Multivariate Hw02

Lin Chian Hung

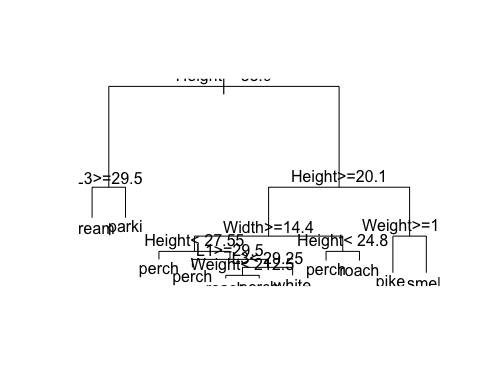
2018/4/21

# Classification tree

fish<-read.csv("~/Desktop/碩一下/多變量/Fishdata.csv")  
library(rpart)  
fish.control<-rpart.control(minisplit=10,minbucket=3,xval=0)  
fish.treeorig<-rpart(Species~  
Weight+L1+L2+L3+Height+Width,data=fish,method="class",control=fish.control)

Let’s now plot the tree:

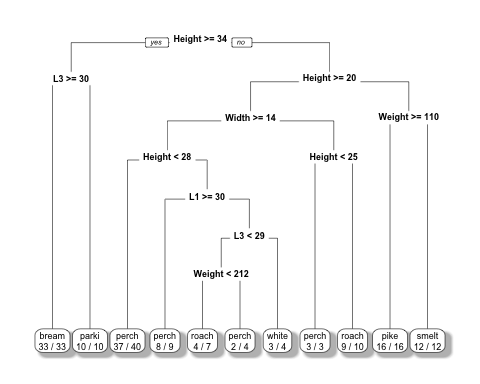
# 第一種畫法  
plot(fish.treeorig)  
text(fish.treeorig)



# 第二種畫法  
require(rpart.plot)

## Loading required package: rpart.plot

prp(fish.treeorig, # 模型  
 faclen=0, # 呈現的變數不要縮寫  
 fallen.leaves=TRUE, # 讓樹枝以垂直方式呈現  
 shadow.col="gray", # 最下面的節點塗上陰影  
 # number of correct classifications / number of observations in that node  
 extra=2)



Also check out the complexity parameter (CP):

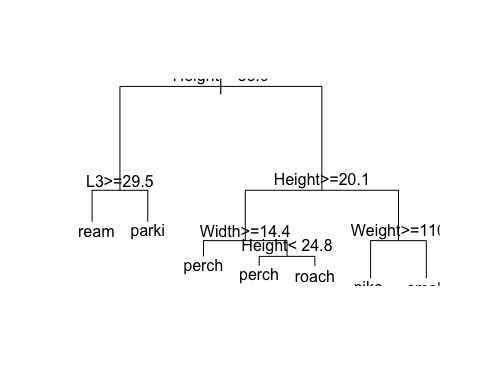
printcp(fish.treeorig)

##   
## Classification tree:  
## rpart(formula = Species ~ Weight + L1 + L2 + L3 + Height + Width,   
## data = fish, method = "class", control = fish.control)  
##   
## Variables actually used in tree construction:  
## [1] Height L1 L3 Weight Width   
##   
## Root node error: 94/148 = 0.63514  
##   
## n= 148   
##   
## CP nsplit rel error  
## 1 0.351064 0 1.00000  
## 2 0.170213 1 0.64894  
## 3 0.127660 2 0.47872  
## 4 0.106383 3 0.35106  
## 5 0.053191 4 0.24468  
## 6 0.031915 5 0.19149  
## 7 0.010638 6 0.15957  
## 8 0.010000 10 0.11702

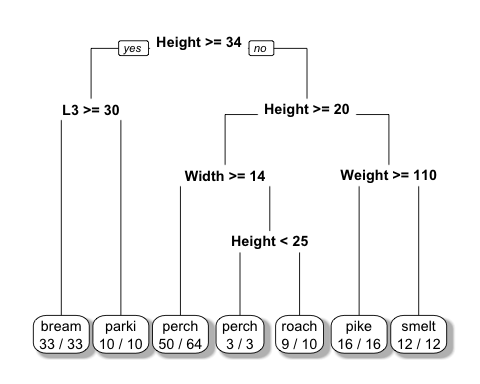
summary(fish.treeorig)

## Call:  
## rpart(formula = Species ~ Weight + L1 + L2 + L3 + Height + Width,   
## data = fish, method = "class", control = fish.control)  
## n= 148   
##   
## CP nsplit rel error  
## 1 0.35106383 0 1.0000000  
## 2 0.17021277 1 0.6489362  
## 3 0.12765957 2 0.4787234  
## 4 0.10638298 3 0.3510638  
## 5 0.05319149 4 0.2446809  
## 6 0.03191489 5 0.1914894  
## 7 0.01063830 6 0.1595745  
## 8 0.01000000 10 0.1170213  
##   
## Variable importance  
## Height L3 L2 L1 Weight Width   
## 26 16 15 15 15 13   
##   
## Node number 1: 148 observations, complexity param=0.3510638  
## predicted class=perch expected loss=0.6351351 P(node) =1  
## class counts: 33 10 54 16 18 12 5  
## probabilities: 0.223 0.068 0.365 0.108 0.122 0.081 0.034   
## left son=2 (43 obs) right son=3 (105 obs)  
## Primary splits:  
## Height < 33.9 to the right, improve=29.75863, (0 missing)  
## Width < 11.85 to the right, improve=17.98385, (0 missing)  
## L3 < 29.7 to the right, improve=13.80398, (0 missing)  
## L2 < 28.85 to the right, improve=12.96300, (0 missing)  
## L1 < 26.1 to the right, improve=12.56245, (0 missing)  
##   
## Node number 2: 43 observations, complexity param=0.106383  
## predicted class=bream expected loss=0.2325581 P(node) =0.2905405  
## class counts: 33 10 0 0 0 0 0  
## probabilities: 0.767 0.233 0.000 0.000 0.000 0.000 0.000   
## left son=4 (33 obs) right son=5 (10 obs)  
## Primary splits:  
## L3 < 29.5 to the right, improve=15.348840, (0 missing)  
## L2 < 26.15 to the right, improve=13.530660, (0 missing)  
## L1 < 23.1 to the right, improve=13.407660, (0 missing)  
## Weight < 331.5 to the right, improve=12.015500, (0 missing)  
## Width < 14.85 to the right, improve= 1.063123, (0 missing)  
## Surrogate splits:  
## L1 < 23.1 to the right, agree=0.977, adj=0.9, (0 split)  
## L2 < 25.2 to the right, agree=0.977, adj=0.9, (0 split)  
## Weight < 221 to the right, agree=0.953, adj=0.8, (0 split)  
##   
## Node number 3: 105 observations, complexity param=0.1702128  
## predicted class=perch expected loss=0.4857143 P(node) =0.7094595  
## class counts: 0 0 54 16 18 12 5  
## probabilities: 0.000 0.000 0.514 0.152 0.171 0.114 0.048   
## left son=6 (77 obs) right son=7 (28 obs)  
## Primary splits:  
## Height < 20.1 to the right, improve=21.78355, (0 missing)  
## Width < 12.45 to the right, improve=20.93000, (0 missing)  
## Weight < 25.95 to the right, improve=13.35778, (0 missing)  
## L3 < 15.6 to the right, improve=10.68888, (0 missing)  
## L1 < 12.3 to the right, improve=10.63876, (0 missing)  
## Surrogate splits:  
## Width < 12.45 to the right, agree=0.990, adj=0.964, (0 split)  
## Weight < 25.95 to the right, agree=0.838, adj=0.393, (0 split)  
## L1 < 12.3 to the right, agree=0.819, adj=0.321, (0 split)  
## L2 < 13.35 to the right, agree=0.819, adj=0.321, (0 split)  
## L3 < 14.25 to the right, agree=0.819, adj=0.321, (0 split)  
##   
## Node number 4: 33 observations  
## predicted class=bream expected loss=0 P(node) =0.222973  
## class counts: 33 0 0 0 0 0 0  
## probabilities: 1.000 0.000 0.000 0.000 0.000 0.000 0.000   
##   
## Node number 5: 10 observations  
## predicted class=parki expected loss=0 P(node) =0.06756757  
## class counts: 0 10 0 0 0 0 0  
## probabilities: 0.000 1.000 0.000 0.000 0.000 0.000 0.000   
##   
## Node number 6: 77 observations, complexity param=0.05319149  
## predicted class=perch expected loss=0.2987013 P(node) =0.5202703  
## class counts: 0 0 54 0 18 0 5  
## probabilities: 0.000 0.000 0.701 0.000 0.234 0.000 0.065   
## left son=12 (64 obs) right son=13 (13 obs)  
## Primary splits:  
## Width < 14.4 to the right, improve=5.777691, (0 missing)  
## Height < 25.25 to the left, improve=4.275974, (0 missing)  
## L1 < 25.1 to the right, improve=2.872913, (0 missing)  
## L2 < 27.15 to the right, improve=2.872913, (0 missing)  
## Weight < 548 to the right, improve=2.448383, (0 missing)  
## Surrogate splits:  
## L1 < 13.35 to the right, agree=0.844, adj=0.077, (0 split)  
## L2 < 14.55 to the right, agree=0.844, adj=0.077, (0 split)  
##   
## Node number 7: 28 observations, complexity param=0.1276596  
## predicted class=pike expected loss=0.4285714 P(node) =0.1891892  
## class counts: 0 0 0 16 0 12 0  
## probabilities: 0.000 0.000 0.000 0.571 0.000 0.429 0.000   
## left son=14 (16 obs) right son=15 (12 obs)  
## Primary splits:  
## Weight < 109.95 to the right, improve=13.714290, (0 missing)  
## L1 < 21.9 to the right, improve=13.714290, (0 missing)  
## L2 < 23.65 to the right, improve=13.714290, (0 missing)  
## L3 < 25.5 to the right, improve=13.714290, (0 missing)  
## Height < 16.05 to the left, improve= 4.571429, (0 missing)  
## Surrogate splits:  
## L1 < 21.9 to the right, agree=1.000, adj=1.000, (0 split)  
## L2 < 23.65 to the right, agree=1.000, adj=1.000, (0 split)  
## L3 < 25.5 to the right, agree=1.000, adj=1.000, (0 split)  
## Height < 16.05 to the left, agree=0.786, adj=0.500, (0 split)  
## Width < 9.45 to the right, agree=0.714, adj=0.333, (0 split)  
##   
## Node number 12: 64 observations, complexity param=0.0106383  
## predicted class=perch expected loss=0.21875 P(node) =0.4324324  
## class counts: 0 0 50 0 9 0 5  
## probabilities: 0.000 0.000 0.781 0.000 0.141 0.000 0.078   
## left son=24 (40 obs) right son=25 (24 obs)  
## Primary splits:  
## Height < 27.55 to the left, improve=3.314583, (0 missing)  
## Width < 15.65 to the right, improve=1.557526, (0 missing)  
## L1 < 30 to the right, improve=1.174970, (0 missing)  
## L2 < 32.25 to the right, improve=1.174970, (0 missing)  
## Weight < 548 to the right, improve=1.058472, (0 missing)  
## Surrogate splits:  
## Width < 17.4 to the left, agree=0.719, adj=0.250, (0 split)  
## Weight < 267.5 to the left, agree=0.688, adj=0.167, (0 split)  
## L3 < 29.05 to the left, agree=0.656, adj=0.083, (0 split)  
##   
## Node number 13: 13 observations, complexity param=0.03191489  
## predicted class=roach expected loss=0.3076923 P(node) =0.08783784  
## class counts: 0 0 4 0 9 0 0  
## probabilities: 0.000 0.000 0.308 0.000 0.692 0.000 0.000   
## left son=26 (3 obs) right son=27 (10 obs)  
## Primary splits:  
## Height < 24.8 to the left, improve=3.7384620, (0 missing)  
## Weight < 174.5 to the right, improve=1.0051280, (0 missing)  
## L1 < 22.5 to the right, improve=1.0051280, (0 missing)  
## L2 < 24.5 to the right, improve=1.0051280, (0 missing)  
## L3 < 21.1 to the left, improve=0.4273504, (0 missing)  
##   
## Node number 14: 16 observations  
## predicted class=pike expected loss=0 P(node) =0.1081081  
## class counts: 0 0 0 16 0 0 0  
## probabilities: 0.000 0.000 0.000 1.000 0.000 0.000 0.000   
##   
## Node number 15: 12 observations  
## predicted class=smelt expected loss=0 P(node) =0.08108108  
## class counts: 0 0 0 0 0 12 0  
## probabilities: 0.000 0.000 0.000 0.000 0.000 1.000 0.000   
##   
## Node number 24: 40 observations  
## predicted class=perch expected loss=0.075 P(node) =0.2702703  
## class counts: 0 0 37 0 3 0 0  
## probabilities: 0.000 0.000 0.925 0.000 0.075 0.000 0.000   
##   
## Node number 25: 24 observations, complexity param=0.0106383  
## predicted class=perch expected loss=0.4583333 P(node) =0.1621622  
## class counts: 0 0 13 0 6 0 5  
## probabilities: 0.000 0.000 0.542 0.000 0.250 0.000 0.208   
## left son=50 (9 obs) right son=51 (15 obs)  
## Primary splits:  
## L1 < 29.5 to the right, improve=2.772222, (0 missing)  
## L2 < 31.9 to the right, improve=2.772222, (0 missing)  
## Width < 16.45 to the right, improve=2.772222, (0 missing)  
## Weight < 295 to the right, improve=2.583333, (0 missing)  
## L3 < 32.4 to the right, improve=2.216667, (0 missing)  
## Surrogate splits:  
## L2 < 31.9 to the right, agree=1.000, adj=1.000, (0 split)  
## Weight < 410 to the right, agree=0.958, adj=0.889, (0 split)  
## L3 < 32.4 to the right, agree=0.958, adj=0.889, (0 split)  
## Width < 16.45 to the right, agree=0.833, adj=0.556, (0 split)  
## Height < 29.35 to the right, agree=0.667, adj=0.111, (0 split)  
##   
## Node number 26: 3 observations  
## predicted class=perch expected loss=0 P(node) =0.02027027  
## class counts: 0 0 3 0 0 0 0  
## probabilities: 0.000 0.000 1.000 0.000 0.000 0.000 0.000   
##   
## Node number 27: 10 observations  
## predicted class=roach expected loss=0.1 P(node) =0.06756757  
## class counts: 0 0 1 0 9 0 0  
## probabilities: 0.000 0.000 0.100 0.000 0.900 0.000 0.000   
##   
## Node number 50: 9 observations  
## predicted class=perch expected loss=0.1111111 P(node) =0.06081081  
## class counts: 0 0 8 0 0 0 1  
## probabilities: 0.000 0.000 0.889 0.000 0.000 0.000 0.111   
##   
## Node number 51: 15 observations, complexity param=0.0106383  
## predicted class=roach expected loss=0.6 P(node) =0.1013514  
## class counts: 0 0 5 0 6 0 4  
## probabilities: 0.000 0.000 0.333 0.000 0.400 0.000 0.267   
## left son=102 (11 obs) right son=103 (4 obs)  
## Primary splits:  
## L3 < 29.25 to the left, improve=2.003030, (0 missing)  
## Weight < 247.5 to the left, improve=1.866667, (0 missing)  
## L1 < 22.85 to the left, improve=1.866667, (0 missing)  
## L2 < 25 to the left, improve=1.866667, (0 missing)  
## Height < 28.45 to the left, improve=1.088889, (0 missing)  
## Surrogate splits:  
## L1 < 24.05 to the left, agree=0.933, adj=0.75, (0 split)  
## L2 < 26.25 to the left, agree=0.933, adj=0.75, (0 split)  
## Weight < 303 to the left, agree=0.867, adj=0.50, (0 split)  
##   
## Node number 102: 11 observations, complexity param=0.0106383  
## predicted class=perch expected loss=0.5454545 P(node) =0.07432432  
## class counts: 0 0 5 0 5 0 1  
## probabilities: 0.000 0.000 0.455 0.000 0.455 0.000 0.091   
## left son=204 (7 obs) right son=205 (4 obs)  
## Primary splits:  
## Weight < 212.5 to the left, improve=0.4350649, (0 missing)  
## L1 < 22.05 to the left, improve=0.4350649, (0 missing)  
## L2 < 23.75 to the left, improve=0.4350649, (0 missing)  
## L3 < 26.15 to the left, improve=0.4350649, (0 missing)  
## Height < 28.5 to the left, improve=0.4350649, (0 missing)  
## Surrogate splits:  
## L2 < 23.75 to the left, agree=1.000, adj=1.00, (0 split)  
## Height < 28.5 to the left, agree=1.000, adj=1.00, (0 split)  
## L1 < 21.25 to the left, agree=0.909, adj=0.75, (0 split)  
## L3 < 25.4 to the left, agree=0.909, adj=0.75, (0 split)  
## Width < 14.95 to the right, agree=0.727, adj=0.25, (0 split)  
##   
## Node number 103: 4 observations  
## predicted class=white expected loss=0.25 P(node) =0.02702703  
## class counts: 0 0 0 0 1 0 3  
## probabilities: 0.000 0.000 0.000 0.000 0.250 0.000 0.750   
##   
## Node number 204: 7 observations  
## predicted class=roach expected loss=0.4285714 P(node) =0.0472973  
## class counts: 0 0 3 0 4 0 0  
## probabilities: 0.000 0.000 0.429 0.000 0.571 0.000 0.000   
##   
## Node number 205: 4 observations  
## predicted class=perch expected loss=0.5 P(node) =0.02702703  
## class counts: 0 0 2 0 1 0 1  
## probabilities: 0.000 0.000 0.500 0.000 0.250 0.000 0.250

fish.prunetree<-prune.rpart(fish.treeorig,cp=0.02)   
# 第一種畫法  
plot(fish.prunetree)  
text(fish.prunetree)



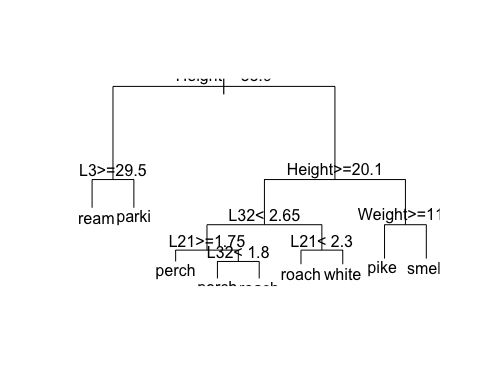
# 第二種畫法  
require(rpart.plot)   
prp(fish.prunetree, # 模型  
 faclen=0, # 呈現的變數不要縮寫  
 fallen.leaves=TRUE, # 讓樹枝以垂直方式呈現  
 shadow.col="gray", # 最下面的節點塗上陰影  
 # number of correct classifications / number of observations in that node  
 extra=2)



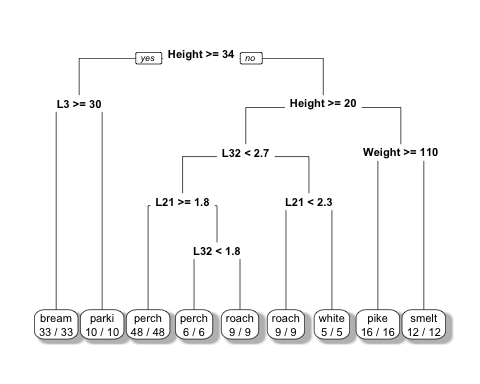
L21<-fish$L2-fish$L1  
L32<-fish$L3-fish$L2  
L31<-fish$L3-fish$L1  
newfish<-cbind(fish,L21,L32,L31)  
newfish.treenew<-rpart(Species~., data=newfish,method="class",parms=list(split="information"),control=fish.control)   
printcp(newfish.treenew)

##   
## Classification tree:  
## rpart(formula = Species ~ ., data = newfish, method = "class",   
## parms = list(split = "information"), control = fish.control)  
##   
## Variables actually used in tree construction:  
## [1] Height L21 L3 L32 Weight  
##   
## Root node error: 94/148 = 0.63514  
##   
## n= 148   
##   
## CP nsplit rel error  
## 1 0.351064 0 1.000000  
## 2 0.170213 1 0.648936  
## 3 0.127660 2 0.478723  
## 4 0.106383 3 0.351064  
## 5 0.095745 4 0.244681  
## 6 0.053191 5 0.148936  
## 7 0.047872 6 0.095745  
## 8 0.010000 8 0.000000

# 第一種畫法  
plot(newfish.treenew)   
text(newfish.treenew)



# 第二種畫法  
require(rpart.plot)   
prp(newfish.treenew, # 模型  
 faclen=0, # 呈現的變數不要縮寫  
 fallen.leaves=TRUE, # 讓樹枝以垂直方式呈現  
 shadow.col="gray", # 最下面的節點塗上陰影  
 # number of correct classifications / number of observations in that node  
 extra=2)



fish.control<-rpart.control(minbucket=3,minsplit=10,xval=148)  
newfish.treenewcv<- rpart(Species~., data=newfish,method="class",  
parms=list(split='information'),control=fish.control)   
printcp(newfish.treenewcv)

##   
## Classification tree:  
## rpart(formula = Species ~ ., data = newfish, method = "class",   
## parms = list(split = "information"), control = fish.control)  
##   
## Variables actually used in tree construction:  
## [1] Height L21 L3 L32 Weight  
##   
## Root node error: 94/148 = 0.63514  
##   
## n= 148   
##   
## CP nsplit rel error xerror xstd  
## 1 0.351064 0 1.000000 1.000000 0.062302  
## 2 0.170213 1 0.648936 0.648936 0.063704  
## 3 0.127660 2 0.478723 0.478723 0.059534  
## 4 0.106383 3 0.351064 0.351064 0.053870  
## 5 0.095745 4 0.244681 0.361702 0.054442  
## 6 0.053191 5 0.148936 0.170213 0.040187  
## 7 0.047872 6 0.095745 0.180851 0.041267  
## 8 0.010000 8 0.000000 0.031915 0.018238

newfish.test<-read.csv("~/Desktop/碩一下/多變量/Fish\_test\_data.csv")  
L31<-newfish.test$L3- newfish.test$L1  
L32<-newfish.test$L3- newfish.test$L2  
L21<-newfish.test$L2- newfish.test$L1  
newfish.test<-cbind(newfish.test,L21,L32,L31)  
newfish.tpred<-predict(newfish.treenewcv,newfish.test)  
newfish.tpred

## bream parki perch pike roach smelt white  
## 1 1 0 0 0 0 0 0  
## 2 1 0 0 0 0 0 0  
## 3 0 0 1 0 0 0 0  
## 4 0 0 1 0 0 0 0  
## 5 0 0 0 1 0 0 0  
## 6 0 0 0 0 0 1 0  
## 7 0 0 0 0 0 1 0  
## 8 0 1 0 0 0 0 0  
## 9 0 0 0 0 1 0 0  
## 10 0 0 0 0 1 0 0  
## 11 0 0 0 0 0 0 1

# Linear Discriminant Analysis

library(MASS)  
newfish.lda<-lda(Species~.,data=newfish)  
# 有共線性存在

newfish.lda<-lda(Species~Weight+L1+Height+Width+L21+L32,data=newfish)   
newfish.lda

## Call:  
## lda(Species ~ Weight + L1 + Height + Width + L21 + L32, data = newfish)  
##   
## Prior probabilities of groups:  
## bream parki perch pike roach smelt   
## 0.22297297 0.06756757 0.36486486 0.10810811 0.12162162 0.08108108   
## white   
## 0.03378378   
##   
## Group means:  
## Weight L1 Height Width L21 L32  
## bream 636.1818 30.60606 39.52727 14.10000 2.8060606 5.272727  
## parki 155.8000 18.62000 39.20000 14.18000 1.6100000 2.430000  
## perch 360.9333 25.31852 26.17778 15.78519 2.1259259 1.650000  
## pike 742.0625 42.88125 15.85625 10.48125 3.0375000 3.281250  
## roach 159.1111 20.66667 26.88333 14.57222 1.6388889 2.716667  
## smelt 11.5000 11.34167 16.99167 10.21667 0.6916667 1.091667  
## white 477.2000 27.82000 29.10000 15.76000 2.4800000 2.960000  
##   
## Coefficients of linear discriminants:  
## LD1 LD2 LD3 LD4 LD5  
## Weight 0.000911022 -0.002710071 0.007553399 0.001688806 0.006182751  
## L1 0.132200166 0.036926540 -0.259794107 -0.235599786 -0.330471903  
## Height -0.618519868 -0.332732865 -0.053863042 -0.330737436 -0.029226039  
## Width 0.464670922 -0.341184928 -0.353062958 0.842951264 -0.201141743  
## L21 -0.114071841 0.712452136 -2.278059990 0.277900320 2.700516892  
## L32 -2.311243186 2.141452146 0.539501848 1.803654269 -0.461925634  
## LD6  
## Weight -0.003600115  
## L1 -0.119589009  
## Height -0.019796935  
## Width -0.159484049  
## L21 2.813216431  
## L32 -0.080912628  
##   
## Proportion of trace:  
## LD1 LD2 LD3 LD4 LD5 LD6   
## 0.7998 0.1327 0.0473 0.0167 0.0035 0.0000

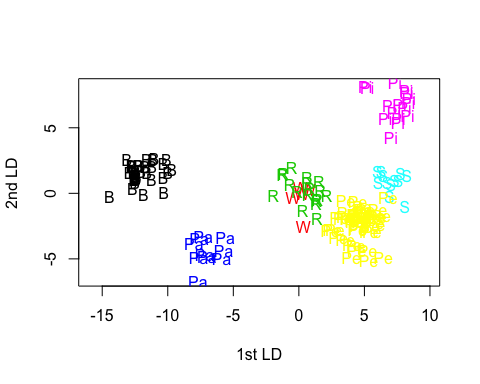
newfish.ldapred<-predict(newfish.lda,newfish[,-1])  
table(newfish$Species,newfish.ldapred$class)

##   
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 1 0 4

newfish.ldacv<-lda(Species~Weight+L1+Height+Width+L21+L32,data=newfish,CV=T)   
table(newfish$Species,newfish.ldacv$class)

##   
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 1 0 4

eqscplot(newfish.ldapred$x,type='n',xlab="1st LD",ylab="2nd LD")  
fish.species<-c(rep("B",33),rep("W",5),rep("R",18),rep("Pa",10),rep("S",12),rep("Pi",16),rep("Pe",54))  
fish.colors<-c(rep(1,33),rep(2,5),rep(3,18),rep(4,10),rep(5,12),rep(6,16),rep(7,54))   
text(newfish.ldapred$x[,1:2],fish.species,col=fish.colors)



newfish.ldatest<-predict(newfish.lda,newfish.test)   
newfish.ldatest$class

## [1] bream bream perch perch pike smelt smelt parki roach roach white  
## Levels: bream parki perch pike roach smelt white

# Quadratic Discriminant Analysis

Let us examine how to apply QDA to this dataset. Let us pretend that the multivariate normal assumption is valid (you can check for this by using the R package “MVN”, as shown in the lab of CCA)

Check Normailty

library(MVN)

## sROC 0.1-2 loaded

fish1 <- newfish[,2:6]  
mvn(fish1,mvnTest = "hz")

## $multivariateNormality  
## Test HZ p value MVN  
## 1 Henze-Zirkler 5.571187 0 NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Shapiro-Wilk Weight 0.8814 <0.001 NO   
## 2 Shapiro-Wilk L1 0.9696 0.0023 NO   
## 3 Shapiro-Wilk L2 0.9720 0.004 NO   
## 4 Shapiro-Wilk L3 0.9728 0.0048 NO   
## 5 Shapiro-Wilk Height 0.9171 <0.001 NO   
##   
## $Descriptives  
## n Mean Std.Dev Median Min Max 25th 75th  
## Weight 148 400.69865 358.364429 272.50 5.9 1650.0 120.000 650.00  
## L1 148 26.32905 10.037659 25.30 7.5 59.0 19.000 32.55  
## L2 148 28.50676 10.749091 27.40 8.4 63.4 21.000 35.00  
## L3 148 31.32230 11.671320 29.35 8.8 68.0 23.025 39.55  
## Height 148 28.35811 8.251992 26.90 14.5 44.5 24.300 37.50  
## Skew Kurtosis  
## Weight 1.1088548 0.87680061  
## L1 0.6170594 0.41022666  
## L2 0.5731405 0.37843479  
## L3 0.4232795 0.03046029  
## Height 0.1267317 -1.01732434

newfish.qda<-qda(Species~.,data=newfish)

newfish.q<-read.csv("~/Desktop/碩一下/多變量/QDA.csv")

newfish.qda<-qda(Species~.,data=newfish.q)

newfish.qda<-qda(Species~Weight+L1+Height+Width+L21+L32,data=newfish.q)   
newfish.qdapred<-predict(newfish.qda,newfish.q)  
table(newfish.q$Species,newfish.qdapred$class)

##   
## bream parki perch pike roach smelt  
## bream 33 0 0 0 0 0  
## parki 0 10 0 0 0 0  
## perch 0 0 54 0 0 0  
## pike 0 0 0 16 0 0  
## roach 0 0 0 0 18 0  
## smelt 0 0 0 0 0 12

predict(newfish.qda,newfish.test)$class

## [1] bream bream perch perch pike smelt smelt parki roach roach perch  
## Levels: bream parki perch pike roach smelt

newfish.qda<-qda(Species~Weight+L1+Height+Width+L21+L32,data=newfish.q,CV=T)   
table(newfish.q$Species,newfish.qda$class)

##   
## bream parki perch pike roach smelt  
## bream 33 0 0 0 0 0  
## parki 0 10 0 0 0 0  
## perch 0 0 54 0 0 0  
## pike 0 0 0 16 0 0  
## roach 0 0 1 0 17 0  
## smelt 0 0 1 0 0 11

# Nearest Neighbor Methdos

k = 3

library(class)  
newfish.knn<-knn(newfish[,2:10],newfish[,2:10],newfish[,"Species"],k=3,prob=T)   
table(newfish$Species,newfish.knn)

## newfish.knn  
## bream parki perch pike roach smelt white  
## bream 30 0 2 0 0 0 1  
## parki 0 6 1 0 3 0 0  
## perch 3 0 47 0 2 1 1  
## pike 1 0 3 11 0 0 1  
## roach 1 0 8 0 8 0 1  
## smelt 0 0 0 0 0 12 0  
## white 0 0 2 0 0 0 3

k = 2

newfish.knn<-knn(newfish[,2:10],newfish[,2:10],newfish[,"Species"],k=2,prob=T)   
table(newfish$Species,newfish.knn)

## newfish.knn  
## bream parki perch pike roach smelt white  
## bream 28 0 2 0 2 0 1  
## parki 0 8 0 0 2 0 0  
## perch 1 0 49 0 3 1 0  
## pike 2 0 2 12 0 0 0  
## roach 2 0 5 0 10 0 1  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

k = 1

newfish.knn<-knn(newfish[,2:10],newfish[,2:10],newfish[,"Species"],k=1,prob=T)   
table(newfish$Species,newfish.knn)

## newfish.knn  
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

newfish1<-newfish[,c(1,2,3,6,8,9)]  
newfish.knncv<-knn.cv(newfish1[,2:6],newfish1[,"Species"],k=1,prob=T)   
table(newfish1$Species,newfish.knncv)

## newfish.knncv  
## bream parki perch pike roach smelt white  
## bream 26 0 4 0 2 0 1  
## parki 1 4 0 0 4 0 1  
## perch 3 0 37 0 11 1 2  
## pike 2 0 4 9 0 0 1  
## roach 2 0 10 0 5 0 1  
## smelt 0 0 0 0 0 12 0  
## white 0 0 3 0 0 0 2

newfish1.test<-newfish.test[,c(1,2,5,7,8)]  
newfish.knntest<-knn(newfish1[,2:6],newfish1.test,newfish1[,"Species"],k=1,prob=T)   
newfish.knntest

## [1] bream bream perch white perch smelt smelt parki perch perch perch  
## attr(,"prob")  
## [1] 1 1 1 1 1 1 1 1 1 1 1  
## Levels: bream parki perch pike roach smelt white

# Logistic Discrimination

Again, let us pretend that all assumptions (including “multivariate normal” and “common covariance matrix” over all classes) are valid (need to check that in practice). In order to use this method in R, we need another package.

library(nnet)  
newfish.logd<-multinom(Species~.,data=newfish,maxit=250)

## # weights: 77 (60 variable)  
## initial value 287.994702   
## iter 10 value 189.100680  
## iter 20 value 82.739762  
## iter 30 value 15.668415  
## iter 40 value 0.165377  
## iter 50 value 0.003851  
## final value 0.000000   
## converged

newfish.logd

## Call:  
## multinom(formula = Species ~ ., data = newfish, maxit = 250)  
##   
## Coefficients:  
## (Intercept) Weight L1 L2 L3 Height  
## parki -29.45533 0.02917110 6.349592 17.8259067 -23.500970 9.645257  
## perch -80.11405 0.16021628 3.267803 56.6489219 -53.765483 6.684178  
## pike 15.22567 -0.05874368 8.093673 0.9753102 -3.095179 -13.084687  
## roach -277.16410 -0.51539078 54.195310 -43.6844449 4.362472 -2.952463  
## smelt 455.64639 0.18459382 29.363751 -20.5072505 -10.290211 -13.228223  
## white -57.01255 0.19991067 -17.467222 31.7667561 -20.454096 -4.118171  
## Width L21 L32 L31  
## parki 3.247584 11.476314 -41.326877 -29.850563  
## perch 21.052273 53.381119 -110.414404 -57.033286  
## pike 21.652958 -7.118363 -4.070489 -11.188852  
## roach 40.080837 -97.879755 48.046917 -49.832838  
## smelt 18.368009 -49.871001 10.217040 -39.653961  
## white 26.549556 49.233978 -52.220852 -2.986874  
##   
## Residual Deviance: 2.009681e-11   
## AIC: 84

table(newfish$Species,predict(newfish.logd,newfish))

##   
## bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

Now we examine the true error rate by cross-validation. To do this, we use another package called “glmnet”:

library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16

x <- as.matrix(newfish[,-1])  
y <- newfish$Species  
cvfit <- cv.glmnet(x, y, family='multinomial', type.measure='class', nfolds=148)   
predict.value <- predict(cvfit, x, s = "lambda.min", type = "class")  
table(predict.value,newfish$Species)

##   
## predict.value bream parki perch pike roach smelt white  
## bream 33 0 0 0 0 0 0  
## parki 0 10 0 0 0 0 0  
## perch 0 0 54 0 0 0 0  
## pike 0 0 0 16 0 0 0  
## roach 0 0 0 0 18 0 0  
## smelt 0 0 0 0 0 12 0  
## white 0 0 0 0 0 0 5

predict(newfish.logd,newfish.test)

## [1] bream bream perch perch pike smelt smelt parki roach roach white  
## Levels: bream parki perch pike roach smelt white