Analise Multivariada II

Luciane Alcoforado

setembro de 2018

1 Aula 10 e 11 - Análise Discriminante

2 O que é?

Técnica estatística que auxilia a identificar quais variáveis que diferenciam os grupos e quantas dessas variáveis são necessárias para obter a melhor classificação dos indivíduos de uma determinada população.

3 Modelo Matemático

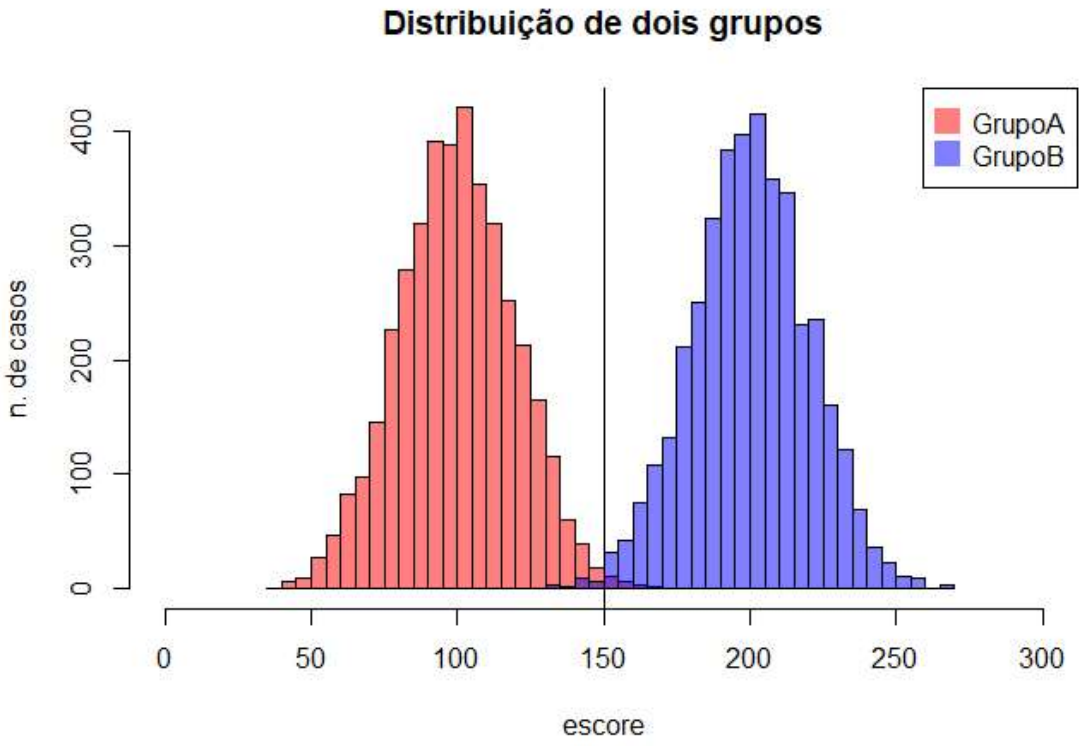
$Z = \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$

Z representa o escore discriminante

β_i é o coeficiente discriminante, ou seja, o peso de cada variável para discriminar os grupos.

X_i são as variáveis independentes do modelo

Para obter a classificação a partir do modelo, compara-se Z a um valor crítico que estabelecerá a qual grupo deve-se classificar um indivíduo da população.



4 Pressupostos

- Normalidade Multivariada
- Linearidade
- Ausência de outliers
- Ausência de Multicolinearidade
- Homogeneidade das matrizes de variância/covariância

5 Exemplo 1

Grupos: Compradores e não compradores

X1: avaliação da durabilidade do produto

X2: avaliação do desempenho do produto

X3: avaliação do estilo do produto

```
## # A tibble: 10 x 4
##       x1    x2    x3 g
##    <dbl> <dbl> <dbl> <fct>
##  1     8     9     6 Compraria
##  2     6     7     5 Compraria
##  3    10     6     3 Compraria
##  4     9     4     4 Compraria
```

```
##  5     4     8     2 Compraria
##  6     5     4     7 NãoCompraria
##  7     3     7     2 NãoCompraria
##  8     4     5     5 NãoCompraria
##  9     2     4     3 NãoCompraria
## 10     2     2     2 NãoCompraria
```

Pontuação média em cada variável por cada grupo:

O modelo deve ser bom o suficiente para captar o que é importante para o comprador ao tomar sua decisão entre comprar ou não o produto. Vemos que entre os compradores as notas maiores foram em X1 (durabilidade) e X2 (desempenho) e para os não compradores a pior nota foi em X1.

5.1 Ajuste do Modelo

No R, ajustamos um modelo LDA usando a função lda, que faz parte da biblioteca MASS. Note que a sintaxe para o lda é idêntica à de lm e à de glm.

```
modelocoef=MASS::lda(g~x1+x2+x3, data=dt)
modelocoef #modelo com os coeficientes
```

```
## Call:
## lda(g ~ x1 + x2 + x3, data = dt)
##
## Prior probabilities of groups:
##    Compraria NãoCompraria
##          0.5          0.5
##
## Group means:
##                x1  x2  x3
## Compraria     7.4 6.8 4.0
## NãoCompraria 3.2 4.4 3.8
##
## Coefficients of linear discriminants:
##        LD1
## x1 -0.5729
## x2 -0.3792
## x3  0.2970
```

O modelo calcula automaticamente as probabilidades a priori (π_0 = 0.5, π_1 = 0.5), 50% das observações são clientes que compram e 50% são clientes que não compram. Temos a média de cada variável dentro de cada grupo, para X1 a média é de 7.4 no grupo 1 e 3.2 no grupo 2, para X2 é 6.8 e 4.4 para cada grupo 1 e 2 respectivamente e para X3 é 4.0 e 3.8 para cada grupo 1 e 2 respectivamente.

Os coeficientes (Coefficients of linear discriminants) proporcionam a combinação das variáveis preditoras para generalizar as diferenças lineares para cada uma das observações. A função discriminante ficou:

$$-0.57 * x_1 - 0.38 * x_2 + 0.30 * x_3$$

O escore discriminante é obtido através desta função discriminante avaliado na distância entre o valor médio e o valor observado. Veremos mais a frente como obter estes valores de escore através da função predict.

```
valormedio=colMeans(modelocoef$means)
valormedio
```

```
##  x1  x2  x3
## 5.3 5.6 3.9
```

```
distancia=dt[,1:3]-matrix(rep(valormedio,10),10,byrow=T)
distancia
```

```
##      x1   x2   x3
## 1   2.7  3.4  2.1
## 2   0.7  1.4  1.1
## 3   4.7  0.4 -0.9
## 4   3.7 -1.6  0.1
## 5  -1.3  2.4 -1.9
## 6  -0.3 -1.6  3.1
## 7  -2.3  1.4 -1.9
## 8  -1.3 -0.6  1.1
## 9  -3.3 -1.6 -0.9
## 10 -3.3 -3.6 -1.9
```

```
escore= as.matrix(distancia)%*%as.matrix(modelocoef$scaling)
escore
```

```
##           LD1
##  [1,] -2.2125
##  [2,] -0.6052
```

```
##  [3,] -3.1119
##  [4,] -1.4834
##  [5,] -0.7297
##  [6,]  1.6995
##  [7,]  0.2225
##  [8,]  1.2991
##  [9,]  2.2302
## [10,]  2.6916
```

A regra de decisão será escore negativo classifica a observação no grupo 1 e escore positivo classifica a observação no grupo 2.

5.2 A classificação

```
#modelo com as classificações
modeloclass=MASS::lda(g~.,data=dt, CV=T)
modeloclass$class
```

```
##  [1] Compraria    Compraria    Compraria    Compraria    NãoCompraria
##  [6] NãoCompraria Compraria    NãoCompraria NãoCompraria NãoCompraria
## Levels: Compraria NãoCompraria
```

```
modeloclass$posterior
```

```
##     Compraria NãoCompraria
## 1  0.99882375 0.0011762539
## 2  0.77123305 0.2287669519
## 3  0.99999990 0.0000001005
## 4  0.95195714 0.0480428631
## 5  0.28255383 0.7174461700
## 6  0.02509826 0.9749017405
## 7  0.99011943 0.0098805736
## 8  0.02750984 0.9724901606
## 9  0.00128666 0.9987133443
## 10 0.00003801 0.9999619921
```

Com o argumento CV=T na função lda podemos ver a classificação gerada pelo modelo, assim como a probabilidade de pertencimento em cada grupo. Probabilidade menor do que 0.5 classifica não pertencente ao grupo e probabilidade maior do que 0.5 como pertencente.

6 Usando a função predict

Sobre os dados que geraram o modelo:

```
predict(modelocoef,dt[,1:3])
```

```
## $class
##  [1] Compraria    Compraria    Compraria    Compraria    Compraria
##  [6] NãoCompraria NãoCompraria NãoCompraria NãoCompraria NãoCompraria
## Levels: Compraria NãoCompraria
##
## $posterior
##    Compraria NãoCompraria
## 1  0.9992588   0.00074120
## 2  0.8777527   0.12224725
## 3  0.9999604   0.00003964
## 4  0.9920904   0.00790958
## 5  0.9150311   0.08496888
## 6  0.0039292   0.99607076
## 7  0.3263665   0.67363354
## 8  0.0143245   0.98567552
## 9  0.0006999   0.99930009
## 10 0.0001558   0.99984419
##
## $x
##        LD1
## 1  -2.2125
## 2  -0.6052
## 3  -3.1119
## 4  -1.4834
## 5  -0.7297
## 6   1.6995
## 7   0.2225
## 8   1.2991
## 9   2.2302
## 10  2.6916
```

6.1 Avaliando a acurácia

```
table(dt$g, predict(modelocoef,dt[,1:3])$class)
```

```
##
##                Compraria NãoCompraria
##   Compraria            5            0
##   NãoCompraria         0            5
```

Sobre novos dados

```
x1=c(8,5,5,6,6)
x2=c(8,4,9,8,6)
x3=c(3,7,9,3,6)
g=c("Compraria","NãoCompraria","Compraria","Compraria","Compraria")
novodt=tibble(x1,x2,x3,g)
novodt
```

```
## # A tibble: 5 x 4
##      x1    x2    x3 g
##   <dbl> <dbl> <dbl> <chr>
## 1     8     8     3 Compraria
## 2     5     4     7 NãoCompraria
## 3     5     9     9 Compraria
## 4     6     8     3 Compraria
## 5     6     6     6 Compraria
```

```
predict(modelocoef,novodt[,1:3])
```

```
## $class
## [1] Compraria    NãoCompraria NãoCompraria Compraria    NãoCompraria
## Levels: Compraria NãoCompraria
##
## $posterior
##   Compraria NãoCompraria
## 1  0.999860     0.000140
## 2  0.003929     0.996071
## 3  0.215100     0.784900
## 4  0.994185     0.005815
## 5  0.442417     0.557583
##
## $x
##        LD1
```

```
## 1 -2.72444
## 2  1.69947
## 3  0.39742
## 4 -1.57854
## 5  0.07103
```

```
table(novodt$g, predict(modelocoef,novodt[,1:3])$class)
```

```
##
##                Compraria NãoCompraria
##   Compraria            2            2
##   NãoCompraria         0            1
```

Desse modo teremos a seguinte classificação para os novos clientes:

```
novodt %>% mutate(previsao=predict(modelocoef,novodt)$class)
```

```
## # A tibble: 5 x 5
##      x1    x2    x3 g            previsao
##   <dbl> <dbl> <dbl> <chr>        <fct>
## 1     8     8     3 Compraria    Compraria
## 2     5     4     7 NãoCompraria NãoCompraria
## 3     5     9     9 Compraria    NãoCompraria
## 4     6     8     3 Compraria    Compraria
## 5     6     6     6 Compraria    NãoCompraria
```

7 Exercício

Pratique agora usando o banco de dados iris!

7.0.1 1-Aplique a função lda ao banco de dados iris. Avalie a acurácia do modelo.

7.0.2 2- Gráfico das funções discriminantes

Execute o código abaixo para gerar um gráfico. Interprete o mesmo.

```
modeloiris=MASS::lda(Species~., iris)
modeloiris
```

```
## Call:
## lda(Species ~ ., data = iris)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##     0.3333     0.3333     0.3333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## 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
## Sepal.Length  0.8294  0.0241
## Sepal.Width   1.5345  2.1645
## Petal.Length -2.2012 -0.9319
## Petal.Width  -2.8105  2.8392
##
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

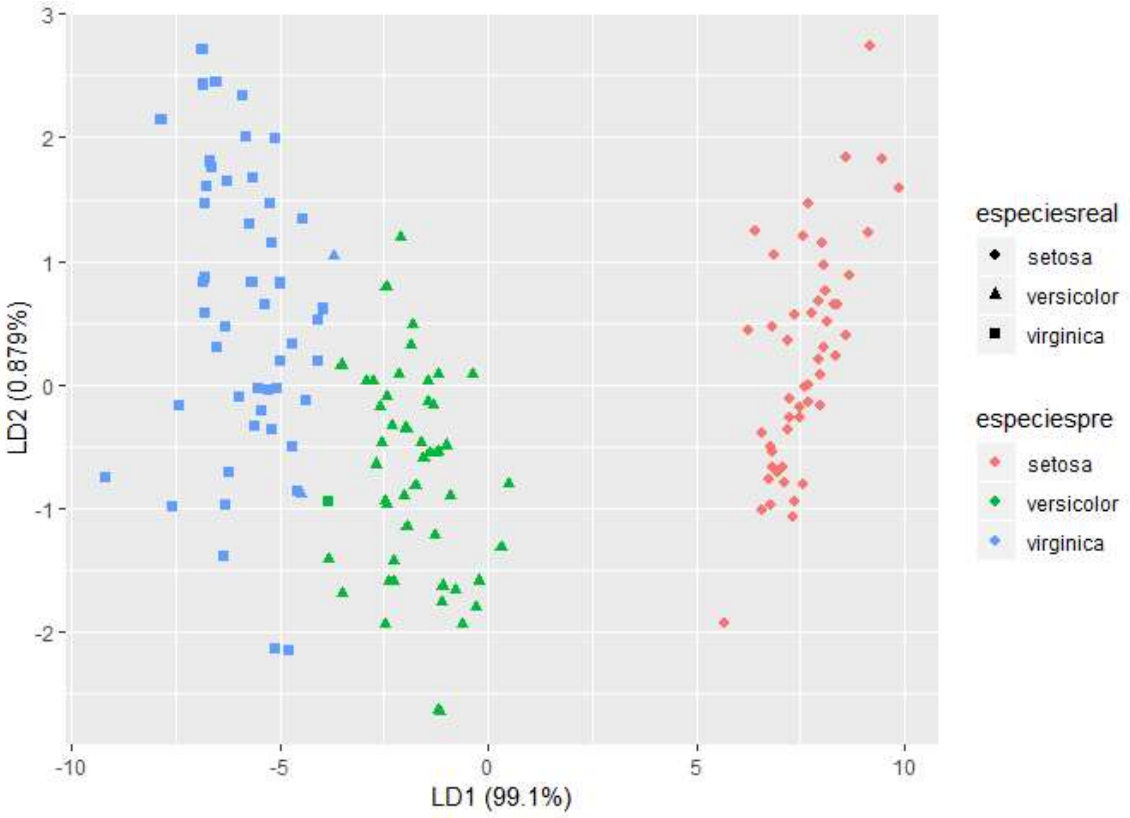
```
previsaomodeloiris=predict(modeloiris,iris[1:4])

dados = data.frame(especiespre = previsaomodeloiris$class,especiesreal=iris[,5], lda = previsa

prop=modeloiris$svd^2/sum(modeloiris$svd^2)#r quadrado

p <- ggplot(dados) +
  geom_point(aes(lda.LD1, lda.LD2, colour = especiespre, shape = especiesreal), size = 1.8) +
  labs(x = paste("LD1 (", scales::percent(prop[1]), ")", sep=""),
       y = paste("LD2 (", scales::percent(prop[2]), ")", sep=""))

p
```



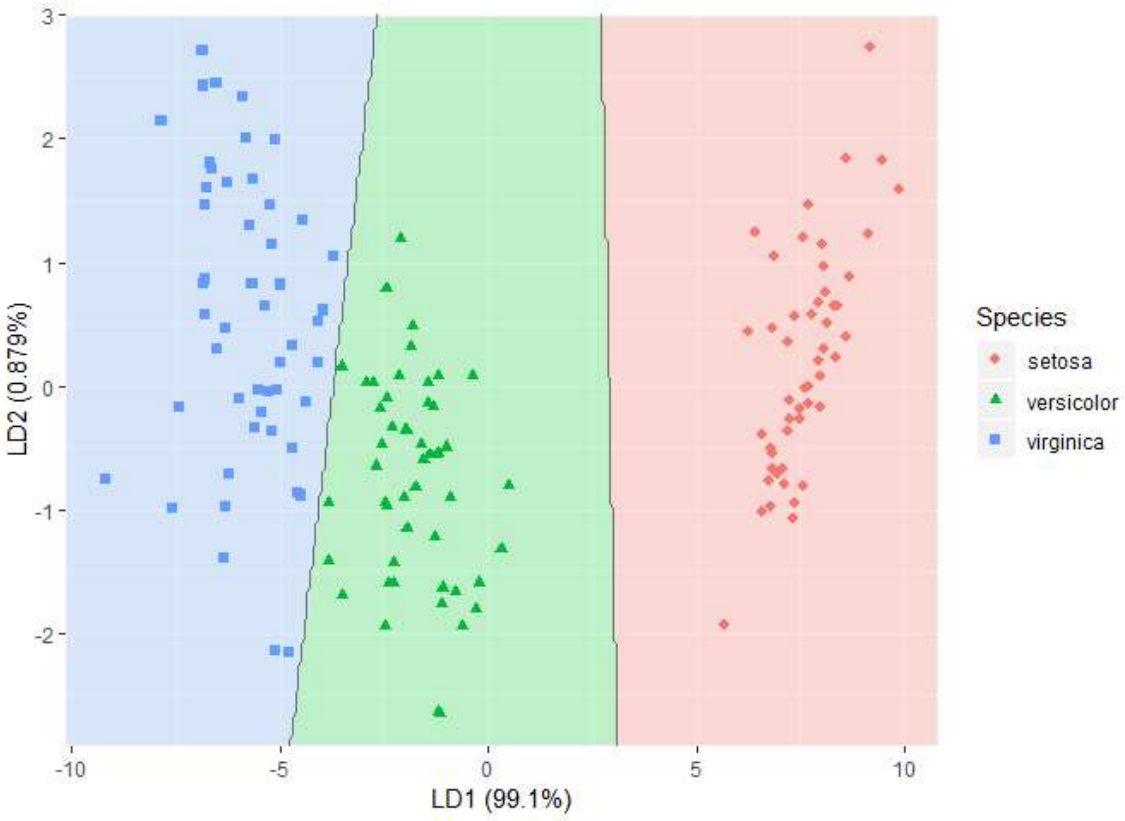
Outro gráfico que mostra a área de classificação sobre uma malha.

```
library(scales)

datPred <- data.frame(Species=predict(modeloiris)$class,predict(modeloiris)$x)
#Criando os limites de classificação
fit2 <- MASS::lda(Species ~ LD1 + LD2, data=datPred, prior = rep(1, 3)/3)
ld1lim <- expand_range(c(min(datPred$LD1),max(datPred$LD1)),mul=0.05) #soma e subtrai 5% da di
ld2lim <- expand_range(c(min(datPred$LD2),max(datPred$LD2)),mul=0.05)
ld1 <- seq(ld1lim[[1]], ld1lim[[2]], length.out=300)
ld2 <- seq(ld2lim[[1]], ld1lim[[2]], length.out=300)
newdat <- expand.grid(list(LD1=ld1,LD2=ld2)) #produz uma malha cruzando todos os valores
preds <-predict(fit2,newdata=newdat) #classifica a malha
predclass <- preds$class
postprob <- preds$posterior #fornece a posteriori da malha
#organizando o data frame da malha
df <- data.frame(x=newdat$LD1, y=newdat$LD2, class=predclass)
df$classnum <- as.numeric(df$class)
df <- cbind(df,postprob)

colorfun <- function(n,l=65,c=100) { hues = seq(15, 375, length=n+1); hcl(h=hues, l=l, c=c)[1:
#colors <- colorfun(3)
colorslight <- colorfun(3,l=90,c=50)
ggplot(datPred, aes(x=LD1, y=LD2) ) +
    geom_raster(data=df, aes(x=x, y=y, fill = factor(class)),alpha=0.7,show_guide=FALSE) +
    geom_contour(data=df, aes(x=x, y=y, z=classnum), colour="black", alpha=0.5, breaks=c(1.5,2
    geom_point(data = datPred, size = 1.8, aes(pch = Species,  colour=Species)) +
    scale_x_continuous(limits = ld1lim, expand=c(0,0)) +
    scale_y_continuous(limits = ld2lim, expand=c(0,0)) +
```

```
    scale_fill_manual(values=colorslight,guide=F)+
  labs(x = paste("LD1 (", scales::percent(prop[1]), ")", sep=""),
       y = paste("LD2 (", scales::percent(prop[2]), ")", sep=""))
```



7.0.3 3-O que fazer antes de iniciar a modelagem

Não esqueça de verificar os outliers, a normalidade multivariada, tratar os dados faltantes e realizar transformações.

Vamos aqui utilizar a função do pacote caret para realizar transformações boxcox.

```
require(caret)

trans = caret::preProcess(iris,
                  c("BoxCox", "center", "scale"))
trans
```

```
## Created from 150 samples and 5 variables
##
## Pre-processing:
##   - Box-Cox transformation (4)
##   - centered (4)
##   - ignored (1)
##   - scaled (4)
```

```
##
## Lambda estimates for Box-Cox transformation:
## -0.1, 0.3, 0.9, 0.6
```

```
iristrans = data.frame(
      trans = predict(trans, iris))
```

O lambda estimado para cada variável é estabelecido pela função preProcess

Modelando após a transformação boxcox

```
modeloiris=MASS::lda(trans.Species~., iristrans)
modeloiris
```

```
## Call:
## lda(trans.Species ~ ., data = iristrans)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##     0.3333     0.3333     0.3333
##
## Group means:
##            trans.Sepal.Length trans.Sepal.Width trans.Petal.Length
## setosa                -1.0425            0.8359            -1.3006
## versicolor             0.1553           -0.6643             0.2844
## virginica              0.8872           -0.1716             1.0163
##            trans.Petal.Width
## setosa               -1.3048
## versicolor            0.2917
## virginica             1.0131
##
## Coefficients of linear discriminants:
##                        LD1     LD2
## trans.Sepal.Length  0.4593 -0.9938
## trans.Sepal.Width   0.7580  1.4212
## trans.Petal.Length -3.1513  2.0833
## trans.Petal.Width  -2.9953 -0.5054
##
## Proportion of trace:
##    LD1    LD2
## 0.9936 0.0064
```

```
#Como o modelo classificou
classificacao <- iristrans %>%
  mutate(prev=predict(modeloiris,iristrans[1:4])$class) %>%
  dplyr::select(trans.Species,prev)

#Quais foram classificadas incorretamente

iristrans %>%
  mutate(prev=predict(modeloiris,iristrans[1:4])$class) %>%
  filter(trans.Species!=prev)
```

```
##   trans.Sepal.Length trans.Sepal.Width trans.Petal.Length
## 1             0.1385            0.3719             0.5903
## 2             0.2576           -0.8071             0.7602
## 3             0.6031           -0.5597             0.7602
##   trans.Petal.Width trans.Species       prev
## 1            0.8000    versicolor  virginica
## 2            0.5957    versicolor  virginica
## 3            0.4897     virginica versicolor
```

Observe que no caso dos dados iris o ganho em usar transformação boxcox foi praticamente nenhum. O objetivo aqui foi apenas ilustrar esse procedimento.

8 Análise discriminante usando o pacote caret

```
library(caret)

modeloiris.caret <- train(Species ~ .,
                          method ='lda',
                          data=iris)

modeloiris.caret
```

```
## Linear Discriminant Analysis
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 150, 150, 150, 150, 150, 150, ...
## Resampling results:
```

```
##
##   Accuracy  Kappa
##   0.9746    0.9615
```

```
modeloiris.caret$finalModel
```

```
## Call:
## lda(x, grouping = y)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##     0.3333     0.3333     0.3333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## 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
## Sepal.Length  0.8294  0.0241
## Sepal.Width   1.5345  2.1645
## Petal.Length -2.2012 -0.9319
## Petal.Width  -2.8105  2.8392
##
## Proportion of trace:
##    LD1    LD2
## 0.9912 0.0088
```

```
pred <-  predict(object = modeloiris.caret, newdata = iris[,-5])
```

Matriz de confusão

```
confusionMatrix(data = pred,
reference = iris$Species)
```

```
## Confusion Matrix and Statistics
##
```

```
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         1
##   virginica       0          2        49
##
## Overall Statistics
##
##                Accuracy : 0.98
##                  95% CI : (0.943, 0.996)
##     No Information Rate : 0.333
##     P-Value [Acc > NIR] : <0.0000000000000002
##
##                   Kappa : 0.97
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                  1.000             0.960            0.980
## Specificity                  1.000             0.990            0.980
## Pos Pred Value               1.000             0.980            0.961
## Neg Pred Value               1.000             0.980            0.990
## Prevalence                   0.333             0.333            0.333
## Detection Rate               0.333             0.320            0.327
## Detection Prevalence         0.333             0.327            0.340
## Balanced Accuracy            1.000             0.975            0.980
```

9 Analise Discriminante de Mahalanobis

Veja a referêcia do livro do Anderson Rodrigo da Silva: "Métodos de Análise Multivariada em R"

Aqui o classificador baseia-se na menor distância de mahalanobis em cada grupo.

```
require(biotools)
```

```
## ---
## biotools version 3.1
```

```
x=iris[,-5]
D2.disc(x,grouping=iris[,5])
```

```
##
## Call:
## D2.disc.default(data = x, grouping = iris[, 5])
##
## Mahalanobis distances from each class and class prediction (first 6 rows):
##   setosa versicolor virginica grouping   pred misclass
## 1 0.2911      98.88     191.8   setosa setosa
## 2 2.0313      80.97     169.2   setosa setosa
## 3 0.5533      87.29     177.1   setosa setosa
## 4 2.0867      75.29     160.7   setosa setosa
## 5 0.5956     100.92     193.9   setosa setosa
## 6 1.9448      95.94     183.1   setosa setosa
##
## Class means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## 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
##
## Confusion matrix:
##            new setosa new versicolor new virginica
## setosa             50              0             0
## versicolor          0             48             2
## virginica           0              1            49
```

10 Exercício

1-Faça a análise nos dados do titanic

10.1 Referências

Hair

Mingoti

código da função lda:

```
methods(lda)
getAnywhere('lda.default')
```