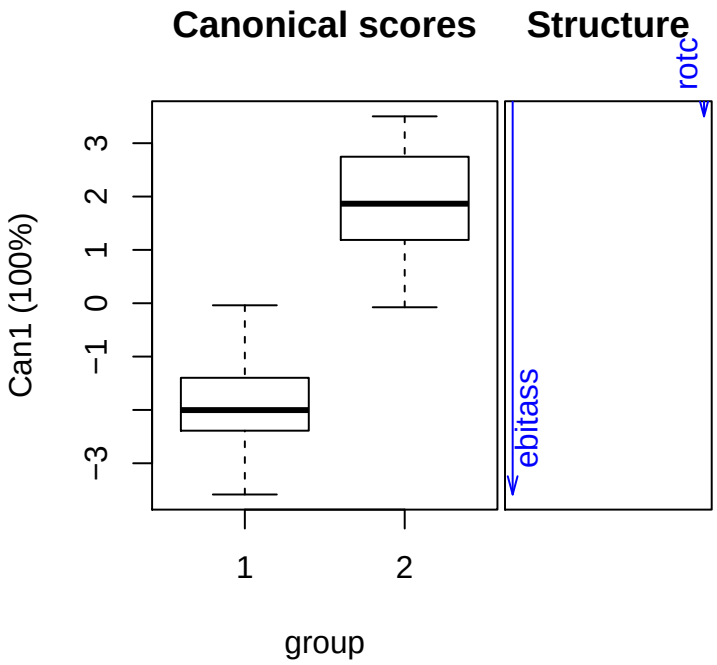
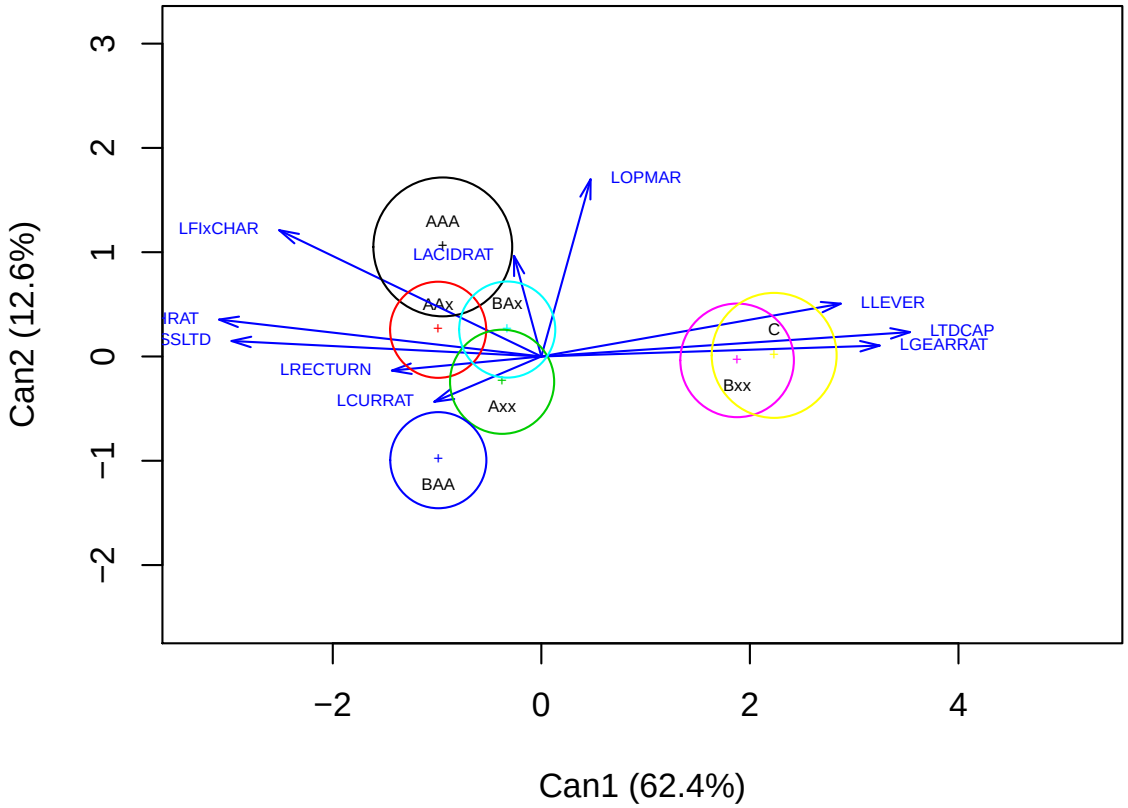
DISCRIMINANT ANALYSIS

```
> ##########################################################################
> #canonical discriminant analysis with 2 groups
> ##########################################################################
> library(heplots)
> library(candisc)
> ddiscr <- read.table("C:/R/rmmva/data STEPDISCR SHARMA.txt", header=T, quote="\"")
> attach(ddiscr)
> ddiscr

   group mktbook   rotc    roe  reass ebitass
1      1   2.304  0.182  0.191  0.377   0.158
2      1   2.703  0.206  0.205  0.469   0.210
3      1   2.385  0.188  0.182  0.581   0.207
4      1   5.981  0.236  0.258  0.491   0.280
5      1   2.762  0.193  0.178  0.587   0.197
6      1   2.984  0.173  0.178  0.546   0.227
7      1   2.070  0.196  0.178  0.443   0.148
8      1   2.762  0.212  0.219  0.472   0.254
9      1   1.345  0.147  0.148  0.297   0.079
10     1   1.716  0.128  0.118  0.597   0.149
11     1   3.000  0.150  0.157  0.530   0.200
12     1   3.006  0.191  0.194  0.575   0.187
13     2   0.975 -0.031 -0.280  0.105  -0.012
14     2   0.945  0.053  0.019  0.306   0.036
15     2   0.270  0.036  0.012  0.269   0.038
16     2   0.739 -0.074 -0.150  0.204  -0.063
17     2   0.833 -0.119 -0.358  0.155  -0.054
18     2   0.716 -0.005 -0.305  0.027   0.000
19     2   0.574  0.039 -0.042  0.268   0.005
20     2   0.800  0.122  0.080  0.339   0.091
21     2   2.028 -0.072 -0.836 -0.185  -0.036
22     2   1.225  0.064 -0.430 -0.057   0.045
23     2   1.502 -0.024 -0.545 -0.050  -0.026
24     2   0.714  0.026 -0.110  0.021   0.016

> cand.mod <- lm(cbind(ebitass, rotc)~ group, data=ddiscr)
> cand.out <-candisc(cand.mod, data=ddiscr)

> plot(cand.out,cex=0.25,cex.lab=0.5)
```




```
> cand.out$eigenvalues

[1]  4.1239e+00 -1.2490e-16

> cand.out$canrsq

[1] 0.80484

> cand.out$coeffs.raw

            Can1
ebitass -15.0919
rotc     -5.7685

> cand.out$coeffs.std

            Can1
ebitass -0.74337
rotc    -0.30547

> cand.out$scores[1:5,]

  group    Can1
1     1 -1.4326
2     1 -2.3558
3     1 -2.2067
4     1 -3.5853
5     1 -2.0846
```

```
> ##############################################################################
> #canonical discriminant analysis with multiple groups
> ##############################################################################
> training <- read.table("C:/R/rmmva/training data discrim in R.txt", header=T, quote="\"")
> attach(training)
> cand.mod <- lm(cbind(LOPMAR,LFIxCHAR,LGEARRAT,LTDCAP,LLEVER,LCASHRAT,
+          LACIDRAT,LCURRAT,LRECTURN ,LASSLTD)~ RATING, data=training)
> cand.out <-candisc(cand.mod, data=training)

> plot(cand.out, type="n",scale=4)
```




```
> cand.out$eigenvalues

 [1]  1.6912e+00  3.4099e-01  2.6797e-01  1.9603e-01  1.6500e-01  4.9405e-02  9.9297e-15
 [8]  1.1861e-15 -1.5733e-16 -1.6638e-15

> cand.out$canrsq

[1] 0.628422 0.254281 0.211338 0.163902 0.141633 0.047079

> cand.out$coeffs.raw
```

```
             Can1      Can2     Can3      Can4      Can5      Can6
LOPMAR   -0.73584   3.07160 -1.09554  -0.60409   0.64560   1.06205
LFIxCHAR  0.33488   0.93718  2.04368   0.50465   0.49612  -0.44035
LGEARRAT  2.74688  18.34874  3.81957 -33.48078  -4.20339   8.06266
LTDCAP   27.09014 -19.03845  1.52933  24.89427 -13.36769 -22.40470
LLEVER   -5.55828  -7.20063 -4.94711  16.99502   5.30011  -4.16357
LCASHRAT -0.83967  -3.57219 -0.60746  -0.17672  -2.87384  -2.13627
LACIDRAT -0.24914   3.41665 -0.69228   0.32935  -0.42113   0.89010
LCURRAT  -2.04495  -2.17102 -1.15283  -0.51191   3.09711  -3.20162
LRECTURN -1.10688   2.35793 -0.64643  -0.67772  -0.14683   0.88824
LASSLTD   5.88932  18.81681  0.89984 -28.30446  -3.34119   5.16190
```

```
> ########################################################################
> #stepwise selection
> ########################################################################
>
> #install.packages("klaR")
> library(klaR)
> training <- read.table("C:/R/rmmva/training data discrim in R.txt", header=T, quote="\"")
> attach(training)
> selection <- greedy.wilks(RATING ~ LOPMAR+ LFIxCHAR +LGEARRAT+ LTDCAP
+                           + LLEVER+ LCASHRAT + LACIDRAT +LCURRAT
+                           +LRECTURN +LASSLTD,niveau = 0.20)
> selection

Formula containing included variables:

RATING ~ LTDCAP + LGEARRAT + LFIxCHAR + LCASHRAT
<environment: 0x000000000947d1e0>


Values calculated in each step of the selection procedure:

      vars Wilks.lambda F.statistics.overall p.value.overall F.statistics.diff
1   LTDCAP      0.50458              12.1095      1.9562e-09           12.1095
2 LGEARRAT      0.42800               6.4305      4.2064e-09            2.1768
3 LFIxCHAR      0.35979               4.9373      3.4038e-09            2.2753
4 LCASHRAT      0.30874               4.1544      3.8332e-09            1.9566
  p.value.diff
1   1.9562e-09
2   5.4684e-02
3   4.5481e-02
4   8.3235e-02

> ########################################################################
> #linear and quadratic classification functions
> ########################################################################
> library(MASS)
> ddiscr <- read.table("C:/R/rmmva/data STEPDISCR SHARMA.txt", header=T, quote="\"")
> attach(ddiscr)
> #lda uses by default priors proportional to sample sizes
> fitl<- lda(group ~ ebitass + rotc,prior=c(1,1)/2)
> fitl
```

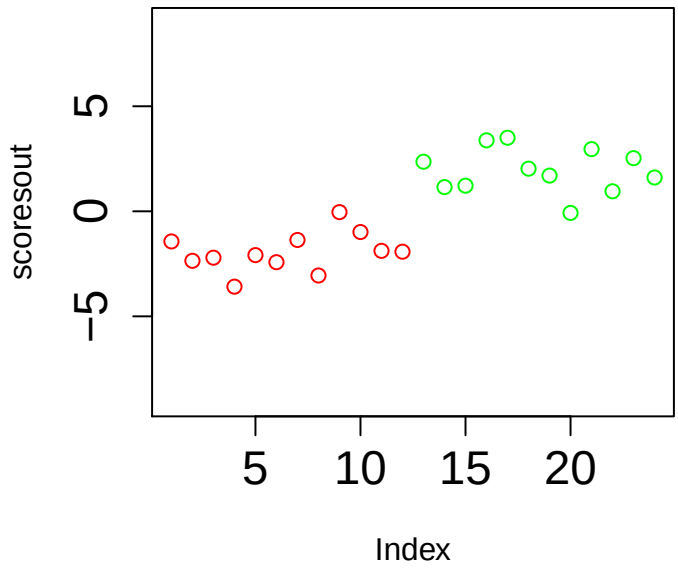
```
Call:
lda(group ~ ebitass + rotc, prior = c(1, 1)/2)

Prior probabilities of groups:
  1   2
0.5 0.5

Group means:
    ebitass    rotc
1 0.1913333 0.18350
2 0.0033333 0.00125

Coefficients of linear discriminants:
             LD1
ebitass -15.0919
rotc     -5.7685
```

```
> pp <- predict(fitl,ddiscr)
> table(group,pp$class )
```

```
group  1  2
    1 12  0
    2  1 11
```

```
> scoresout <-pp$x
```

```
> plot(scoresout, col=rainbow(3)[group], asp=1,cex.axis=1.5)
```



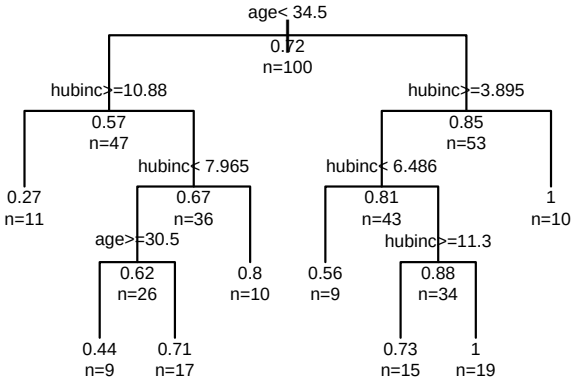

```
> fitq<- qda(group ~ ebitass + rotc,prior=c(9,1)/10,CV=T)
> fitq$class
```

```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2 2 2
Levels: 1 2

> fitq$posterior[1:10,]

         1          2
1  0.99996 3.7200e-05
2  1.00000 1.3842e-11
3  1.00000 3.0515e-13
4  1.00000 1.2232e-25
5  1.00000 1.8239e-10
6  1.00000 1.1134e-20
7  0.99678 3.2184e-03
8  1.00000 3.0345e-22
9  0.65731 3.4269e-01
10 1.00000 7.4754e-07

> table(group,fitq$class )

group  1  2
    1 12  0
    2  1 11

> ###########################################################################
> #Support vector machines with ebitass data
> ###########################################################################
>
> library(e1071)
> library(rpart)
> ddiscr <- read.table("C:/R/rmmva/data STEPDISCR SHARMA.txt", header=T, quote="\"")
> attach(ddiscr)
> svm.model <- svm(as.factor(group) ~ rotc+  ebitass, data = ddiscr, cost = 100, gamma = 1)
> svm.pred <- predict(svm.model, ddiscr)
> table(svm.pred,ddiscr[,1])

svm.pred  1  2
       1 12  0
       2  0 12

> ###########################################################################
> #Support vector machines with with rating data
> ###########################################################################
>
> library(e1071)
> library(rpart)
> ddiscrm <- read.table("C:/R/rmmva/data discrim in R.txt", header=T, quote="\"")
> attach(ddiscrm)
> #cost = the regularization parameter
> #gamma = parameter in kernel
> svm.model <- svm(as.factor(RATING) ~ LOPMAR+ LFIxCHAR +LGEARRAT+ LTDCAP+ LLEVER
+                  + LCASHRAT + LACIDRAT +LCURRAT +LRECTURN +LASSLTD,
+                   data = ddiscrm, cost = 100, gamma = 1)
> svm.pred <- predict(svm.model, ddiscrm)
> svm.model
```

```
Call:
svm(formula = as.factor(RATING) ~ LOPMAR + LFIxCHAR + LGEARRAT + LTDCAP +
    LLEVER + LCASHRAT + LACIDRAT + LCURRAT + LRECTURN + LASSLTD, data = ddiscrm,
    cost = 100, gamma = 1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  100
      gamma:  1

Number of Support Vectors:  95

> svm.pred

  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22
AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAA AAx AAx AAx AAx AAx AAx AAx AAx AAx AAx AAx
 23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44
AAx AAx AAx AAx Axx Axx Axx Axx Axx Axx Axx Axx Axx Axx Axx Axx Axx Axx Axx BAA BAA BAA BAA
 45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66
BAA BAA BAA BAA BAA BAA BAA BAA BAA BAA BAA BAx BAx BAx BAx BAx BAx BAx BAx BAx BAx BAx
 67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88
BAx BAx BAx BAx Bxx Bxx Bxx Bxx Bxx Bxx Bxx Bxx Bxx Bxx Bxx Bxx Bxx   C   C   C   C   C
 89  90  91  92  93  94  95
  C   C   C   C   C   C   C
Levels: AAA AAx Axx BAA BAx Bxx C

> table(svm.pred,ddiscrm[,1])

svm.pred AAA AAx Axx BAA BAx Bxx  C
     AAA  11   0   0   0   0   0  0
     AAx   0  15   0   0   0   0  0
     Axx   0   0  14   0   0   0  0
     BAA   0   0   0  15   0   0  0
     BAx   0   0   0   0  15   0  0
     Bxx   0   0   0   0   0  13  0
     C     0   0   0   0   0   0 12

> ###########################################################################
> #classification trees
> ###########################################################################
> par(mar=c(0.2, 0.2, 0.2, 0))
> par(oma=c(0, 3, 3, 0))
> workdatact <- read.table("C:/R/rmmva/workdata.txt", header=T, quote="\"")
> attach(workdatact)
> #classification trees
> library(rpart)
> tree <- rpart(thisyr ~ child1 +child2 +black +hubinc+ educ+ age,parms=list(split="gini"))
> tree

n= 100

node), split, n, deviance, yval
```

```
      * denotes terminal node

 1) root 100 20.1600 0.72000
   2) age< 34.5 47 11.4890 0.57447
     4) hubinc>=10.885 11  2.1818 0.27273 *
     5) hubinc< 10.885 36  8.0000 0.66667
      10) hubinc< 7.9645 26  6.1538 0.61538
        20) age>=30.5 9  2.2222 0.44444 *
        21) age< 30.5 17  3.5294 0.70588 *
      11) hubinc>=7.9645 10  1.6000 0.80000 *
   3) age>=34.5 53  6.7925 0.84906
     6) hubinc>=3.895 43  6.5116 0.81395
      12) hubinc< 6.486 9  2.2222 0.55556 *
      13) hubinc>=6.486 34  3.5294 0.88235
        26) hubinc>=11.296 15  2.9333 0.73333 *
        27) hubinc< 11.296 19  0.0000 1.00000 *
     7) hubinc< 3.895 10  0.0000 1.00000 *

> printcp(tree) #display the crossvalidation results

Regression tree:
rpart(formula = thisyr ~ child1 + child2 + black + hubinc + educ +
    age, parms = list(split = "gini"))

Variables actually used in tree construction:
[1] age    hubinc

Root node error: 20.2/100 = 0.202

n= 100

      CP nsplit rel error xerror  xstd
1 0.0932      0     1.000   1.02 0.100
2 0.0649      1     0.907   1.07 0.119
3 0.0258      2     0.842   1.09 0.123
4 0.0161      5     0.761   1.21 0.136
5 0.0100      7     0.729   1.22 0.137

> # summary(tree) #display more details about the tree construction

> plot(tree,uniform = TRUE, margin = 0.20,main ="working mom classification tree" )
> text(tree, use.n = TRUE, all=TRUE, cex=0.5)
```

working mom classification tree




```
> smallertree <- prune(tree,cp=0.041)
> smallertree

n= 100

node), split, n, deviance, yval
      * denotes terminal node

1) root 100 20.1600 0.72000
  2) age< 34.5 47 11.4890 0.57447
    4) hubinc>=10.885 11  2.1818 0.27273 *
    5) hubinc< 10.885 36  8.0000 0.66667 *
  3) age>=34.5 53  6.7925 0.84906 *
```