

Example of Supervised, unsupervised, multivariate and univariate approaches

Gilles San Martin

03 September 2018

Contents

1	Introduction	2
1.1	Univariate unsupervised method : data exploration	2
1.2	Multivariate Unsupervised methods	4
1.2.1	Clustering : Hierarchical agglomerative clustering	4
1.2.2	Ordination : Principal Component Analysis	4
1.3	Multivariate Supervised Methods	7
1.4	Univariate Supervised methods	10

1 Introduction

A frequent error : use only unsupervised methods for questions requiring also supervised methods

```
n <- 100
set.seed(123)
x1 <- runif(n, 0, 1)
x2 <- runif(n, 0, 1)

Species <- factor(c("SpA", "SpB")[round(0.5*x1 + 0.5*x2 + rnorm(n,0,0.05))+1)])
summary(Species)
```

```
## SpA SpB
## 55 45
```

```
rbinom(1, 1, 0.5*x1 + 0.5*x2)
```

```
## [1] 1
```

```
d <- as.data.frame(matrix(rnorm(n*6, 0, 1), ncol = 6))
colnames(d) <- paste0("x", 3:8)
d$x3 <- d$x3 + c(0, 10, 20)
d$x4 <- d$x4 + c(0, 5, 10)
d$x5 <- d$x5 + c(0, 1, 2)
d$x6 <- d$x6 + c(-5, 5)
d$x7 <- d$x7 + c(-2, 2)
d$x9 <- 0.5*d$x8 + rnorm(n, 0, 0.5)
d$x10 <- d$x8 + rnorm(n, 0, 0.5)

d <- data.frame(Species, x1, x2, d)
d <- d[order(d$Species),]
row.names(d) <- paste0(1:n, "_", d$Species)
```

1.1 Univariate unsupervised method : data exploration

```
# dev.new(width = 7/2.54, height = 7/2.54)
par(mfrow = c(1,1), mar = c(2.5,2.5,2,1), mgp = c(1.6, 0.5, 0), cex = 0.8, las = 1)
plot(d$x1, d$x2, col = as.numeric(d$Species),
     pch = c(1,3)[as.numeric(d$Species)])
legend("top", xpd = NA, inset = -0.15, horiz = TRUE,
     pch = c(16, 17), col = 1:2, legend = levels(d$Species), bty = "n")
```

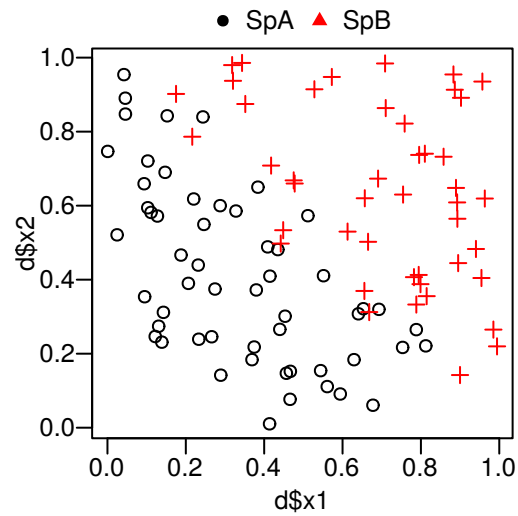


Figure 1:

```
# dev.new(width = 20/2.54, height = 14/2.54)
pairs2(d[, -1], pt.cex = 0.5,
       col = as.numeric(d$Species),
       pch = c(1, 3)[as.numeric(d$Species)])
```

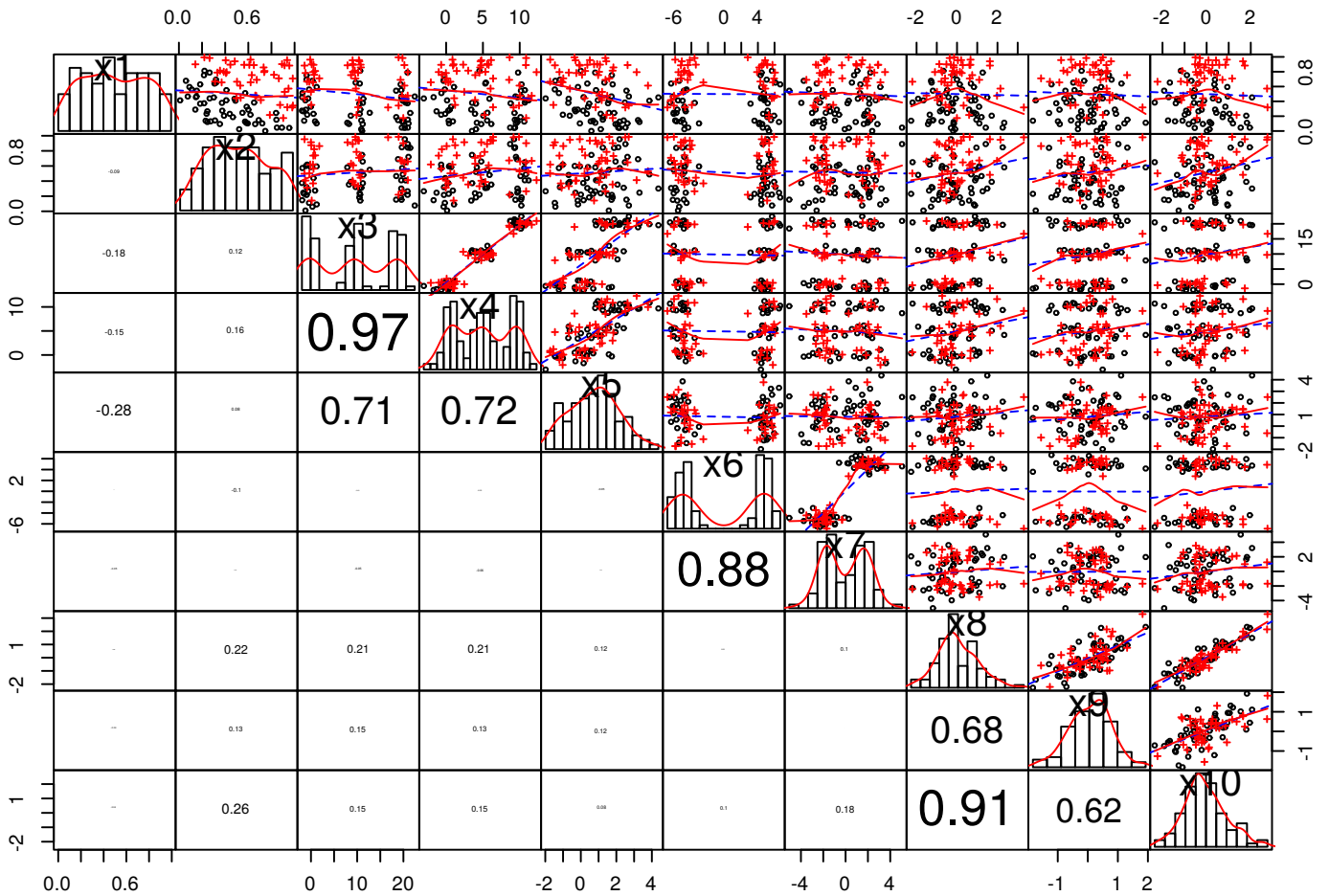


Figure 2:

1.2 Multivariate Unsupervised methods

1.2.1 Clustering : Hierarchical agglomerative clustering

```
hcl <- hclust(dist(scale(d[, -1])), method = "ward.D2")  
  
# dev.new(width = 18/2.54, height = 10/2.54)  
par(mfrow = c(1,1), mar = c(1.5,2.5,1,1), mgp = c(1.8, 0.6, 0), cex = 0.8, las = 1)  
plot(hcl, hang = -1, cex = 0.6)  
points(x = 1:n, y = rep(0,n),  
       col = as.numeric(d$Species)[hcl$order],  
       pch = c(16,17)[as.numeric(d$Species)][hcl$order])  
legend("topright", pch = c(16, 17), col = 1:2, legend = levels(d$Species), bty = "n")
```

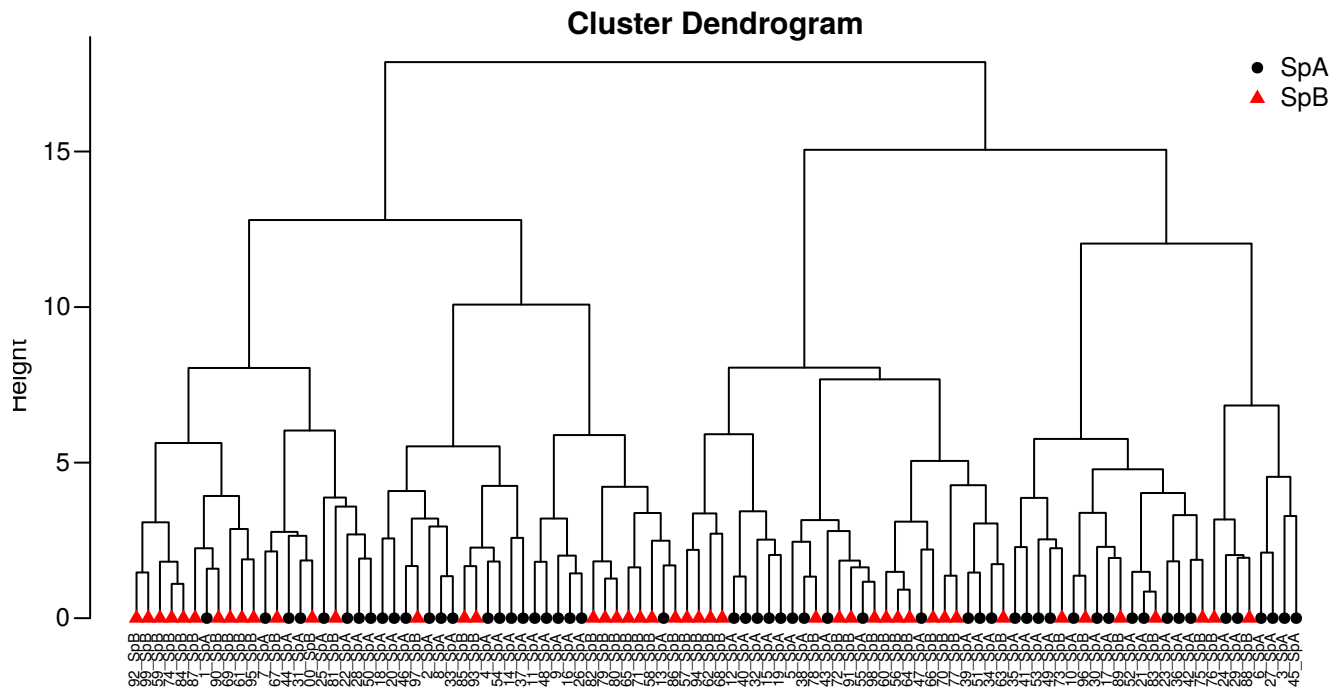


Figure 3:

1.2.2 Ordination : Principal Component Analysis

```
pca <- vegan::rda(d[, -1], scale = TRUE)  
  
# dev.new(width = 12/2.54, height = 8/2.54)  
par(mfrow = c(1,1), mar = c(4,3.5,1,1), mgp = c(1.8, 0.6, 0), cex = 0.8, las = 1)  
eigenplot(pca)
```

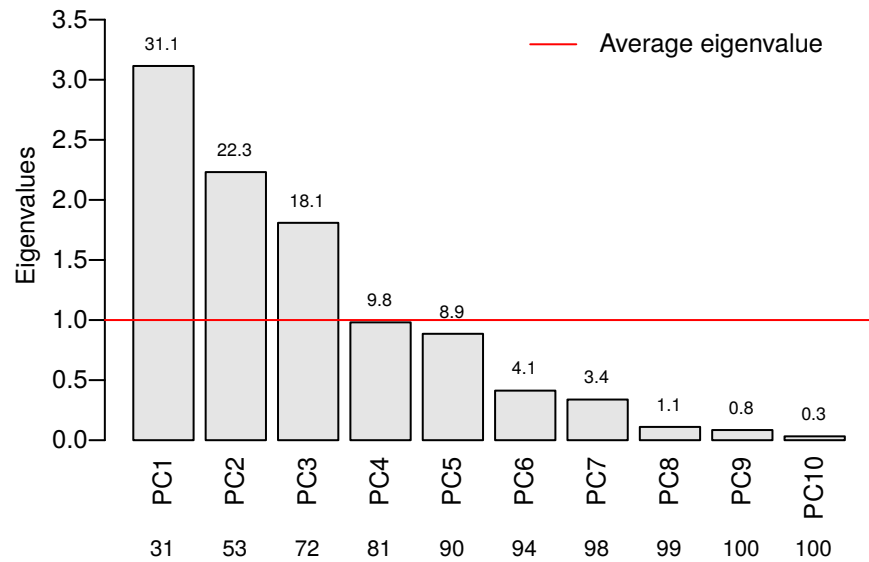


Figure 4:

```
# dev.new(width = 18/2.54, height = 9/2.54)
par(mfrow = c(1,2), mar = c(3.5,3.5,1,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 1)
biplot2(pca, sc = 0.25,
        obs.pch = c(16,17)[as.numeric(d$Species)],
        obs.col = as.numeric(d$Species))
biplot2(pca, sc = 0.15, choices = c(3,4),
        obs.pch = c(16,17)[as.numeric(d$Species)],
        obs.col = as.numeric(d$Species))
legend("topright", pch = c(16, 17), col = 1:2, legend = levels(d$Species), bty = "n")
```

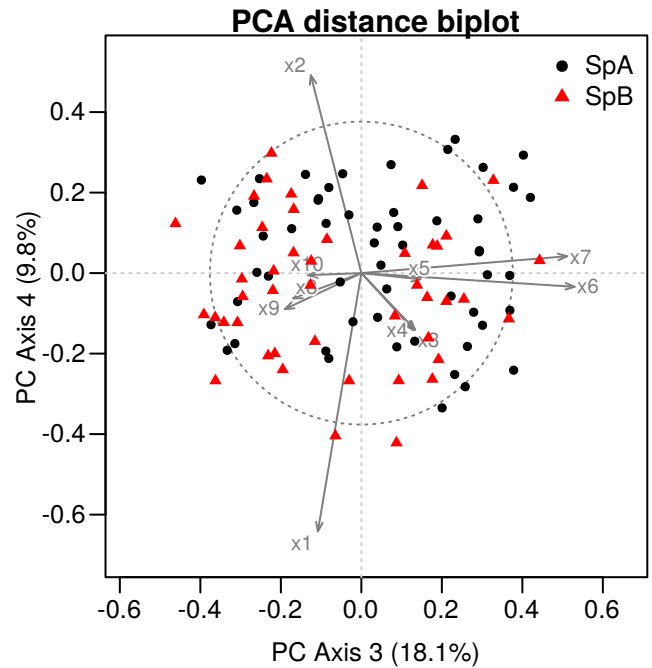
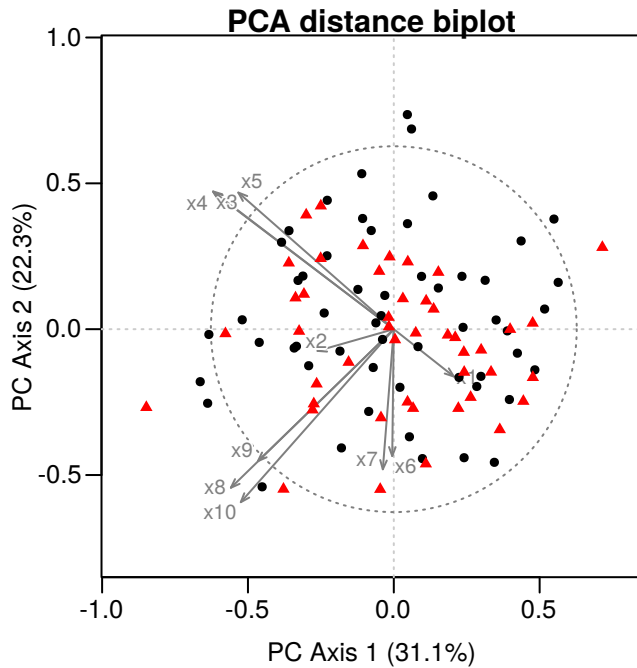


Figure 5:

```
# dev.new(width = 20/2.54, height = 14/2.54)
pairs(pca$CA$u, pch = c("A", "B")[as.numeric(d$Species)],
      col = as.numeric(d$Species), gap = 0, cex = 0.75)
```

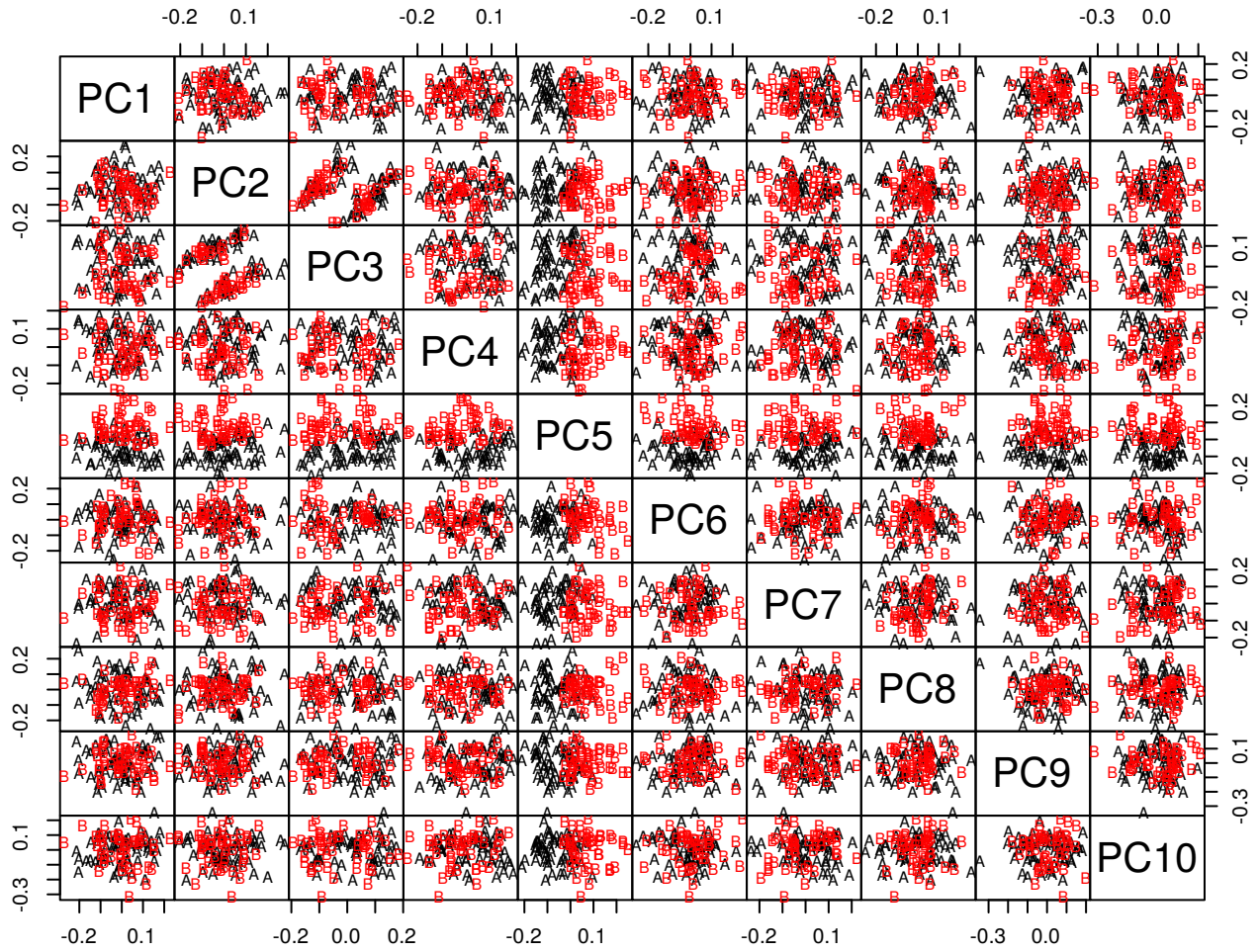


Figure 6:

```
# dev.new(width = 10/2.54, height = 7/2.54)
cos2heatmap(cos2vars(pca))
```

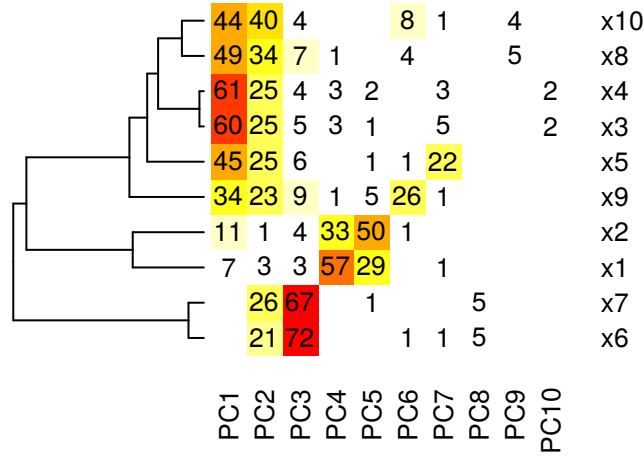


Figure 7:

```
# dev.new(width = 9/2.54, height = 9/2.54)
par(mar = c(3.5,3.5,1,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 1)
biplot2(pca, sc = 0.15, choices = c(4,5),
        obs.pch = c(16,17)[as.numeric(d$Species)],
        obs.col = as.numeric(d$Species))
legend("topleft", pch = c(16, 17), col = 1:2, legend = levels(d$Species), bty = "n")
```

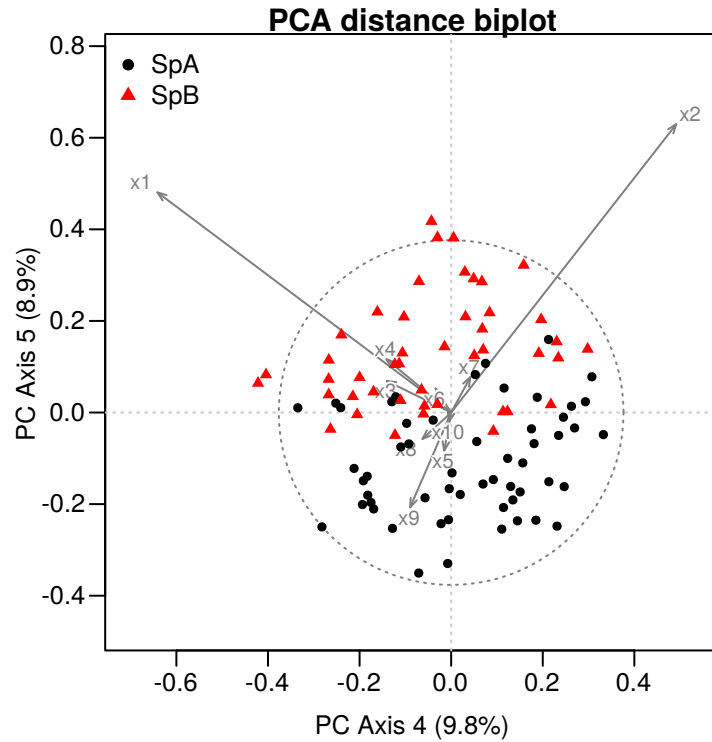


Figure 8:

1.3 Multivariate Supervised Methods

Testing if the 10 variables considered together differ between the 2 species. There are several more or less adapted ways to perform a null hypothesis test on this question. All the tests show that the difference

between the two species is highly significant. However this results in itself is not very useful because a p-value alone is always vary tricky to interpret.

MANOVA ‘Multivariate Analysis of Variance’: requires multinormality and homogeneity of the variances → probably not OK here (distributions are uniform, not normal)

```
m <- manova(as.matrix(d[, -1]) ~ Species, data = d)
summary(m, test="Wilks")
```

```
##           Df  Wilks approx F num Df den Df    Pr(>F)
## Species    1 0.33467   17.694     10    89 < 2.2e-16 ***
## Residuals 98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOSIM : popular method based on any distance matrix (here we use “euclidean”) and permutation tests (no need for multinormality,...). However this method is based on ranks so the data is degraded. Adonis is preferable...

```
ANOSIM <- vegan::anosim(scale(d[, -1]), d$Species, distance = "euclidean",
                        perm = 9999, parallel = 4)
```

ANOSIM

```
##
## Call:
## vegan::anosim(dat = scale(d[, -1]), grouping = d$Species, permutations = 9999,      distance =
## Dissimilarity: euclidean
##
## ANOSIM statistic R: 0.1235
##      Significance: 1e-04
##
## Permutation: free
## Number of permutations: 9999
```

ADONIS : permutation test based on any distance matrix. Here we use a Euclidan distance so this is equivalent to a MANOVA tested by permutation (hence removing many constraints of MANOVA like multinormality)

```
ADONIS <- vegan::adonis2(scale(d[, -1]) ~ Species, data = d, method = "euclidean",
                        perm = 9999, parallel = 4)
```

ADONIS

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## vegan::adonis2(formula = scale(d[, -1]) ~ Species, data = d, permutations = 9999, method = "euc
##           Df SumOfSqs      F Pr(>F)
## Species    1    64.21 6.7969 2e-04 ***
## Residual 98    925.79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Identical to Adonis in that case : RDA followed by a permutation test of the explanatory variables. The

added advantage of RDA is that the triplot helps you to understand where are the major differences between the groups

```
RDA <- vegan::rda( scale(d[, -1]) ~ Species, data = d)
anova(RDA)

## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = scale(d[, -1]) ~ Species, data = d)
##           Df Variance      F Pr(>F)
## Model      1  0.6486 6.7969 0.001 ***
## Residual  98  9.3514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# dev.new(width = 8/2.54, height = 8/2.54)
par(mfrow = c(1,1), mar = c(3.5,3.5,1,1), mgp = c(2, 0.6, 0), cex = 0.8, las = 1)
plot(RDA, type = "n")
points(RDA, cex = 0.8,
       col = as.numeric(d$Species),
       pch = c("A", "B")[as.numeric(d$Species)])
ordilabel(RDA, dis="cn")

sc <- scores(RDA, scaling = 2, display = c("sp"), choices = c(1,2))
arrows(0, 0, sc[,1], sc[,2], length = 0, lty = 1, col = "gray50")
ordilabel(RDA, "species", col="gray50", cex=0.8, border = "gray50")
```

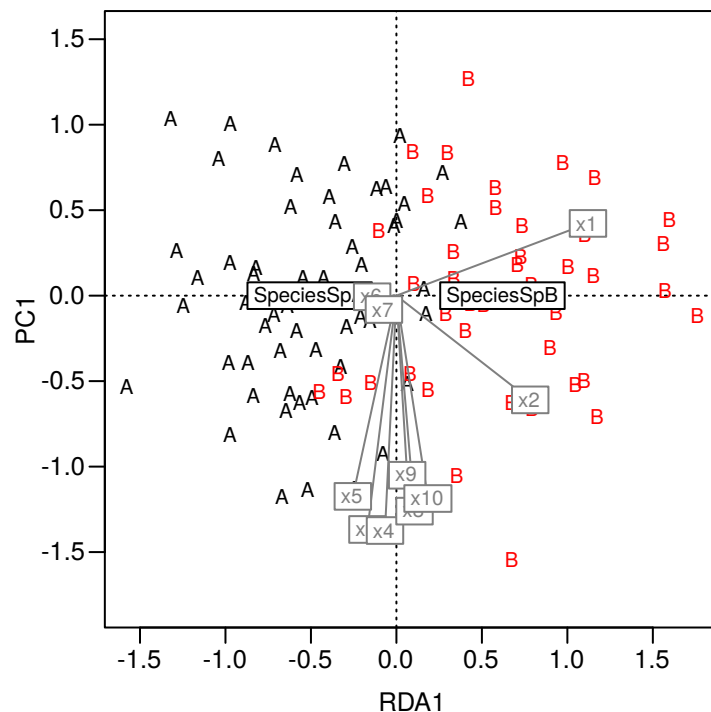


Figure 9:

1.4 Univariate Supervised methods

We can reverse the problem. Instead of trying to explain the 10 morphometric measurements by the species variable we can ask which combination of morphometric measurements predicts best our species. This is in fact our real question here...

We could for example apply a logistic regression. The two explanatory variables x_1 and x_2 are statistically significant and in addition the plots of the predictions of the model show that when x_1 and x_2 have value >0.5 the probability to have Species B increases abruptly.

Logistic regressions are quite robust (for example the explanatory variables must not be multigaussian) but a rather frustrating drawback of this method is that when the separation between the 2 groups (2 species here) is perfect or almost perfect the model might become unstable. This is probably the reason of the warning message : some of the predicted values (probability to be Species b) are so low that they have been estimated to be 0 (while in logistic regression the predicted probabilities might tend towards 1 or 0 but should never reach these values.)

```
tmp <- d[,-1]
tmp$SpB <- as.numeric(d$Species)-1

m <- glm(SpB ~ ., data = tmp, family = binomial)
summary(m)

##
## Call:
## glm(formula = SpB ~ ., family = binomial, data = tmp)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.30556  -0.01059  -0.00005   0.00569   1.71855
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -36.29591    13.78616  -2.633  0.00847 **
## x1           35.77562    13.63877   2.623  0.00871 **
## x2           35.68960    14.12930   2.526  0.01154 *
## x3           -0.92641     0.58298  -1.589  0.11204
## x4            1.73905     1.14277   1.522  0.12806
## x5            0.24394     0.75866   0.322  0.74780
## x6           -0.09287     0.32209  -0.288  0.77310
## x7            0.42084     0.95079   0.443  0.65804
## x8           -0.86638     1.80541  -0.480  0.63131
## x9           -0.42679     1.17208  -0.364  0.71576
## x10           1.38392     1.76684   0.783  0.43346
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 137.628  on 99  degrees of freedom
## Residual deviance:  20.532  on 89  degrees of freedom
## AIC: 42.532
```

```
##
## Number of Fisher Scoring iterations: 10
# dev.new(width = 18/2.54, height = 10/2.54)
par(mfrow = c(2,5), mar = c(2.5,2.5,1,0.4), mgp = c(1.5, 0.5, 0), cex = 0.8, las = 1)
visreg::visreg(m, scale = 'response', partial = TRUE, line.par = list(lwd = 0.5))
```

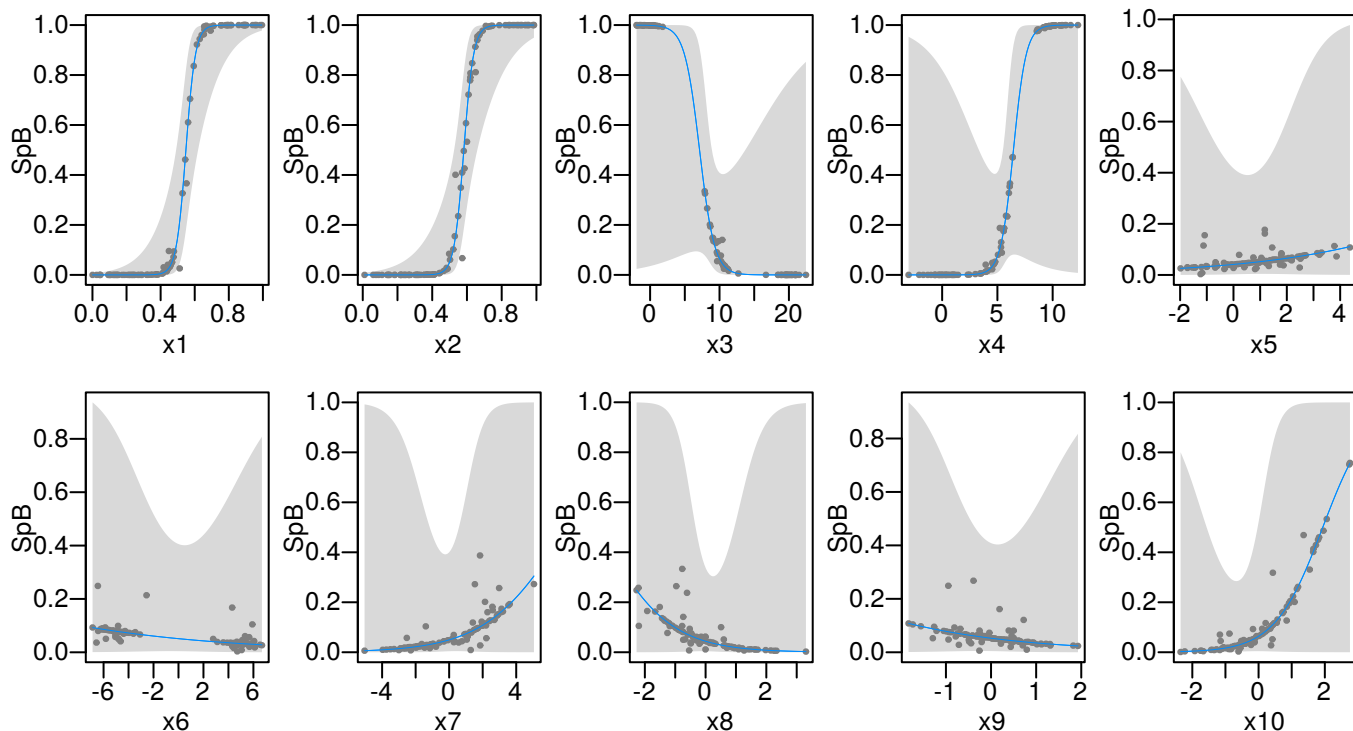


Figure 10:

```
# dev.new(width = 20/2.54, height = 5/2.54)
par(mfrow = c(1,4), mar = c(2.5,2.5,1,1), mgp = c(1.5, 0.5, 0), cex = 0.8, las = 1)
plotmo::plotmo(m, degree1 = 1:2, all2=2, degree2 = 1, do.par = FALSE,
  type2="image", ngrid2=50, level=.95, ylim = c(-0.2,1.2),
  image.col= colorRampPalette(c("lightpink", "lightblue"))(100),
  pt.col=as.numeric(d$Species), pt.pch= c("A","B")[as.numeric(d$Species)],
  pt.cex = 0.8)
plotmo::plotmo(m, degree1 = NULL, all2=2, degree2 = 1, persp.theta=15, do.par = FALSE)
```

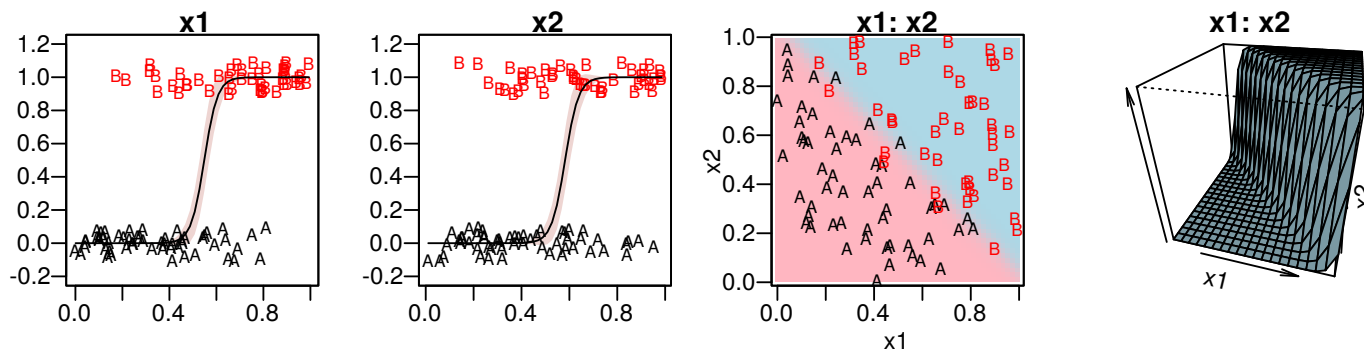


Figure 11:

Another popular method that will be well suited here (because of the simplicity of the simulated dataset) is Linear Discriminant Analysis. this method has the dvantage relative to logistic regression to be unsensitive to perfect separation and that it can work for more than two categories. However there plenty of other assumption that are rarely met to apply this method (similar to MANOVA : multinormality of the explanatory variables, homogenous variance covariance,...)

The `MASS::lda` function has a build-in cross-validation procedure that will allow us to estimate precisely and fairly the quality of the prediction that we might expect with this model.

We drop each observation in turn, build a LDA on the remaining dataset and then predict the value of the observation that has been removed. We can then compute the % of true positives, true negatives and the error rate.

Here wi reach a very low corss-validation error rate of 11% (the LDA makes a wrong prediction about the identity of the species only in 11% of the cases)

```
LDA <- MASS::lda(SpB ~ ., data = tmp, CV = TRUE)
ConfusionMatrix <- table(as.numeric(d$Species), LDA$class)
ConfusionMatrix <- ConfusionMatrix*100/sum(ConfusionMatrix) # in %
ConfusionMatrix
```

```
##
##      0  1
##    1 51  4
##    2  7 38
# leave-one-out cross validated prediction error (in %):
PredError <- sum(ConfusionMatrix) - sum(diag(ConfusionMatrix))
PredError
```

```
## [1] 11
```

here also x1 and x2 have clearly higher coefficients of linear discriminants showing that they are the most important variables to discriminate the species

```
LDA <- MASS::lda(Species ~ ., data = d, CV = FALSE)
LDA
```

```
## Call:
## lda(Species ~ ., data = d, CV = FALSE)
##
## Prior probabilities of groups:
##   SpA  SpB
## 0.55 0.45
##
## Group means:
##           x1           x2           x3           x4           x5           x6           x7           x8
## SpA 0.3365456 0.4094720 10.522876 5.049847 0.9857197 0.3218377 0.03135719 -0.08705745
## SpB 0.6965753 0.6422561  8.915538 4.730958 0.5840684 -0.4985825 -0.15170153 0.03831950
##           x9           x10
## SpA 0.009581508 -0.13435796
## SpB 0.061389033 0.07910217
##
## Coefficients of linear discriminants:
##           LD1
## x1  4.99529347
```

```
## x2 3.83620819
## x3 -0.07583389
## x4 0.13772806
## x5 0.04111980
## x6 -0.04589135
## x7 0.08478234
## x8 -0.50897382
## x9 0.17686026
## x10 0.39950191
```

The graphs of the predictions are similar to the ones of the logistic regression but the transition between Sp A and Sp B is more abrupt here.

```
# dev.new(width = 20/2.54, height = 5/2.54)
par(mfrow = c(1,4), mar = c(2.5,2.5,1,1), mgp = c(1.5, 0.5, 0), cex = 0.8, las = 1)
plotmo::plotmo(LDA, degree1 = 1:2, all2=2, degree2 = 1, do.par = FALSE,
  type2="image", ngrid2=50,
  image.col= colorRampPalette(c("lightpink", "lightblue"))(100),
  pt.col=as.numeric(d$Species), pt.pch= c("A","B")[as.numeric(d$Species)],
  pt.cex = 0.8)
```

```
## plotmo grid:   x1      x2      x3      x4      x5      x6      x7      x8
##               0.4663707 0.4999134 10.28936 4.76589 0.8880767 0.1075062 0.04869008 -0.1682899
##               x9      x10
##               0.09568284 -0.1807299
```

```
plotmo::plotmo(LDA, degree1 = NULL, all2=2, degree2 = 1,
  type2="persp", persp.theta=25, do.par = FALSE)
```

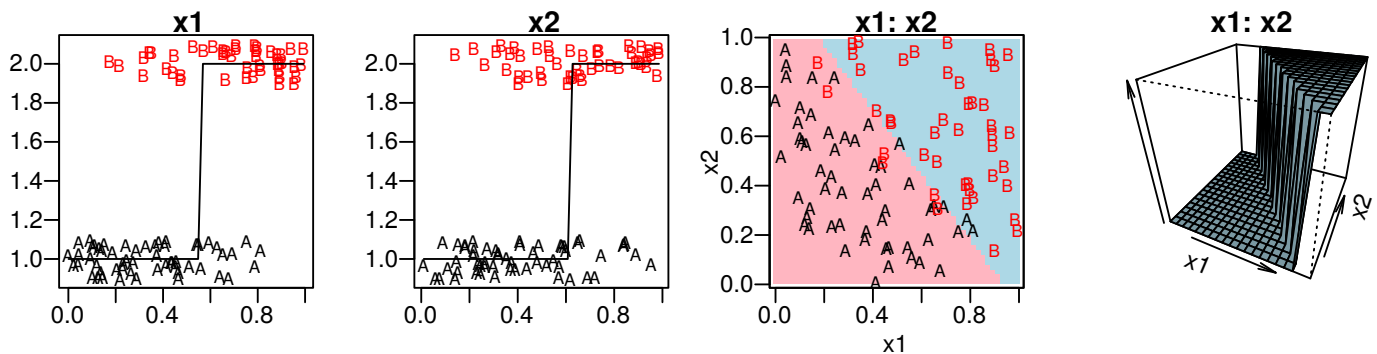


Figure 12: