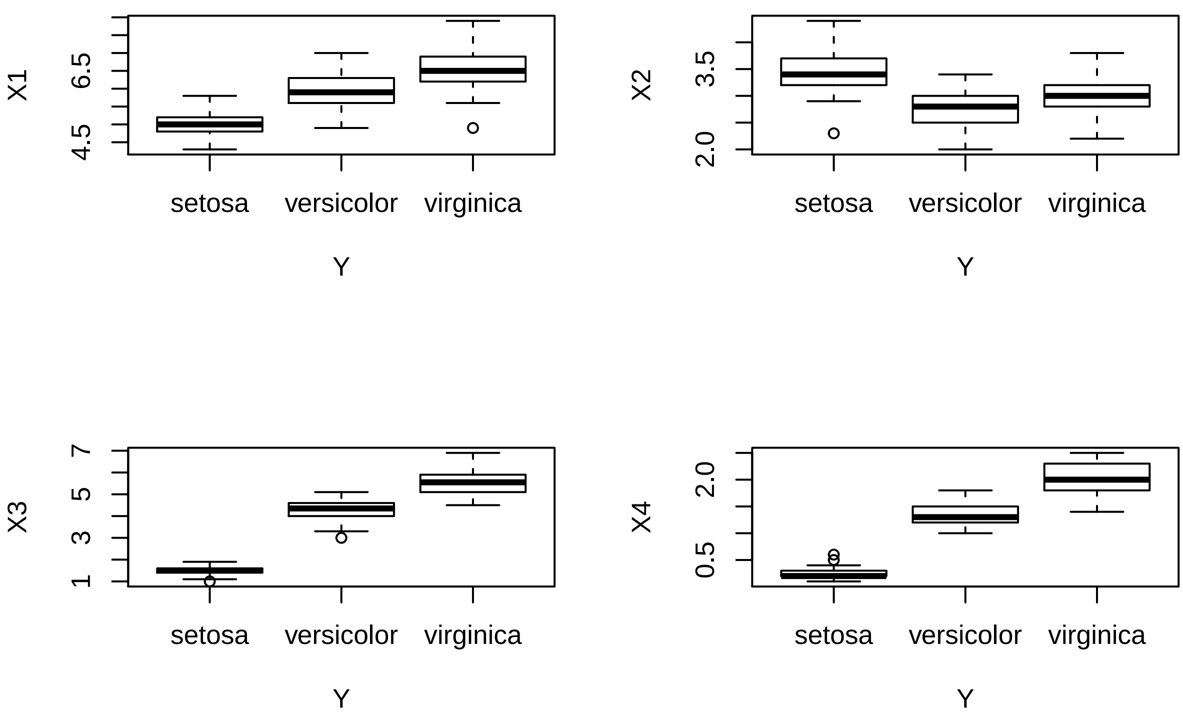
Analyse factorielle discriminante

Guillemette Marot

Graphiques et analyses préliminaires

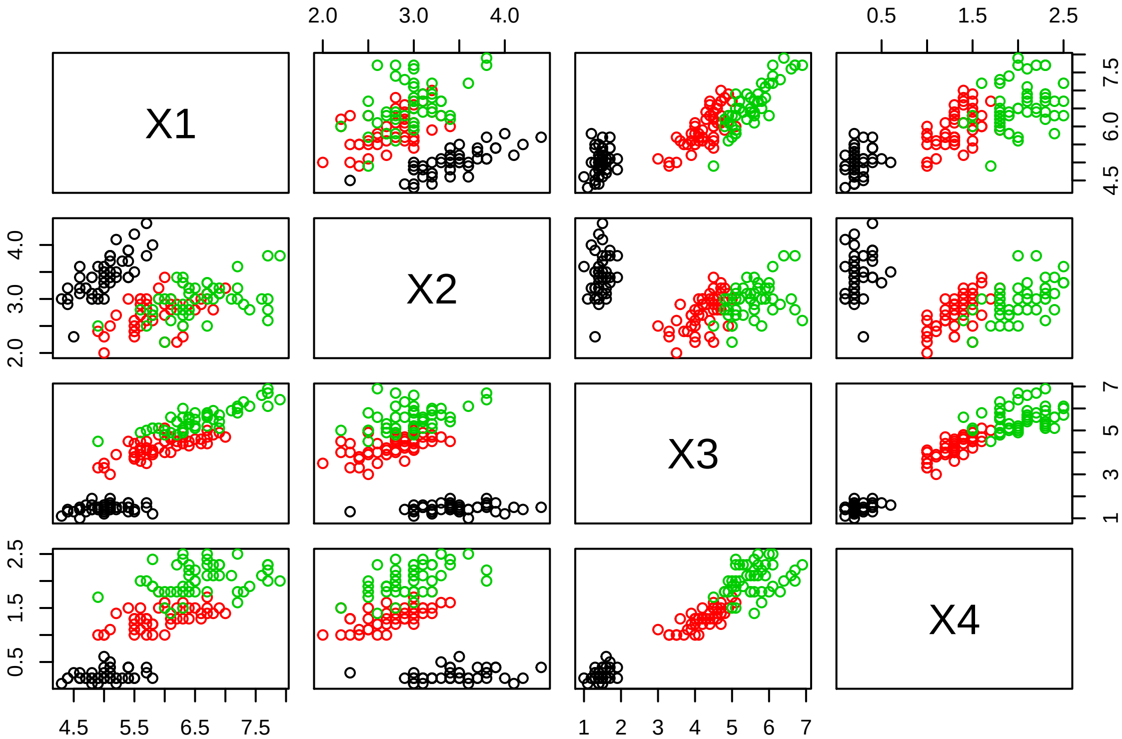
Charger le jeu de données disponible dans R et renommer les variables explicatives avec en X_j et la variable à expliquer Y. Représenter graphiquement le lien entre chaque variable explicative et Y.

```
data("iris")
?iris
colnames(iris)<-c("X1","X2","X3","X4","Y")
par(mfrow=c(2,2))
plot(X1~Y,data=iris)
plot(X2~Y,data=iris)
plot(X3~Y,data=iris)
plot(X4~Y,data=iris)
```



En utilisant la fonction pairs avec l'option col, représenter graphiquement les liens entre les couples de variables quantitatives et le groupe.

```
par(mfrow=c(1,1))
pairs(iris[,1:4],col=iris$Y)
```



Réaliser l'ANOVA de chaque variable explicative en fonction de Y et obtenir le R2 associé, faire de même pour les autres variables. A partir des p-values, indiquer si la variable Y a une influence sur l'ensemble des variables. Quelle est la variable la mieux expliquée par Y ?

```
allanov<-lapply(1:4,FUN=function(i){anova(lm(get(paste0("X",i))~Y,data=iris))})
allpval<-sapply(allanov,FUN=function(x) x$`Pr(>F)`[1])
allpval
```

```
## [1] 1.669669e-31 4.492017e-17 2.856777e-91 4.169446e-85
```

```
allr2<-sapply(summary(lm(cbind(X1,X2,X3,X4)~Y,data=iris)),
FUN=function(x) x$r.squared)
allr2
```

```
## Response X1 Response X2 Response X3 Response X4
##   0.6187057   0.4007828   0.9413717   0.9288829
```

Analyse factorielle discriminante

Calculer V la matrice de variance-covariance globale et W la matrice de variance-covariance intra-groupe.

```
V<-cov.wt(iris[,1:4],method="ML")$cov
V
```

```
##              X1          X2         X3         X4
## X1  0.68112222 -0.04215111  1.2658200  0.5128289
## X2 -0.04215111  0.18871289 -0.3274587 -0.1208284
## X3  1.26582000 -0.32745867  3.0955027  1.2869720
## X4  0.51282889 -0.12082844  1.2869720  0.5771329
```

```
Wi <- lapply(levels(iris$Y), function(k)
  cov.wt(iris[iris$Y== k,1:4],method="ML")$cov) # Liste de Wi
```

```
ni <- table(iris$Y)  # Vecteur de ni
W <- (ni[1]*Wi[[1]] + ni[2]*Wi[[2]] + ni[3]*Wi[[3]])/sum(ni)
#W = Reduce('+',Map('*',Wi,ni))/sum(ni)
```

Calculer B la matrice de variance-covariance inter-groupes, après avoir calculé G, la matrice des moyennes vue en cours. Vérifier que V=B+W.

```
moyennes<-by(iris[,1:4],iris$Y,colMeans)
moyennes
```

```
## iris$Y: setosa
##    X1    X2    X3    X4
## 5.006 3.428 1.462 0.246
## ------------------------------------------------------------
## iris$Y: versicolor
##    X1    X2    X3    X4
## 5.936 2.770 4.260 1.326
## ------------------------------------------------------------
## iris$Y: virginica
##    X1    X2    X3    X4
## 6.588 2.974 5.552 2.026
```

```
G<-matrix(unlist(moyennes),3,4,byrow=T)
rownames(G)=levels(iris$Y)
colnames(G)=paste0(colnames(iris[,1:4]),"bar")
G
```

```
##            X1bar X2bar X3bar X4bar
## setosa     5.006 3.428 1.462 0.246
## versicolor 5.936 2.770 4.260 1.326
## virginica  6.588 2.974 5.552 2.026
```

```
#G <- t(simplify2array(by(iris[,1:4],iris$Y, colMeans)))
B<-cov.wt(G,wt = as.vector(table(iris$Y)),method="ML")$cov
# on precise wt : pour pondérer par l'effectif des classes
V-(B+W)
```

```
##               X1            X2            X3            X4
## X1  1.110223e-16 -9.714451e-17  0.000000e+00  5.551115e-16
## X2 -9.714451e-17 -1.942890e-16 -2.775558e-16 -1.110223e-16
## X3  0.000000e+00 -2.775558e-16  1.332268e-15  1.332268e-15
## X4  5.551115e-16 -1.110223e-16  1.332268e-15  5.551115e-16
```

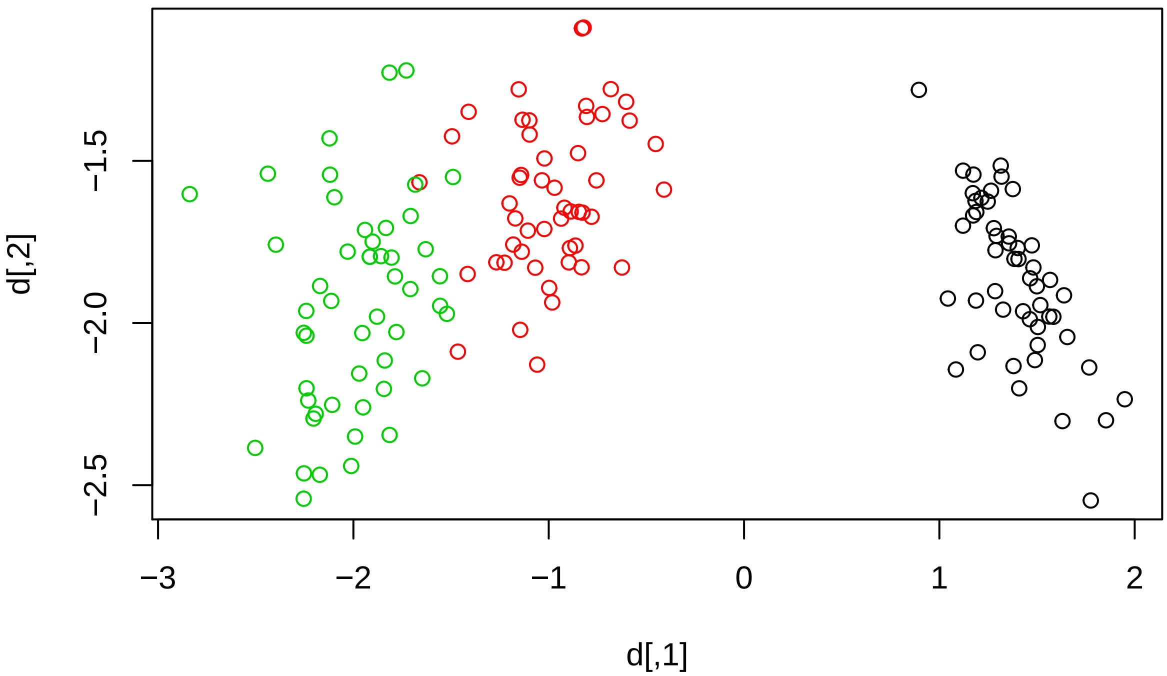
Calculer les coordonnées d1 et d2 des points projetés sur les deux premières composantes discriminantes et faire le graphique correspondant à ce premier plan.

```
M = solve(V) %*% B
eigen(M)
```

```
## eigen() decomposition
## $values
## [1]  9.698722e-01  2.220266e-01  5.441883e-15 -5.292628e-16
##
## $vectors
##            [,1]         [,2]       [,3]       [,4]
## [1,]  0.2087418 -0.006531964  0.2106843 -0.8850525
## [2,]  0.3862037 -0.586610553 -0.3962378  0.2969366
```

```
## [3,] -0.5540117  0.252561540 -0.4591404  0.2754433
## [4,] -0.7073504 -0.769453092  0.7666797  0.2294377
```

```
AFD=eigen(M)$vectors
d=as.matrix(iris[,1:4])%*%AFD[,1:2]
plot(d,col=iris$Y)
```



Vérifier qu'on a seulement 2 valeurs propres non nulles. Quelle est la part de variance de d1 expliquée par la classe ?

```
eigen(M)$values
```

```
## [1]  9.698722e-01  2.220266e-01  5.441883e-15 -5.292628e-16
```

```
summary(lm(d[,1]~iris$Y))$r.squared
```

```
## [1] 0.9698722
```

Analyse discriminante linéaire

Charger le package MASS, puis utiliser la fonction lda qui permet d'ajuster le modèle d'analyse discriminante linéaire. En utilisant les fonctions predict et table, réaliser la matrice de confusion.

```
library("MASS")
X<-iris[,1:4]
Y<-iris[,5]
LDAXY <-  lda(X,grouping=Y,prior = prop.table(rep(1,nlevels(Y))))
LDAXY
```

```
## Call:
## lda(X, grouping = Y, prior = prop.table(rep(1, nlevels(Y))))
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
```

```
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##                X1    X2    X3    X4
## setosa     5.006 3.428 1.462 0.246
## versicolor 5.936 2.770 4.260 1.326
## virginica  6.588 2.974 5.552 2.026
##
## Coefficients of linear discriminants:
##           LD1         LD2
## X1  0.8293776  0.02410215
## X2  1.5344731  2.16452123
## X3 -2.2012117 -0.93192121
## X4 -2.8104603  2.83918785
##
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

```
Yp <- predict(LDAXY)
table(Y,Ypredit = Yp$class)
```

```
##             Ypredit
## Y             setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

Evaluer le taux de mauvais classement par validation croisée leave-one-out, en utilisant l'option CV = TRUE dans la fonction lda. Réaliser aussi la matrice de confusion. On remarque que les classes d'affectation et les probabilités a posteriori sont directement renvoyées dans l'objet de sortie, sans faire appel à predict

```
LDAXYLOO = lda(X,grouping=Y, CV=TRUE)
table(Yreel=Y,LOO=LDAXYLOO$class)
```

```
##             LOO
## Yreel         setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

Utiliser à nouveau la fonction lda sans préciser l'option CV=TRUE. Dans les sorties on remarque que les coefficients linéaires discriminants peuvent être récupérés par le champ scaling de l'objet retourné par la fonction lda (cette sortie n'est pas disponible dans le cas où on a CV=TRUE). En multipliant la matrice de données par la matrice des coefficients linéaires discriminants, obtenir une projection des individus sur ces axes discriminants. Faire le graphique permettant de visualiser ces données.

```
LDAXY <-  lda(X,grouping=Y,prior = prop.table(rep(1,nlevels(Y))))
LDAv=LDAXY$scaling
LDAv
```

```
##           LD1         LD2
## X1  0.8293776  0.02410215
## X2  1.5344731  2.16452123
## X3 -2.2012117 -0.93192121
## X4 -2.8104603  2.83918785
```

```r
#projection
Projec=as.matrix(X) %*% LDAv
plot(Projec, col=iris$Y)
```