Chapter 4 - 判别分析

张笑竹 / 201618070114

2018年11月2日

第四章实验采用课本例4.1，例4.2和例4.3中给出的数据进行判别分析。例4.1是著名的Fisher鸢尾花数据；例4.2是2012年全国各地区农村居民家庭人均消费支出情况；例4.3为2005年全国城镇居民月平均消费状况。
将本次的实验任务拆分如下：
1）利用MASS包中的函数，对例4.1进行Fisher判别分析；
2）自行编写程序，对例4.2进行距离判别分析；
3）按照课本步骤，自行编写代码，对例4.3进行Fisher判别分析；
4）利用Bayes判别函数，对例4.1进行Bayes判别分析。

1 例4.1 - Fisher判别分析

1.1 预判断：方差分析

首先，在R中读入例4.1的数据集。

```
X = read.csv("eg4-1.csv",header = T)
```

为了更好地进行判别分析，计算各个变量的均值，并对数据进行多元方差分析。

```
attach(X)
aggregate(cbind(Sepal.Length,Sepal.Width,Petal.Length,Petal.Width),by=list(y),FUN=mean)
```

```
##   Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1       1        5.006       3.428        1.462       0.246
## 2       2        5.936       2.770        4.260       1.326
## 3       3        6.588       2.974        5.552       2.026
```

```
X.1 <- as.data.frame(cbind(Sepal.Length,Sepal.Width,Petal.Length,Petal.Width,y))
type <- as.factor(X.1$y)
fit <- manova(cbind(Sepal.Length,Sepal.Width,Petal.Length,Petal.Width)~type,data=X.1)
summary(fit,test=c("Wilks"))
```

```
##            Df    Wilks approx F num Df den Df    Pr(>F)
## type        2 0.023439   199.15      8    288 < 2.2e-16 ***
## Residuals 147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

根据结果不难看出，方差分析的p-value远小于显著性水平，说明各个总体之间的差异极为显著。所以，进行判别分析是可行且必要的。

1.2 利用lda（）函数进行判别

首先加载MASS包。利用MASS包中已经存在的函数，进行Fisher线性判别分析，并输出相应结果。

```
library(MASS)
ld=lda(y~.,X)
ld
```

```
## Call:
## lda(y ~ ., data = X)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3333333 0.3333333 0.3333333
##
## Group means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1        5.006       3.428        1.462       0.246
## 2        5.936       2.770        4.260       1.326
## 3        6.588       2.974        5.552       2.026
##
## Coefficients of linear discriminants:
##                     LD1         LD2
## Sepal.Length  0.8293776  0.02410215
## Sepal.Width   1.5344731  2.16452123
## Petal.Length -2.2012117 -0.93192121
## Petal.Width  -2.8104603  2.83918785
##
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

```
ld$svd
```

```
## [1] 48.642644  4.579983
```

输出的结果提供了Fisher判别的主要信息。“Prior probabilities of groups”为先验概率，此处默认为各个组别的频率。如果提前设定，则此处近乎等价为Bayes判别。而“Group means”与上述方差分析结果一致。“Coefficients of linear discriminants”给出了两个判别函数的系数。“Proportion of trace”给出了两个判别函数判别效率$\Delta(\cdot)$的占比。事实上，判别函数效率为$|B-\lambda E|=0$的特征根，而系数向量就是上述特征根对应的特征向量。此外，还输出了“svd”项，即对两个判别函数进行假设检验时，F统计量的平方根。因此，判别函数表示为：

$$y_1 = 0.829 \cdot x_1 + 1.534 \cdot x_2 - 2.201 \cdot x_3 - 2.810 \cdot x_4$$

$$y_2 = 0.024 \cdot x_1 + 2.165 \cdot x_2 - 0.932 \cdot x_3 + 2.839 \cdot x_4$$

其中，x_1表示Sepal.Length，x_2表示Sepal.Width，x_3表示Petal.Length，x_4表示Petal.Width.
判别函数LD1的判别效率较大(99.12%)，LD2的判别效率较小(0.88%)。根据判别函数就可以计算每一个样本的得分，并对样本进行分类。

1.3 回判和预测

1.3.1 回判

首先，利用predict()指令，输出上述判别分析模型生成的新分类。然后，制作原分类与新分类的列联表，进

而观测误判情况，并计算判别的正确率。

```
X.pre=predict(ld)
newy=X.pre$class
newy
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [71] 3 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3
## [106] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3
## [141] 3 3 3 3 3 3 3 3 3 3
## Levels: 1 2 3
```

```
tab=table(y,newy)
tab
```

```
##    newy
## y    1  2  3
##   1 50  0  0
##   2  0 48  2
##   3  0  1 49
```

```
sum(diag(prop.table(tab)))
```

```
## [1] 0.98
```

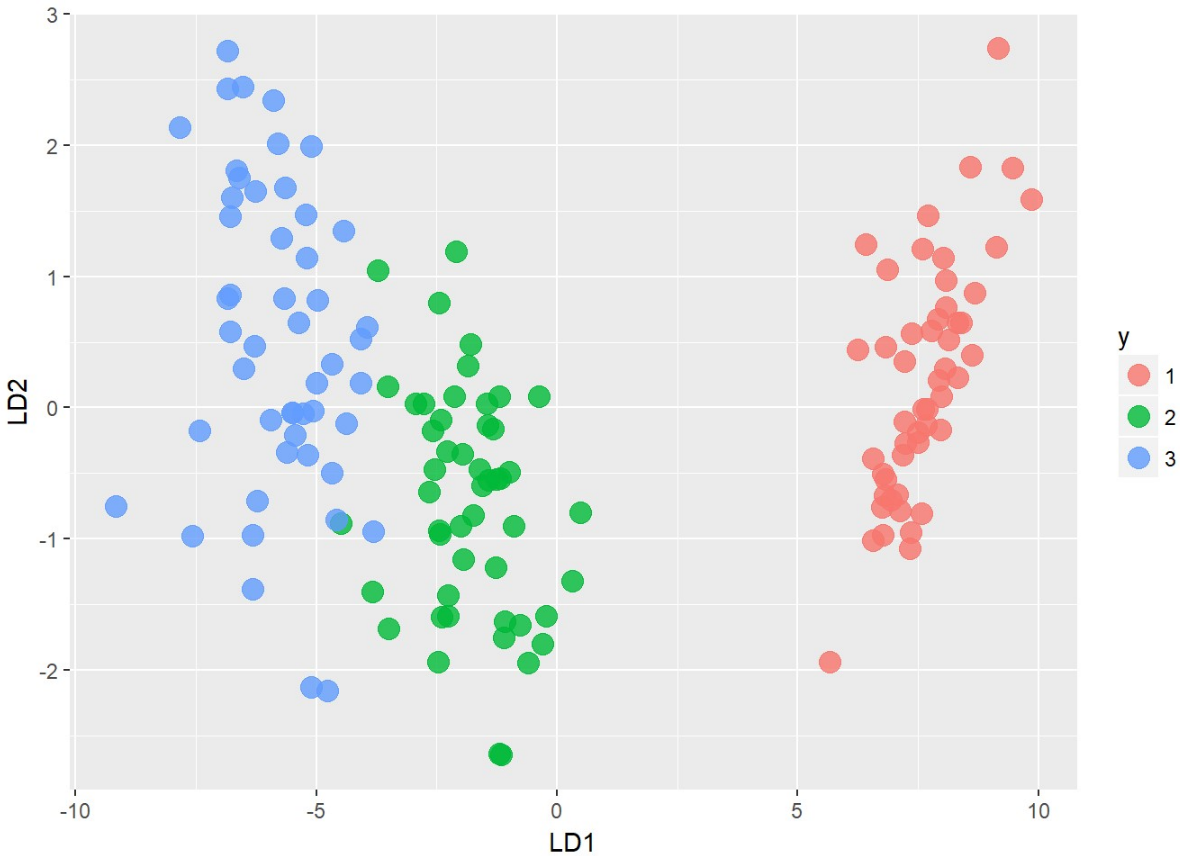
通过列联表可以看出，有2个类别二的样本被误判入类别三，1个类别三的样本被误判入类别二，而其他样本判断正确。计算列联表每个单元格内的频率，并将对角线上的频率相加，就可以得到判别的正确率。根据结果，98%的样本都被判对，认为正确率非常高。

此外，可以分别以两个判别函数值为横轴、纵轴，输出分类结果图。

```
#分类结果图
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
y <- as.factor(X$y)
X.2=cbind(y,as.data.frame(X.pre$x))
p=ggplot((X.2),aes(x=LD1,y=LD2))
p+geom_point(aes(colour=y),alpha=0.8,size=4)
```



从图中可以看到，LD1函数值为横轴，LD2函数值为纵轴。由于横轴代表的LD1判别效率(99.12%)较大，故样本向下投影后区分度较好；而纵轴代表的LD2判别效率(0.88%)较小，故向左投影后区分度较差。此外，类别一和类别二分类较为清晰，而类别二和类别三存在重合区域，即可能存在误判。

1.3.2 预测

假若现在有一朵鸢尾花，参数如下：

符号	变量	数值
x_1	Sepal.Length	5.8
x_2	Sepal.Width	3.1
x_3	Petal.Length	3.8
x_4	Petal.Width	1.2

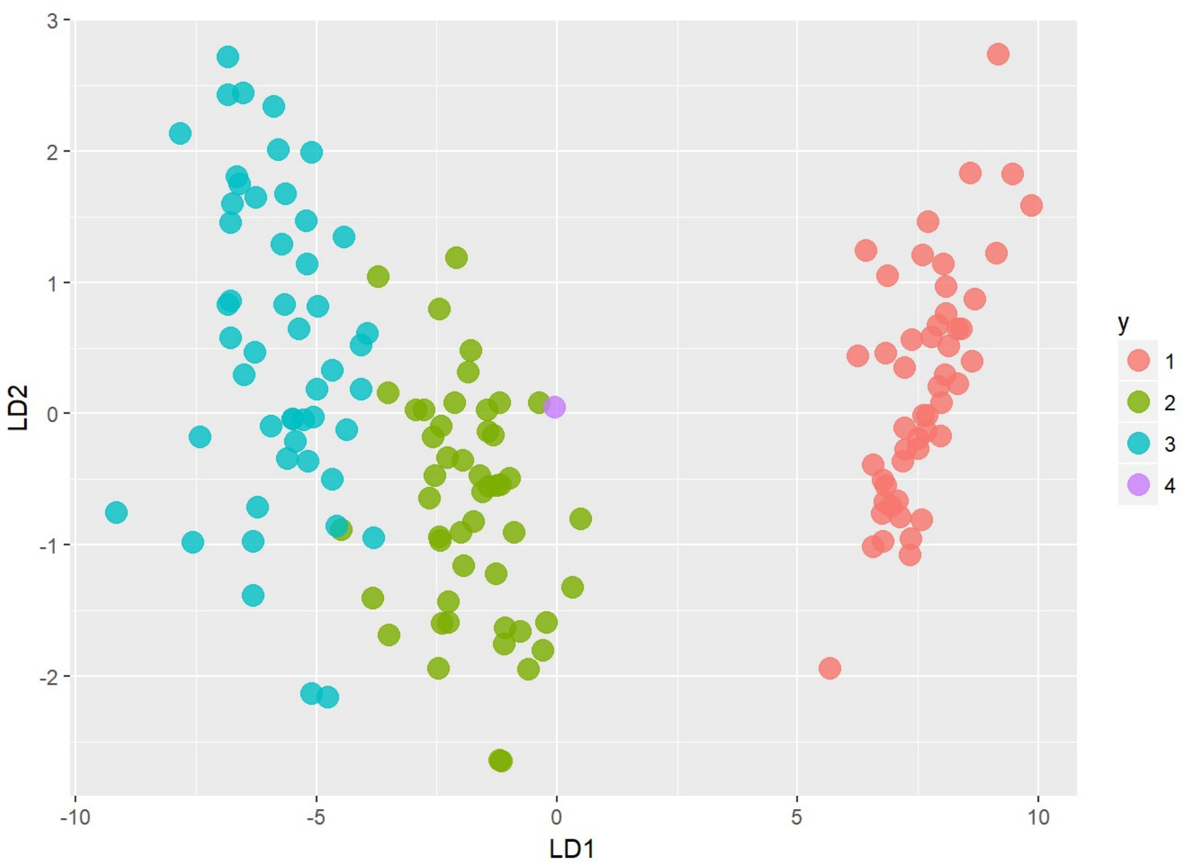
那么，容易得到预测结果如下：

```
pre=predict(ld,data.frame(Sepal.Length=5.8,Sepal.Width=3.1,Petal.Length=3.8,Petal.Width=1.2))
pre
```

```
## $class
## [1] 2
## Levels: 1 2 3
##
## $posterior
##              1         2            3
## 1 1.04104e-12 0.9999995 4.579081e-07
##
## $x
##            LD1        LD2
## 1 -0.06479338 0.05406058
```

并绘制图形：

```
pre.1=cbind("4",as.data.frame(pre$x))
colnames(pre.1)=colnames(X.2)
y<-factor(y,levels = c(1,2,3,4))
pre.2=rbind(X.2[1:nrow(X.2),],pre.1)
p=ggplot(cbind(pre.2),aes(x=LD1,y=LD2))
p+geom_point(aes(colour=y),alpha=0.8,size=4)
```



图形中的紫色圆点，记为类别“4”，就是预测点的位置。不难发现，紫色圆点的坐标就是它的(LD1,LD2)判别函数值。这个紫色原点离类别二的重心最近，所以被判入了类别二。

2 例4.2 – 距离判别分析

2.1 距离的计算

首先，在R中读入例4.2的数据集，并对数据集进行处理，将“地区”名称作为行名。

```
Z = read.csv("eg4-2.csv",header = T)
rownames(Z)=Z[,2]
Z=Z[,-1]
Z=Z[,-1]
```

然后，按既有的组别，将数据集进行拆分，并计算每个组别的样本个数。

```
attach(Z)
r=3
G1 <- subset(Z,Group=="1")
G2 <- subset(Z,Group=="2")
G3 <- subset(Z,Group=="3")
n1=nrow(G1)
n2=nrow(G2)
n3=nrow(G3)
n=n1+n2+n3
```

再次，计算总体的协方差矩阵、它的逆矩阵，以及每个组别的均值向量。

```
L1=(n1-1)*cov(G1[,1:8])
L2=(n2-1)*cov(G2[,1:8])
L3=(n3-1)*cov(G3[,1:8])
L=L1+L2+L3
Sp=L/(n-r)
Sp_1=solve(Sp)

m1=colMeans(G1[,1:8])
m2=colMeans(G2[,1:8])
m3=colMeans(G3[,1:8])
```

最后，通过循环指令，计算所有的31个样本距离3个组别的马氏距离，并进行存储。

```
d1=c()
for (i in 1:nrow(Z)){
  dis1=as.matrix(Z[i,1:8]-m1)%*%as.matrix(Sp_1)%*%as.matrix(t(Z[i,1:8]-m1))
  d1=c(d1,dis1)
}

d2=c()
for (i in 1:nrow(Z)){
  dis2=as.matrix(Z[i,1:8]-m2)%*%as.matrix(Sp_1)%*%as.matrix(t(Z[i,1:8]-m2))
  d2=c(d2,dis2)
}

d3=c()
for (i in 1:nrow(Z)){
  dis3=as.matrix(Z[i,1:8]-m3)%*%as.matrix(Sp_1)%*%as.matrix(t(Z[i,1:8]-m3))
  d3=c(d3,dis3)
}
```

2.2 预测及回判

距离判别法的思想在于，通过比较某一个样本点距离所有组别的距离大小，将其判入距离最短的组别。因此，可以通过循环指令和if指令，对样本进行判别，并存储它们到所属组别的距离。

```
#进行预测及回判
newG=c()
dist=c()
for(z in 1:nrow(Z)){
  if(d1[z]==min(d1[z],d2[z],d3[z])){
    newG=c(newG,1)
  }
  if(d2[z]==min(d1[z],d2[z],d3[z])){
    newG=c(newG,2)
  }
  if(d3[z]==min(d1[z],d2[z],d3[z])){
    newG=c(newG,3)
  }
  dist=c(dist,min(d1[z],d2[z],d3[z]))
}
```

可以得到结果如下。

```
Z.1=data.frame(Z$Group,dist,newG)
rownames(Z.1)=rownames(Z)
colnames(Z.1)=c("原组别","距离","现组别")
Z.1
```

```
##          原组别      距离  现组别
## 上  海      1 12.419685      1
## 北  京      1 12.163187      1
## 广  东      1 11.387575      1
## 浙  江      1  3.503847      1
## 江  苏      1 16.463007      1
## 安  徽      2  1.145772      2
## 天  津      2 13.216134      2
## 江  西      2  2.037411      2
## 山  东      2  5.629111      2
## 湖  北      2  7.797147      2
## 湖  南      2  1.858969      2
## 广  西      2  4.392380      2
## 海  南      2 10.045045      2
## 重  庆      2 11.096831      2
## 四  川      2  3.032020      2
## 贵  州      2  2.691399      2
## 云  南      2  3.525650      2
## 西  藏      2 17.277215      3
## 河  北      3  3.680968      3
## 山  西      3  4.642705      3
## 内蒙古      3  5.868870      3
## 辽  宁      3  5.117014      3
## 吉  林      3 12.042978      3
## 黑龙江      3  9.573854      3
## 河  南      3  3.374577      2
## 甘  肃      3  2.630002      2
## 青  海      3  4.984155      3
## 宁  夏      3  3.619441      3
## 新  疆      3  8.734159      3
## 福  建     NA 14.766885      1
## 陕  西     NA  6.458672      3
```

福建被判入组别一，陕西被判入组别三。观察三个组别的省份之间经济发展差异，可以认为，由于福建经济较为发达，而陕西发展程度弱一些，因此这样的判别结果符合常识。

最后，计算判别的正确率。

```
Z.pre=cbind(Z,newG)
tab=table(Z.pre$Group[1:n],Z.pre$newG[1:n])
tab
```

```
##
##      1  2  3
##   1  5  0  0
##   2  0 12  1
##   3  0  2  9
```

```
sum(diag(prop.table(tab)))
```

```
## [1] 0.8965517
```

根据输出的列联表，有1个类别二的样本被误判入类别三，2个类别三的样本被误判入类别二，而其他样本判断正确。正确率为89.66%，判断十分准确。

3 例4.3 - Fisher判别分析

3.1 判别函数的计算与检验

首先，在R中读入例4.3的数据集，并对数据集进行处理，将“地区”名称作为行名。然后按既有的组别将数据集进行拆分，并计算各个组别的样本数量。

```
Y = read.csv("eg4-3.csv",header = T)
rownames(Y)=Y[,2]
Y=Y[,-1]
Y=Y[,-1]

#筛选出总体G1和G2
attach(Y)
```

```
## The following objects are masked from Z:
##
##     Group, x1, x2, x3, x4, x5, x6, x7, x8
```

```
G1 <- subset(Y,Group=="1")
G2 <- subset(Y,Group=="2")
n1=nrow(G1)
n2=nrow(G2)
n=n1+n2
p=8
r=2
```

之后，严格按照课本提供的步骤，对例4.3进行Fisher线性判别。
Step1：计算总体G_1和G_2各判别变量的均值。

```
x1 <- colMeans(G1[,1:8])
x2 <- colMeans(G2[,1:8])
x1
```

```
##        x1        x2        x3        x4        x5        x6        x7
##  20.79667 145.27333  39.85667  64.94667  89.70000  16.32000  49.43667
##        x8
## 417.00667
```

```
x2
```

```
##        x1        x2        x3        x4        x5        x6        x7
##  19.92885  98.53962  21.48077  35.50962  59.80192  10.49000  39.99462
##        x8
## 184.91308
```

```
x1-x2
```

```
##          x1          x2          x3          x4          x5          x6
##   0.8678205  46.7337179  18.3758974  29.4370513  29.8980769   5.8300000
##          x7          x8
##   9.4420513 232.0935897
```

```
x1+x2
```

```
##        x1        x2        x3        x4        x5        x6        x7
##  40.72551 243.81295  61.33744 100.45628 149.50192  26.81000  89.43128
##        x8
## 601.91974
```

Step2: 计算协方差阵Σ的估计值S_p的逆矩阵。

```
S1=cov(G1[,1:8])
S2=cov(G2[,1:8])
Sp=(S1*(n1-1)+S2*(n2-1))/(n1+n2-2)
Sp_1=solve(Sp)
Sp_1
```

```
##                x1            x2           x3           x4           x5
## x1  4.343902e-01 -0.0095094111  0.015611100 -0.057824559 -0.024335350
## x2 -9.509411e-03  0.0047647095 -0.005654339  0.007544415  0.006920001
## x3  1.561110e-02 -0.0056543386  0.049097071 -0.019358293 -0.009423525
## x4 -5.782456e-02  0.0075444153 -0.019358293  0.076678632  0.010932869
## x5 -2.433535e-02  0.0069200010 -0.009423525  0.010932869  0.018551516
## x6  1.531215e-01 -0.0210477112  0.069887950 -0.098904730 -0.028626160
## x7 -1.222262e-02 -0.0073141406  0.013678775 -0.014492944 -0.010972788
## x8  8.061192e-06 -0.0009952292 -0.001488133 -0.006984818 -0.001559991
##               x6           x7           x8
## x1  0.153121477 -0.012222621  8.061192e-06
## x2 -0.021047711 -0.007314141 -9.952292e-04
## x3  0.069887950  0.013678775 -1.488133e-03
## x4 -0.098904730 -0.014492944 -6.984818e-03
## x5 -0.028626160 -0.010972788 -1.559991e-03
## x6  0.460759222  0.057386303 -3.131191e-03
## x7  0.057386303  0.048407057 -1.767405e-03
## x8 -0.003131191 -0.001767405  2.482779e-03
```

Step3: 计算Fisher样本判别函数。

```
alpha=t(x1-x2)%*%Sp_1
alpha
```

```
##             x1        x2           x3        x4       x5          x6
## [1,] -1.431172 0.1167417 -0.008877874 0.1961467 0.373039 -0.8324913
##              x7        x8
## [1,] -0.4743385 0.2151917
```

Step4: 计算两个总体均值的中点m的估计值$\hat{m}$.

```
m=0.5*alpha%*%(x1+x2)
m
```

```
##          [,1]
## [1,] 54.94797
```

Step5: 计算统计检验量F值。
马氏距离为：

```
D2=t(x1-x2)%*%Sp_1%*%(x1-x2)
D2
```

```
##          [,1]
## [1,] 61.59022
```

F检验统计量为以及其p-value为：

```
F=(n1+n2-p-1)/((n1+n2-2)*p)*n1*n2/(n1+n2)*D2
pvalue=pf(F,p,n1+n2-p-1,ncp=0,lower.tail=FALSE,log.p=FALSE)
F
```

```
##          [,1]
## [1,] 15.33856
```

```
pvalue
```

```
##              [,1]
## [1,] 5.556979e-07
```

p-value远小于显著性水平α=0.05，即两个总体的均值存在显著差异，判别函数有效。

3.2 回判及代判样品的归类

由于在Step3中求出的Fisher样本判别函数，实际上为函数的系数向量（行向量），所以为了计算每一个样本的判别函数值y_0,将数据集进行转置；把上述系数向量与转置的数据集相乘，得到的就是所有样本的判别函数值向量（行向量）。随后，通过循环指令，对每一个样本进行判别。Fisher判别法则为：

$$\text{若} y_0 \geq \hat{m} = 54.946, \text{判} x_0 \text{来自总体} G_1;$$

$$\text{若} y_0 < \hat{m} = 54.946, \text{判} x_0 \text{来自总体} G_2.$$

```
T=alpha%*%t(Y[,1:8])
newG=c()
for(i in 1:31){
  if(T[i]>=m){
    newG=c(newG,1)
  }
  if(T[i]<m){
    newG=c(newG,2)
  }
}
```

最终得到结果为：

```
Y.1=data.frame(Y$Group,t(T),newG)
rownames(Y.1)=rownames(Y)
colnames(Y.1)=c("原组别","判别函数值","现组别")
Y.1
```

```
##         原组别 判别函数值 现组别
## 北  京      1   88.33441      1
## 上  海      1   84.52424      1
## 浙  江      1   84.37057      1
## 天  津      2   40.48507      2
## 河  北      2   23.61754      2
## 山  西      2   25.03958      2
## 内蒙古      2   25.61029      2
## 辽  宁      2   21.55622      2
## 吉  林      2   21.73377      2
## 黑龙江      2   18.68919      2
## 江  苏      2   41.33828      2
## 安  徽      2   21.97394      2
## 福  建      2   24.83897      2
## 江  西      2   15.75685      2
## 山  东      2   32.28628      2
## 河  南      2   18.86114      2
## 湖  北      2   21.13571      2
## 湖  南      2   30.83906      2
## 广  西      2   19.35403      2
## 海  南      2   13.72448      2
## 重  庆      2   45.74558      2
## 四  川      2   26.02052      2
## 贵  州      2   12.86920      2
## 云  南      2   24.50681      2
## 陕  西      2   20.00929      2
## 甘  肃      2   24.26689      2
## 青  海      2   19.18603      2
## 宁  夏      2   18.06984      2
## 新  疆      2   20.45972      2
## 广  东     NA   52.92754      2
## 西  藏     NA   30.88363      2
```

可见，代判省份广东、西藏均被判入类别二。
判别正确率为：

```
tab=table(Y$Group[n1+n2],newG[n1+n2])
tab
```

```
##
##      2
##   2 1
```

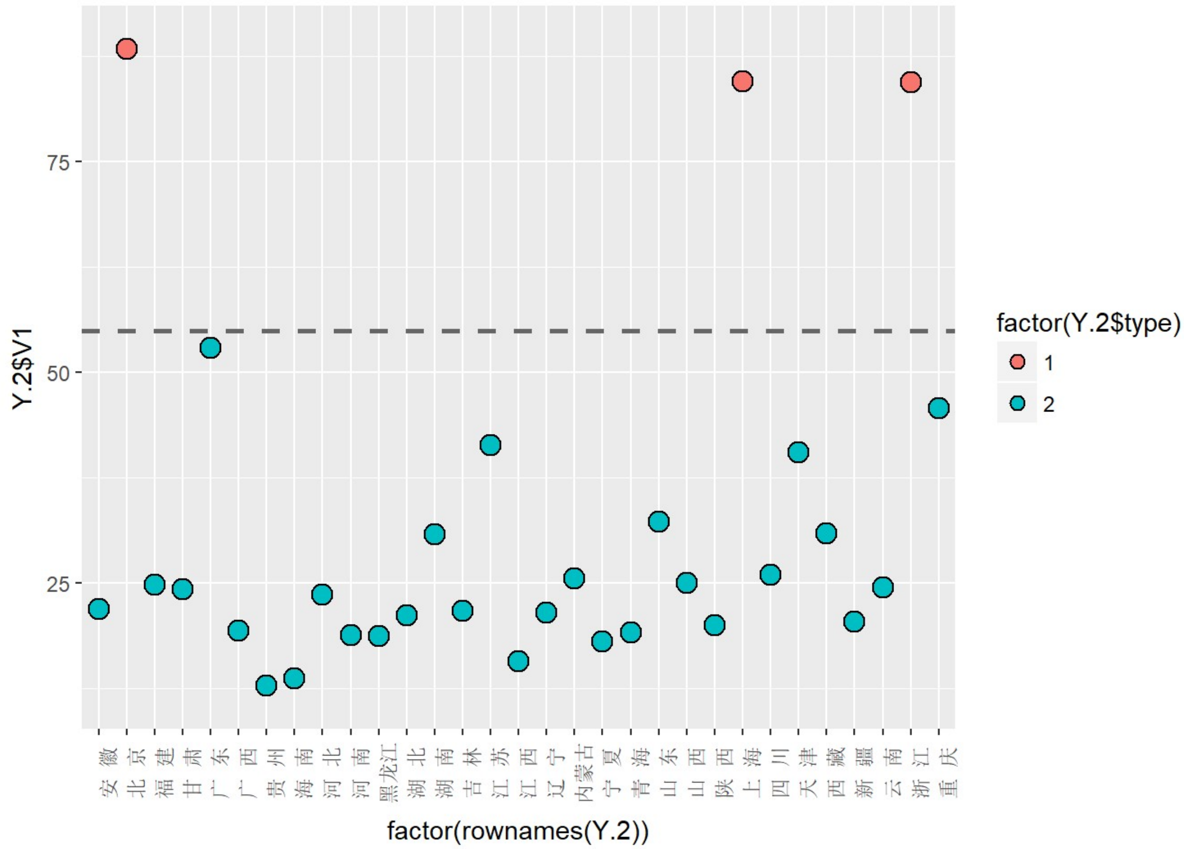
```
sum(diag(prop.table(tab)))
```

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

此外，还可以以各个省份为横轴、判别函数值为纵轴，做出分类结果图。

```
#分类结果图
library(ggplot2)
type <- as.factor(newG)
Y.2=cbind(type,as.data.frame(t(T)))
p<-ggplot(Y.2, aes(x = factor(rownames(Y.2)), fill = factor(Y.2$type), y = Y.2$V1))
q<-geom_dotplot(binaxis = "y", stackdir = "center", position = "dodge")
u<-geom_hline(aes(yintercept = m),color="gray40"  ,linetype="dashed",cex=1)
p+q+u+theme(axis.text.x = element_text(angle = 90))
```

```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



图中灰色虚线表示总体均值的终点$\hat{m}$,即判别的临界值。对于临界值以下的点来说，越接近临界值，居民生活水平越高；尽管广东、江苏、天津、重庆被判入了类别二，但是它们的判别函数值已经十分接近临界值，所以其居民生活水平应该较好。

4 例4.1 - Bayes判别分析

4.1 利用lda（）函数进行Bayes判别

前面提到过，对于MASS包中的lda()函数，当先验概率为各个类别频数（即缺省值）时，进行的是Fisher线性判别。而当另外设定各个总体出现的先验概率时，进行的就是Bayes判别。不妨假定例4.1中的3个总体出现的概率为(0.25,0.5,0.25),利用lda()函数进行Bayes判别。

```
X = read.csv("eg4-1.csv",header = T)
```

```
## Warning in if (!header) rlabp <- FALSE: 条件的长度大于一，因此只能用其第一
## 元素
```

```
## Warning in if (header) {: 条件的长度大于一，因此只能用其第一元素
```

```
attach(X)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     y
```

```
## The following objects are masked from X (pos = 7):
##
##     Petal.Length, Petal.Width, Sepal.Length, Sepal.Width, y
```

```
ld1=lda(y~.,prior=c(1,2,1)/4,X)
pre.by=predict(ld1)
by.1<-data.frame(X$y,pre.by$posterior,pre.by$class)
colnames(by.1)=c("原组别","后验概率G1","后验概率G2","后验概率G3","现组别")
by.1[c(1:5,51:55,101:105),]
```

```
##       原组别    后验概率G1    后验概率G2    后验概率G3  现组别
## 1        1 1.000000e+00 7.792716e-22 2.611168e-42      1
## 2        1 1.000000e+00 1.443594e-17 5.042143e-37      1
## 3        1 1.000000e+00 2.927698e-19 4.675932e-39      1
## 4        1 1.000000e+00 2.537073e-16 3.566610e-35      1
## 5        1 1.000000e+00 3.274775e-22 1.082605e-42      1
## 51       2 9.849203e-19 9.999447e-01 5.529694e-05      2
## 52       2 6.216698e-20 9.996286e-01 3.714027e-04      2
## 53       2 1.046325e-22 9.978991e-01 2.100931e-03      2
## 54       2 1.099646e-22 9.998211e-01 1.788571e-04      2
## 55       2 2.111495e-23 9.977903e-01 2.209699e-03      2
## 101      3 7.503075e-52 1.425461e-08 1.000000e+00      3
## 102      3 5.208186e-38 2.154179e-03 9.978458e-01      3
## 103      3 1.231232e-42 5.185518e-05 9.999481e-01      3
## 104      3 1.535858e-38 2.133999e-03 9.978660e-01      3
## 105      3 6.242489e-46 3.625920e-06 9.999964e-01      3
```

输出的样本是每一个组别的前5个观测。可以看出，lda()函数通过先验概率，计算某一个样本属于总体G_i的后验概率，并将其判入后验概率最大的组别。

然而，将Bayes判别转化为线性判别时，事实上需要满足各类协方差矩阵相等的条件。因此，调用第二章中的编写的协方差检验函数，对$H_0 : \Sigma_1 = \Sigma_2 = \Sigma_3$进行检验。

```r
r=3
G1 <- subset(X,y=="1")
G2 <- subset(X,y=="2")
G3 <- subset(X,y=="3")
n1=nrow(G1)
n2=nrow(G2)
n3=nrow(G3)
n=n1+n2+n3

source("cov_test.R")
cov.test(4,c(1:4))
```

```
## $M
## [1] 146.6632
##
## $b
## [1] 20.13193
##
## $`p-value`
## [1] 4.286489e-21
```

遗憾的是，检验结果p-value几乎接近于零，意味着拒绝原假设，各类协方差阵并不相等。所以，就不得不求助于其他的方法。

4.2 编写程序进行Bayes判别

参考博客https://blog.csdn.net/tiaaaaa/article/details/58145126，可以得到一个两总体的Bayes判别程序，能够处理总体之间方差不相等的情况。

```
discriminiant.bayes<-function(TrnX1, TrnX2, rate=1, TstX = NULL, var.equal = FALSE){
  if (is.null(TstX) == TRUE) TstX<-rbind(TrnX1,TrnX2)
  if (is.vector(TstX) == TRUE)  TstX<-t(as.matrix(TstX))
  else if (is.matrix(TstX) != TRUE)
    TstX<-as.matrix(TstX)
  if (is.matrix(TrnX1) != TRUE) TrnX1<-as.matrix(TrnX1)
  if (is.matrix(TrnX2) != TRUE) TrnX2<-as.matrix(TrnX2)

  nx<-nrow(TstX)
  blong<-matrix(rep(0, nx), nrow=1, byrow=TRUE,
              dimnames=list("blong", 1:nx))
  mu1<-colMeans(TrnX1); mu2<-colMeans(TrnX2)
  if (var.equal == TRUE  || var.equal == T){
    S<-var(rbind(TrnX1,TrnX2)); beta<-2*log(rate)
    w<-mahalanobis(TstX, mu2, S)-mahalanobis(TstX, mu1, S)
  }
  else{
    S1<-var(TrnX1); S2<-var(TrnX2)
    beta<-2*log(rate)+log(det(S1)/det(S2))
    w<-mahalanobis(TstX, mu2, S2)-mahalanobis(TstX, mu1, S1)
  }

  for (i in 1:nx){
    if (w[i]>beta)
      blong[i]<-1
    else
      blong[i]<-2
  }
  blong
}
```

其中

$$rate = \frac{L(1|2)}{L(2|1)} \cdot \frac{p_2}{p_1}$$

缺省值为1.

由于例4.1数据集有3个总体，故这里拆分为3对“两总体”进行检验。不妨仍设3个总体出现的概率为(0.25,0.5,0.25).

```
#总体1&总体2
d.b1<-discriminiant.bayes(G1[,1:4],G2[,1:4],rate=2,var.equal = FALSE)
d.b1
```

```
##       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## blong 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##       27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## blong  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##       50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## blong  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##       73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
## blong  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##       96 97 98 99 100
## blong  2  2  2  2   2
```

```r
table(X$y[1:100],d.b1)
```

```
##    d.b1
##      1  2
##   1 50  0
##   2  0 50
```

```r
sum(diag(prop.table(table(X$y[1:100],d.b1))))
```

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

```r
#总体1&总体3
d.b2<-discriminiant.bayes(G1[,1:4],G3[,1:4],rate=1,var.equal = FALSE)
d.b2
```

```
##       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## blong 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##       27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## blong  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##       50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## blong  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##       73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
## blong  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##       96 97 98 99 100
## blong  2  2  2  2   2
```

```r
table(X$y[1:100],d.b2)
```

```
##    d.b2
##      1  2
##   1 50  0
##   2  0 50
```

```r
sum(diag(prop.table(table(X$y[1:100],d.b2))))
```

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

```r
#总体2&总体3
d.b3<-discriminiant.bayes(G2[,1:4],G3[,1:4],rate=0.5,var.equal = FALSE)
d.b3
```

```
##       1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## blong 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##       27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
## blong  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##       50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## blong  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
##       73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
## blong  2  2  2  2  2  2  2  2  2  2  2  1  2  2  2  2  2  2  2  2  2  2  2
##       96 97 98 99 100
## blong  2  2  2  2   2
```

```
table(X$y[1:100],d.b3)
```

```
##    d.b3
##      1  2
##   1 49  1
##   2  1 49
```

```
sum(diag(prop.table(table(X$y[1:100],d.b3))))
```

```
## [1] 0.98
```

综上，对总体1&总体2进行判别，正确率为100%；对总体1&总体3进行判别，正确率也为100%；而对总体2&总体3进行判别，正确率下降到98%. 事实上，Bayes判别的效果与先验概率关系密切，其判别效果不可一概而论。