```
# dev.new(width = 16/2.54, height = 11/2.54)
pairs2(iris, dummy = TRUE)
```

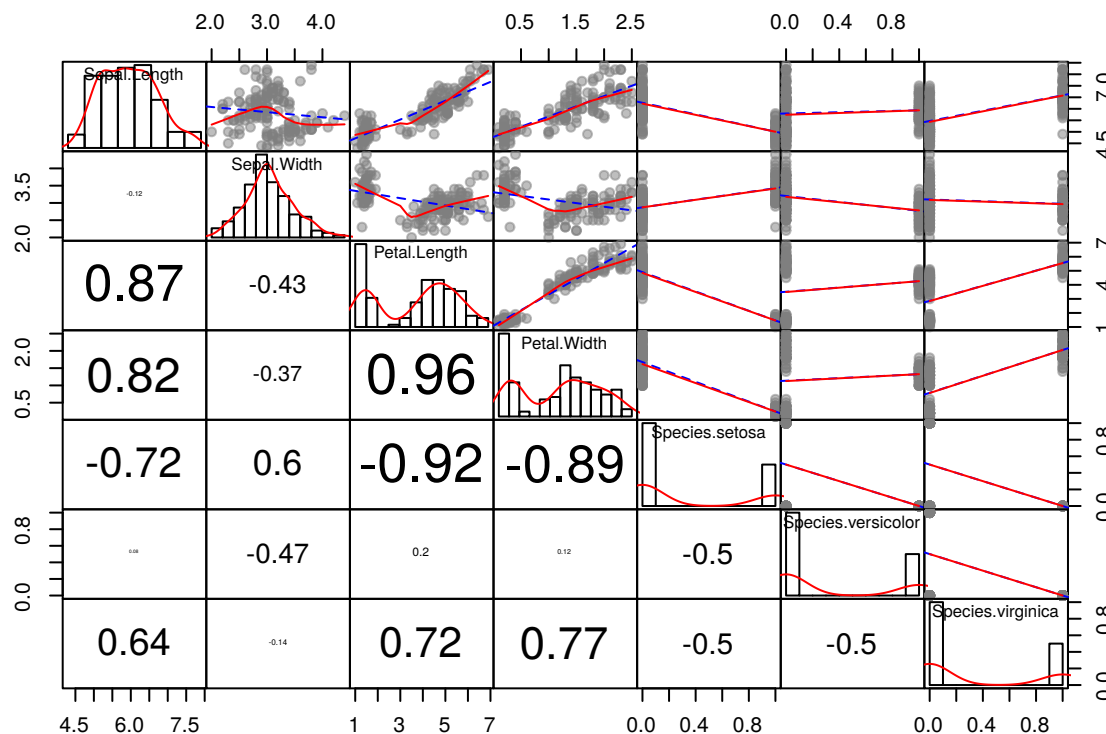


Figure 7:

Here the variable are reordered to group the more correlated variables together (`reorder = TRUE`), we compute the Spearman correlation (instead of the default pearson). We also map the color and shape of the point to the Species variable and add a legend.

```
# dev.new(width = 15/2.54, height = 10/2.54)
mycols <- c("forestgreen", "gold", "dodgerblue")
pairs2(iris[,1:4], Rmethod = "spearman", reorder = TRUE,
       pt.cex = 0.9, oma=c(3,3,5,3), # outer margins
       col = mycols[as.numeric(iris$Species)], pch = c(1:3)[as.numeric(iris$Species)])
legend("top", col = mycols, legend = levels(iris$Species), pch = 1:3,
      xpd = NA, ncol = 3, bty = "n", inset = -0.03, pt.cex = 1)
```

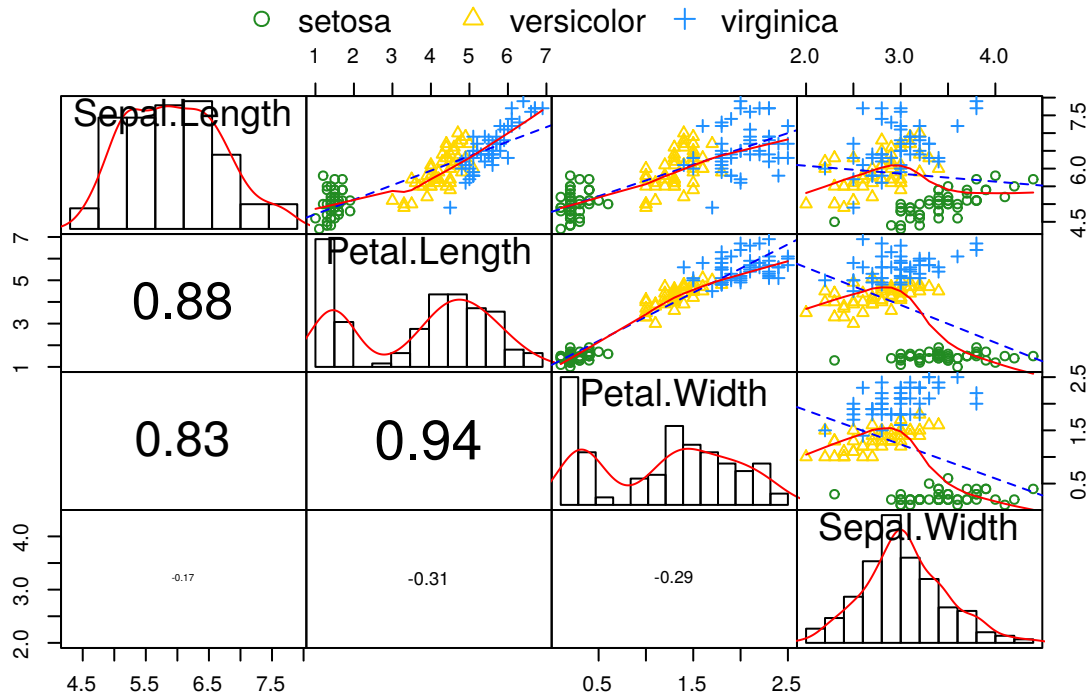


Figure 8:



The package **GGally** provides several functions to perform complex SPLOMs based on **ggplot**. The output is more complex and produces different results depending on the type of data (qualitative, quantitative,...) but the function is very slow...

```
# dev.new(width = 16/2.54, height = 12/2.54)
library(GGally)
ggpairs(iris, aes(color = Species))
```

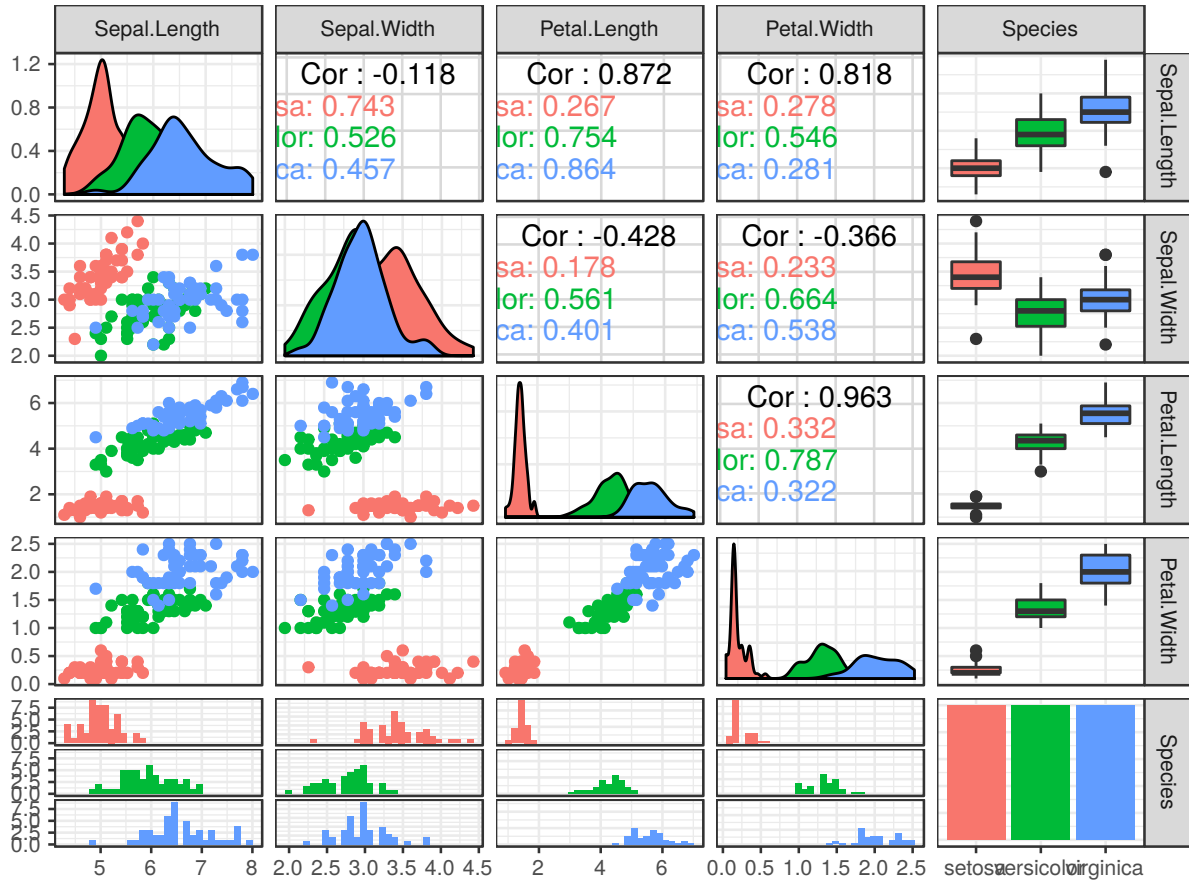


Figure 9:

ggscatmat uses only numeric values and is a little bit faster

```
# dev.new(width = 16/2.54, height = 10/2.54)
ggscatmat(iris, color = "Species") + theme_bw()
```

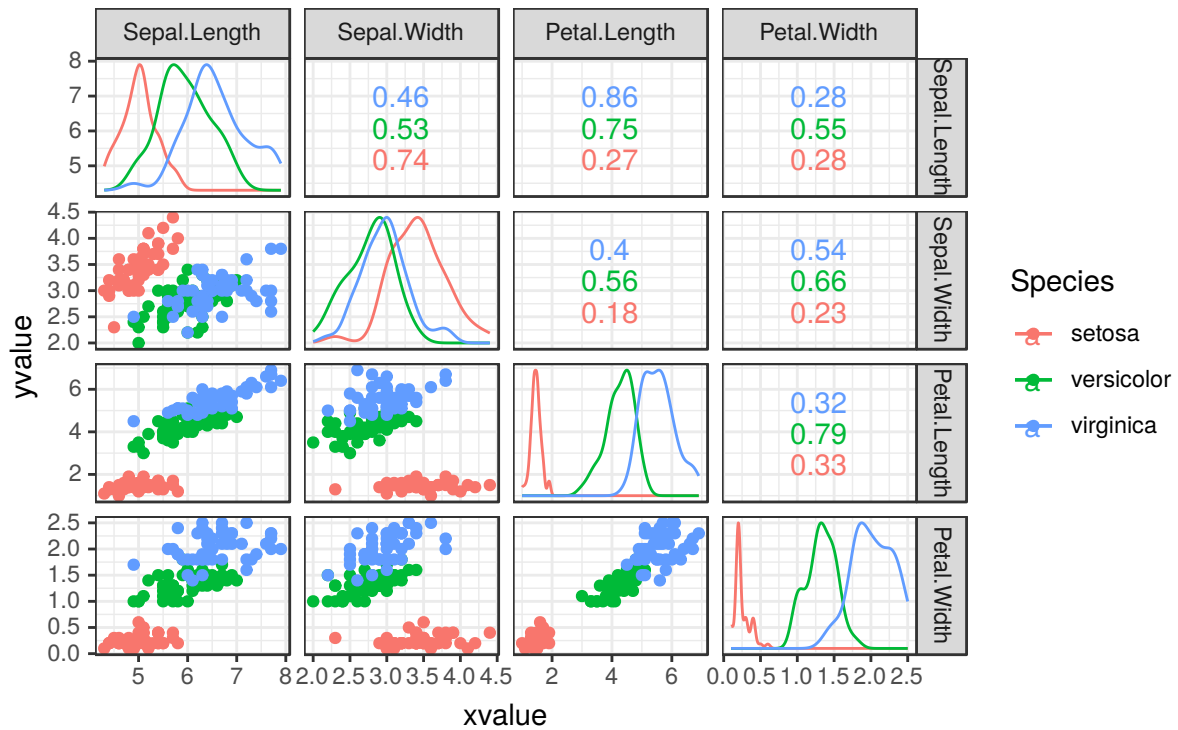


Figure 10:

### 1.2.5 Heatmap of the correlation matrix

Here we use a huge dataset of meteorological data, land use and soil data of a 1375 5x5 km<sup>2</sup> grid of Belgium (UTM grid). With the 157 variables of this dataset it is impossible to make a SPLOM. A heatmap of the correlation matrix is a good way to explore such huge dataset to see the redundancies between the variables.

```
d <- read.csv2("data/UTM5/UTM5data.csv")
# summary(d)
# colnames(d)

# dev.new(width = 18/2.54, height = 18/2.54)
corheatmap(d[,-(1:10)], FALSE, cexRow = 0.6, cexCol = 0.7)
```

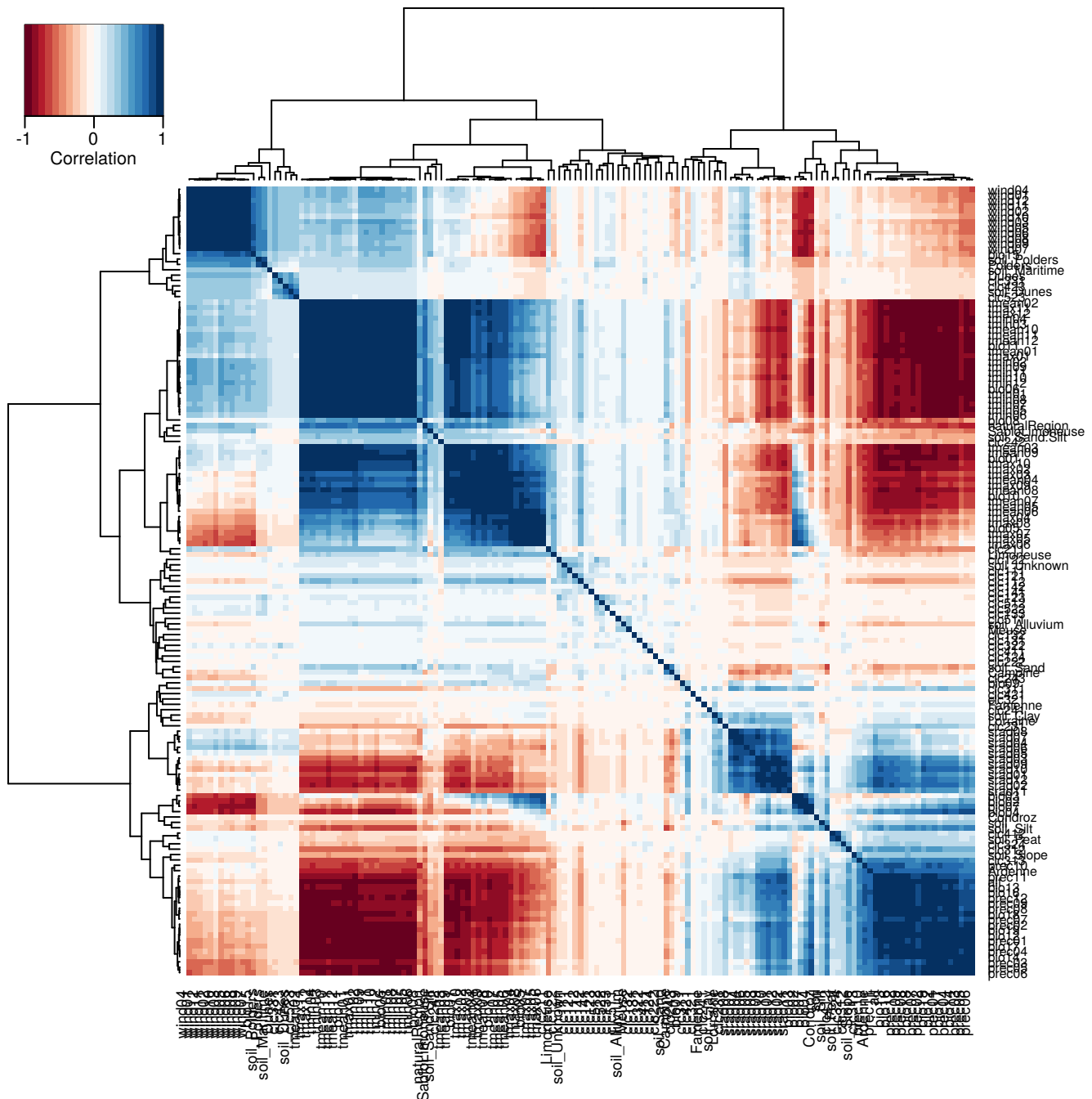


Figure 11:

```
# dev.new(width = 18/2.54, height = 18/2.54)
d2 <- d[, which(colnames(d) == "bio01") : which(colnames(d) == "wind12")]
# corheatmap(d2, R = FALSE, cexRow = 0.6, cexCol = 0.7)
corheatmap(d2, R = FALSE, breaks = c(-1, -0.8, -0.6, 0.6, 0.8, 1), cexRow = 0.6, cexCol = 0.7)
```



Here we add the value of the correlation coefficient (x100) but this is not really necessary.

```
# dev.new(width = 14/2.54, height = 14/2.54)
d2 <- d[, which(colnames(d) == "bio01") : which(colnames(d) == "bio19")]
corheatmap(d2,T, 0.7)
```

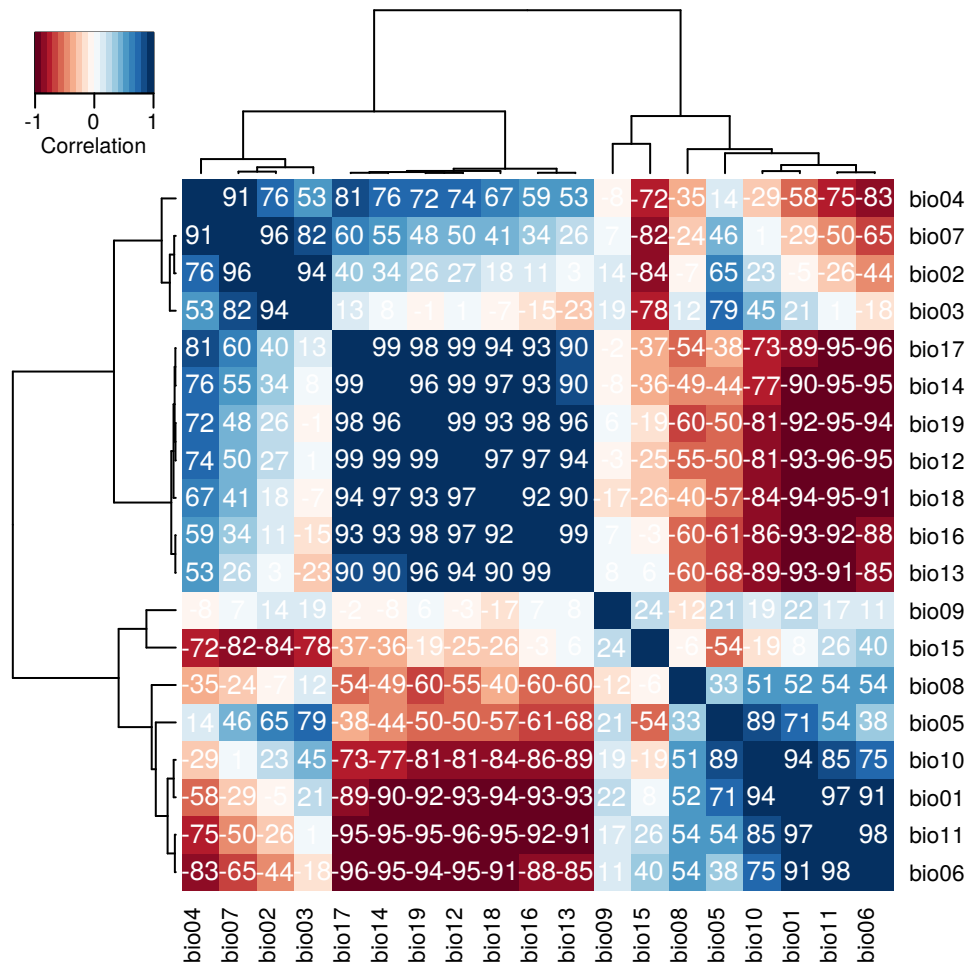


Figure 13:

```
## dev.new(width = 14/2.54, height = 14/2.54)
# d2 <- d[, which(colnames(d) == "clc111") : which(colnames(d) == "clc523")]
# corheatmap(sqrt(d2),T, 0.5)
#
## dev.new(width = 18/2.54, height = 14/2.54)
# d2 <- d[, which(colnames(d) == "tmax01") : which(colnames(d) == "tmin12")]
# corheatmap(d2,F, 0.5)
```

Note that the scatterplot matrix is often more informative than a simple correlation coefficient even with the heatmap.

Here you can see the complex non linear relationship between the winter and summer temperatures in different Belgian regions

```
# dev.new(width = 18/2.54, height = 12/2.54)
d2 <- d[, which(colnames(d) == "tmax01") : which(colnames(d) == "tmin12")]
d2 <- cbind(d$alt, d2[,c( "tmin01", "tmin02", "tmin03", "tmin04",
                        "tmax04", "tmax05", "tmax06", "tmax07")])
pairs2(d2, reorder = FALSE, pt.cex = 0.5)
```

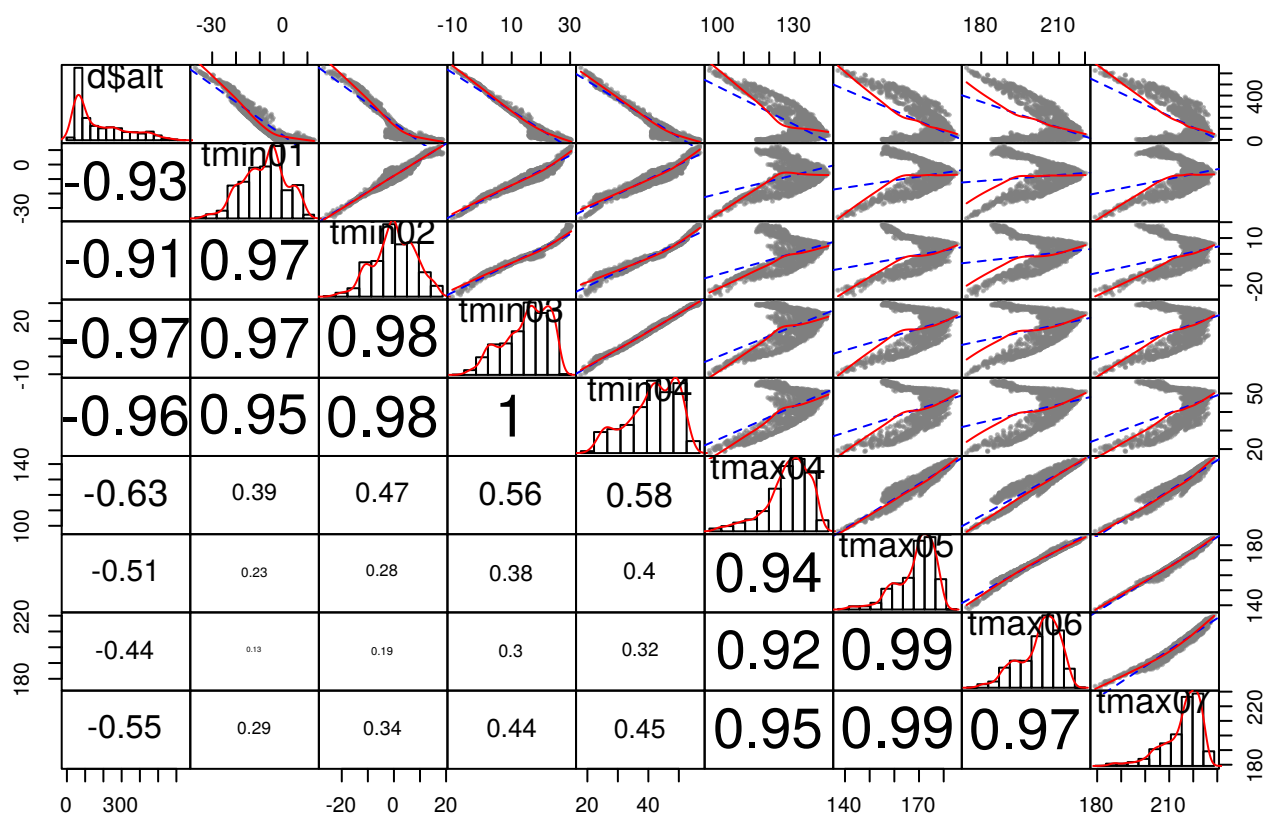


Figure 14:

`corrplot` is a popular package for the visualisation of correlation matrices. Have look at the vignette of the package to have a full view of the many possibilities offered by the package. Keep in mind that simplicity is often better ... There are also lots of example to add p-values information on the graphs. This is generally not usefull (unless you have a very small dataset, the size of the correlation is more important...)

```
d2 <- d[, which(colnames(d) == "bio01") : which(colnames(d) == "bio19")]
corrplot::corrplot(cor(d2), method = "circle", order = "hclust", asrect = 4)
```

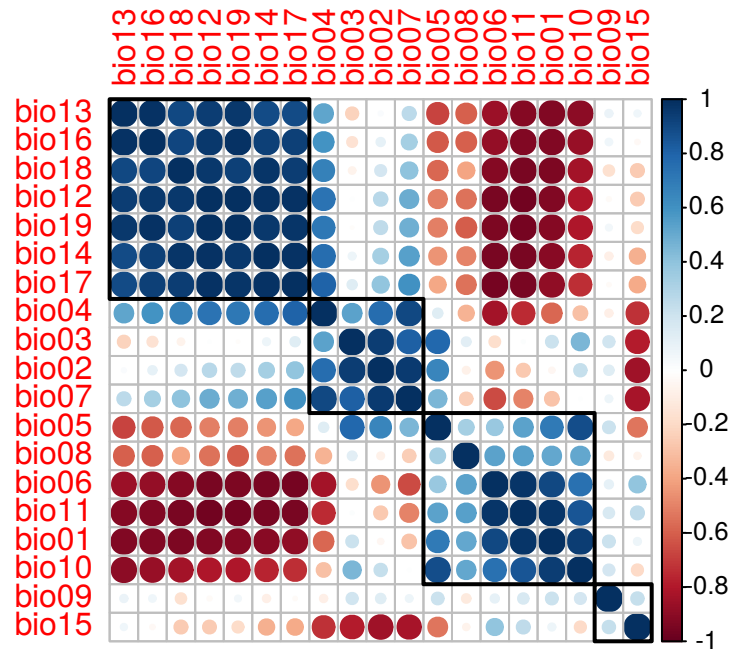


Figure 15:

### 1.2.6 Parallel Coordinate plots

A specific type of plot dedicated to the visualisation of several numeric variables at the same time. Each axis represent one variable and has been normalized to have comparable scales between variables.

```
# dev.new(width = 12/2.54, height = 8/2.54)
mycols <- c("forestgreen", "gold", "dodgerblue")
par(mar = c(3.5,2.5,3,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 1)
MASS::parcoord(iris[,1:4], col = mycols[iris$Species])
legend("top", xpd = NA, inset = -0.1, bty = "n", ncol = 3, cex = 1.1, col = NULL,
      fill = mycols, legend = levels(iris$Species))
```

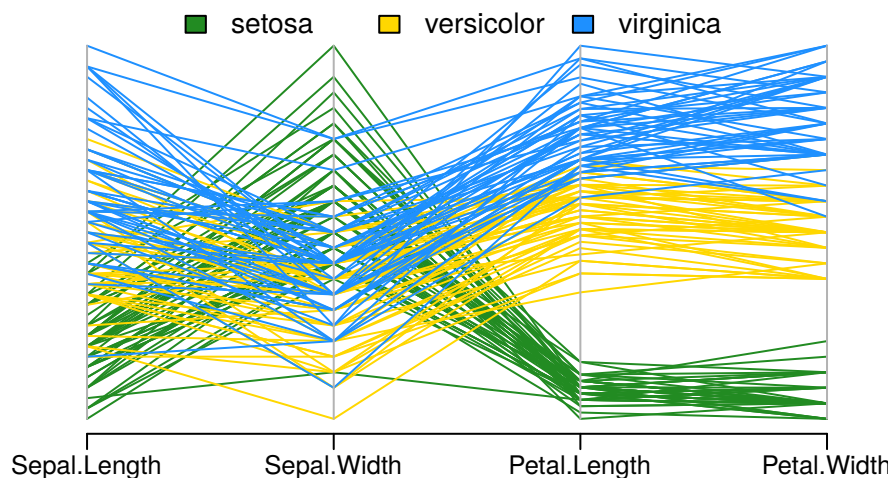


Figure 16:

The problem with this kind of graphs is that they become quickly overcrowded and difficult to read. Here is an example with bioclimatic variables from different regions of Belgium



```

d <- read.csv2("data/UTM5/UTM5data.csv")
d2 <- d[, c( which(colnames(d) == "naturalRegion"),
             which(colnames(d) == "bio01") : which(colnames(d) == "bio19"))]
set.seed(123)
d2 <- d2[sample(1:nrow(d2), size = 500),] # random sample of 500 grids

d2[,1] <- as.numeric(d2[,1])

# Colors
mycols <- RColorBrewer::brewer.pal(10,"Set3")
mycols_trans <- adjustcolor(mycols, 0.25) # add transparency

# dev.new(width = 18/2.54, height = 12/2.54)
par(mfrow = c(1,1), mar = c(3.5,5.5,1,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 1)
MASS::parcoord(d2, col = mycols_trans[d2$naturalRegion], var.label = TRUE)
mtext(text = levels(d$naturalRegion), side = 2, line = -1, cex = 0.8,
      at = seq(0,1, length.out = length(levels(d$naturalRegion))),
      col = mycols)

```

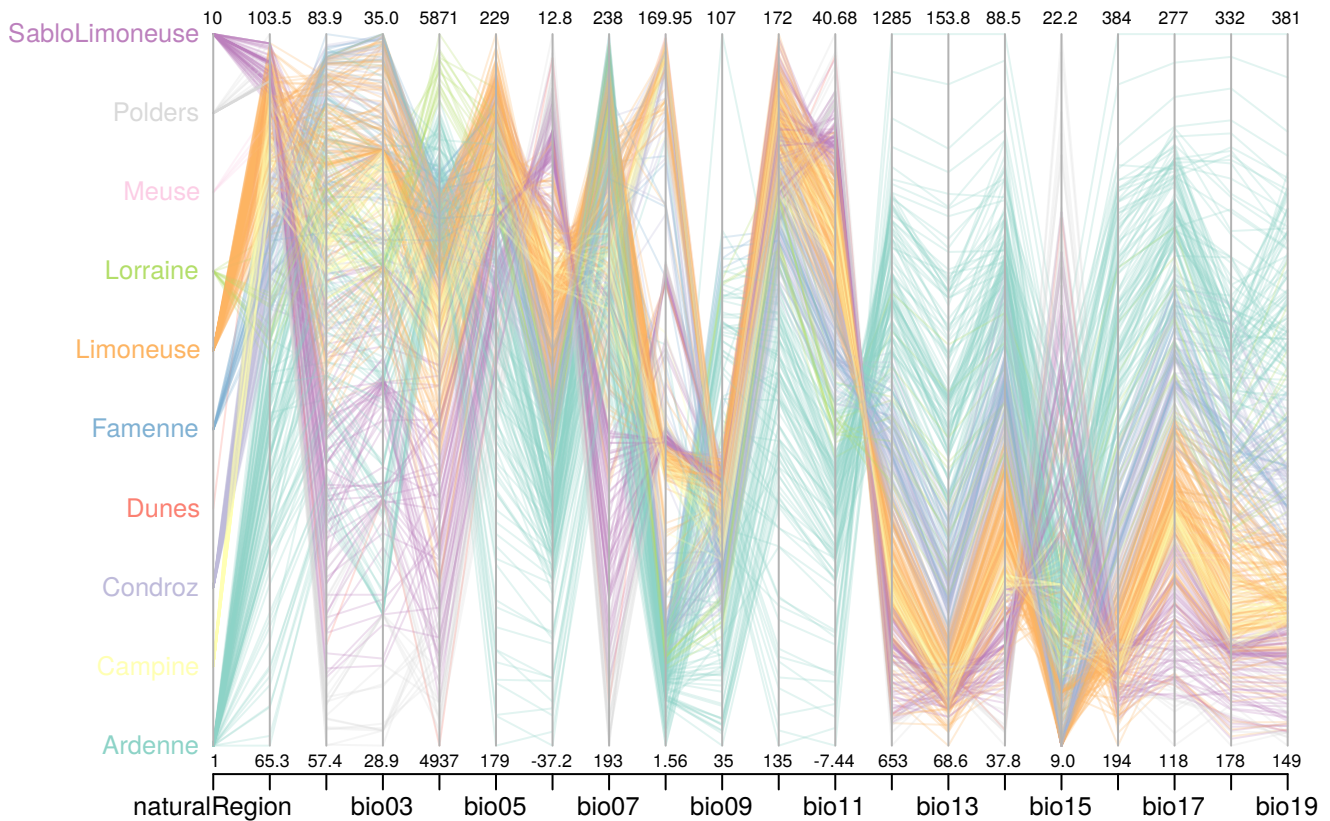


Figure 17:



### 1.2.7 SPLOMs between 2 matrices

In classical SPLOMs you make a graph of each pair of variable within a given matrix of data. The idea here is to make a scatterplot between each pairs of variables in 2 different matrices. This is typically usefull for datasets suited for multivariaty supervised methods (RDA, CCA). In the following example, we use a site x species of mites dataset and a site x 5 environmental characteristics dataset. For each of the 8 most abundant mite species, we plot a graph of the species abundance vs each of the environmental variables.

```
library(GGally)
library(vegan)
data(mite)
data(mite.env)
data(mite.xy)

# 8 most comon species of mites
mite <- mite[,order(-colSums(mite))[1:8]]

# You need to group the 2 matrices (species and environment)
# and to provide the number of the columns of each group you want to plot.
spec <- 1:ncol(mite)
env <- (1:ncol(mite.env)) + ncol(mite)
d <- cbind(mite, mite.env)

# dev.new(width = 18/2.54, height = 18/2.54)
ggduo(d, env, spec,
      types = list(continuous = wrap("smooth_loess", size = 0.5))) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

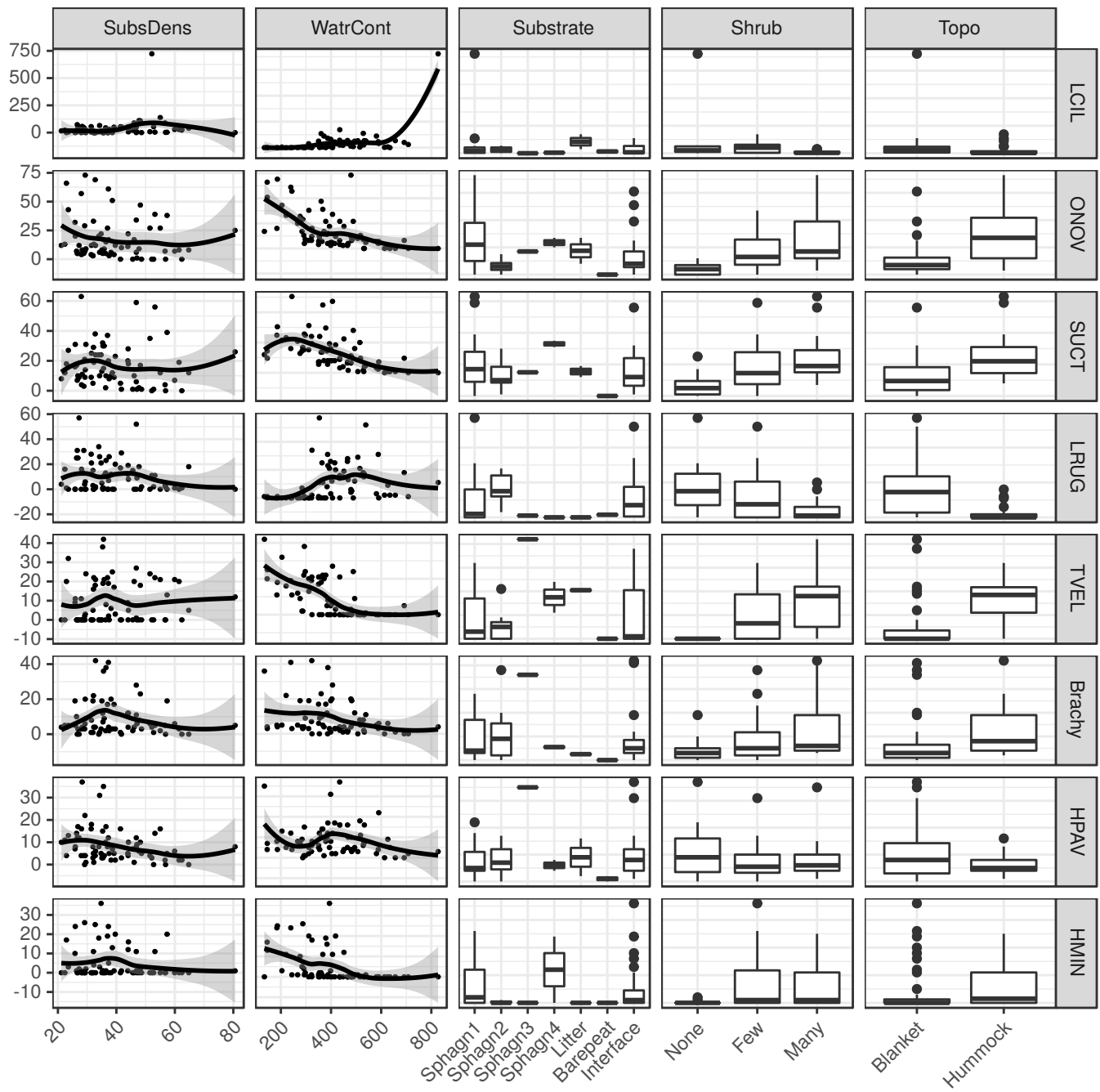


Figure 18: