Função discriminante linear de Fisher em R

```
## Biblioteca com a função discriminante linear de Fisher
library(MASS)

## Dados (?iris apresenta informações sobre o conjunto de dados)
dados <- iris
names(dados)
```

```
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width" "Species"
```

A variável `Species` indica o grupo.

```
cat("\n Tamanho da amostra:", n <- length(dados$Species))
```

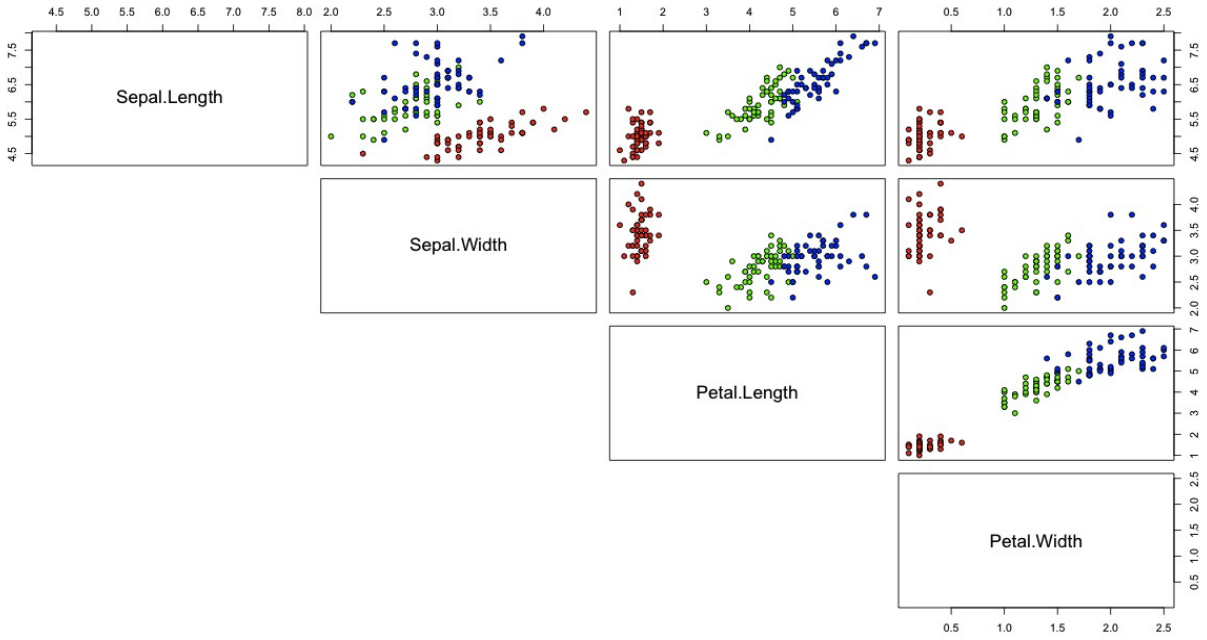
```
Tamanho da amostra: 150
```

```
summary(dados)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Para cada uma das três espécies foram coletadas 50 observações.

```
## Gráficos de dispersão
cores <- rainbow(length(levels(dados[, "Species"])))
pairs(dados[, -5], pch = 21, bg = cores[dados$Species], lower.panel = NULL)
```



```
## Análise discriminante
# Primeiro modelo (validação cruzada)
m1 <- lda(dados[, -5], dados$Species, CV = TRUE)

# Componentes de m1
names(m1)
[1] "class"     "posterior" "call"

# Segundo modelo (ressubstituição)
m2 <- lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
          data = dados)

# Componentes de m2
names(m2)

 [1] "prior"   "counts"  "means"   "scaling" "lev"
 [6] "svd"     "N"       "call"    "terms"   "xlevels"
```

Nota 1. Qual o resultado do comando `m2 <- lda(Species ~ ., data = dados)`?

```
# Matriz de confusão (validação cruzada)
tabela1 <- xtabs(~ m1$class + Species, data = dados)
cat("\n Matriz de confusão (com validação cruzada):")
tabela1

 Matriz de confusão (com validação cruzada):
 tabela1
            Species
m1$class     setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         1
  virginica       0          2        49

cat("\n Acerto (%) = \n", levels(dados[, "Species"]), "\n",
   diag(tabela1) / colSums(tabela1) * 100)

 Acerto (%) =
 setosa versicolor virginica
 100 96 98

cat("\n Acerto global (%) =", sum(diag(tabela1)) / n * 100)

 Acerto global (%) = 98

# Matriz de confusão (ressubstituição)
m2class <- predict(m2, dados)$class
tabela2 <- xtabs(~ m2class + Species, data = dados)
cat("\n Matriz de confusão (com ressubstituição):")
tabela2

cat("\n Acerto (%) = \n", levels(dados[, "Species"]), "\n",
    diag(tabela2) / colSums(tabela2) * 100)
 Acerto (%) =
 setosa versicolor virginica
 100 96 98
```

```
cat("\n Acerto global (%) =", sum(diag(tabela2)) / n * 100)
```

```
 Acerto global (%) = 98

 Matriz de confusão (com ressubstituição):

            Species
m2class      setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         1
  virginica       0          2        49
```

```
cat("\n Funções discriminantes: \n")
coef(m2)
```

```
                   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
```

Nota 2. `m1` foi obtido com o método de validação cruzada. O resultado da chamada `coef(m1)` é `NULL`. Era esperado? Neste caso, como classificar novas observações?

```
cat("\n Razão dos desvios padrão entre e intragrupos para cada FD =")
m2$svd
```

```
Razão dos desvios padrão entre e intragrupos para cada FD =
[1] 48.642644  4.579983
```

O resultado acima fornece uma ideia da importância das diferentes funções discriminantes.

```
# Escores das observações
FD <- as.matrix(dados[, -5]) %*% coef(m2)
dim(FD)
```

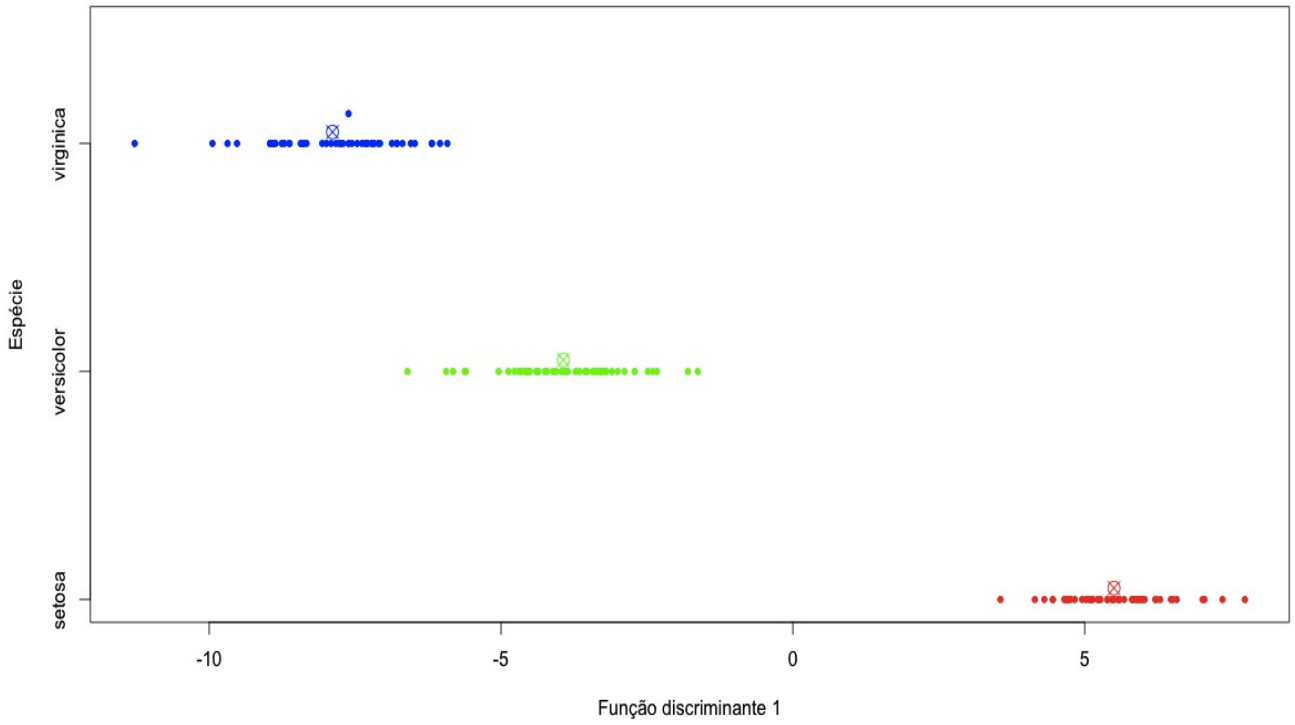
```
[1] 150   2
```

`FD` é uma matriz com 150 linhas (pois $n = 150$) e duas colunas. Cada coluna contém o escore de sua respectiva função discriminante calculado para cada observação (nas linhas de `FD`).

```
# Centróides dos grupos e escores dos centróides
m2$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
```

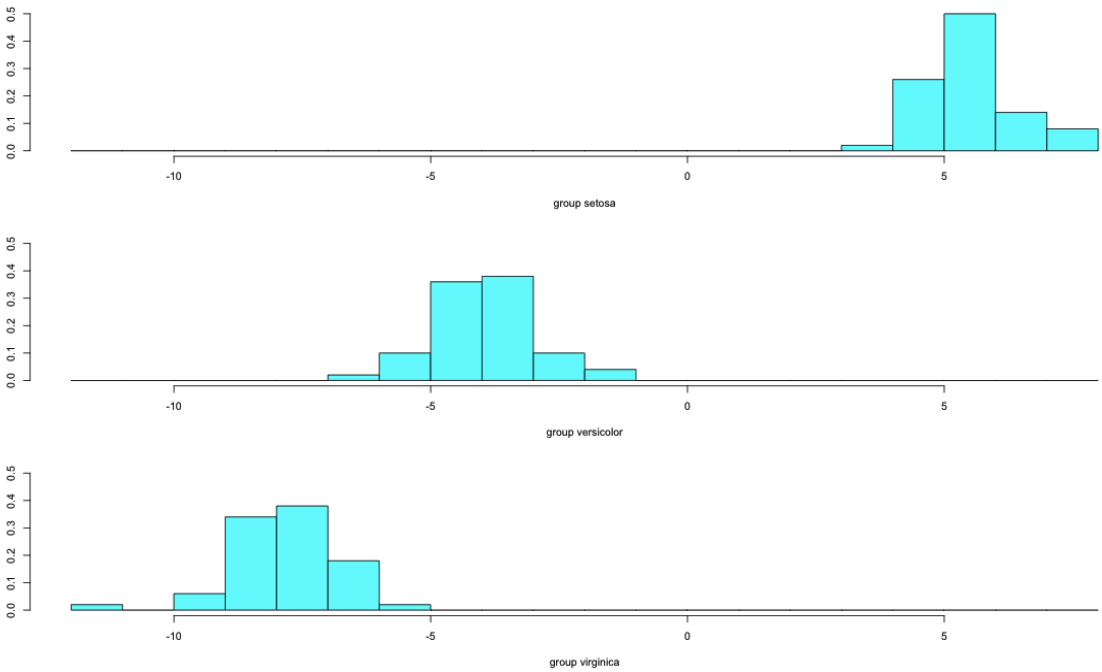
```
(FDb <- m2$means %*% coef(m2))

                 LD1      LD2
setosa      5.502493 6.876606
versicolor -3.930156 5.933573
virginica  -7.887657 7.174239

# Gráfic de pontos de FD1
stripchart(FD[, 1] ~ Species, pch = 20, xlab = "Função discriminante 1",
    ylab = "Espécie", col = cores, method = "stack", data = dados)
points(FDb[, 1], (1:length(m2$lev)) + 0.05, pch = 13, col = cores, cex = 1.5)
```



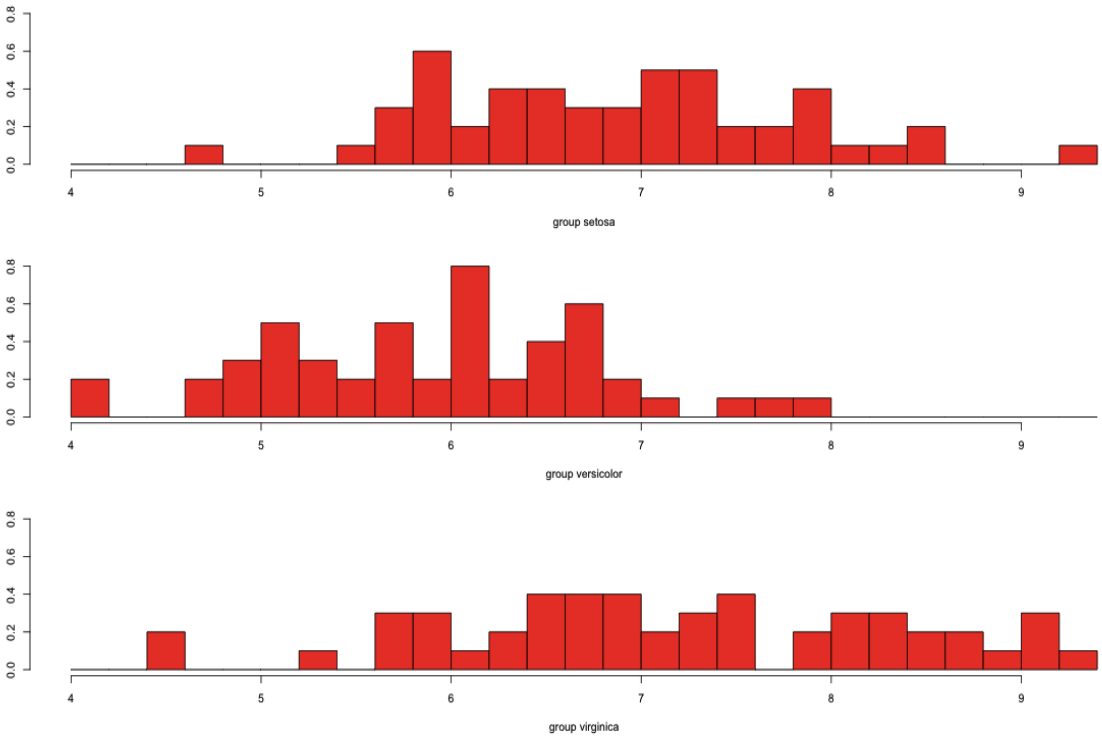
Nota 3. Apresente estimativas das taxas de acerto obtidas apenas com a primeira função discriminante.

Nota 4. Selecione uma amostra de 25 observações de cada espécie, obtenha as funções discriminantes e utilize-as para classificar as demais observações (função `predict`), apresentando as estimativas das taxas de acerto.

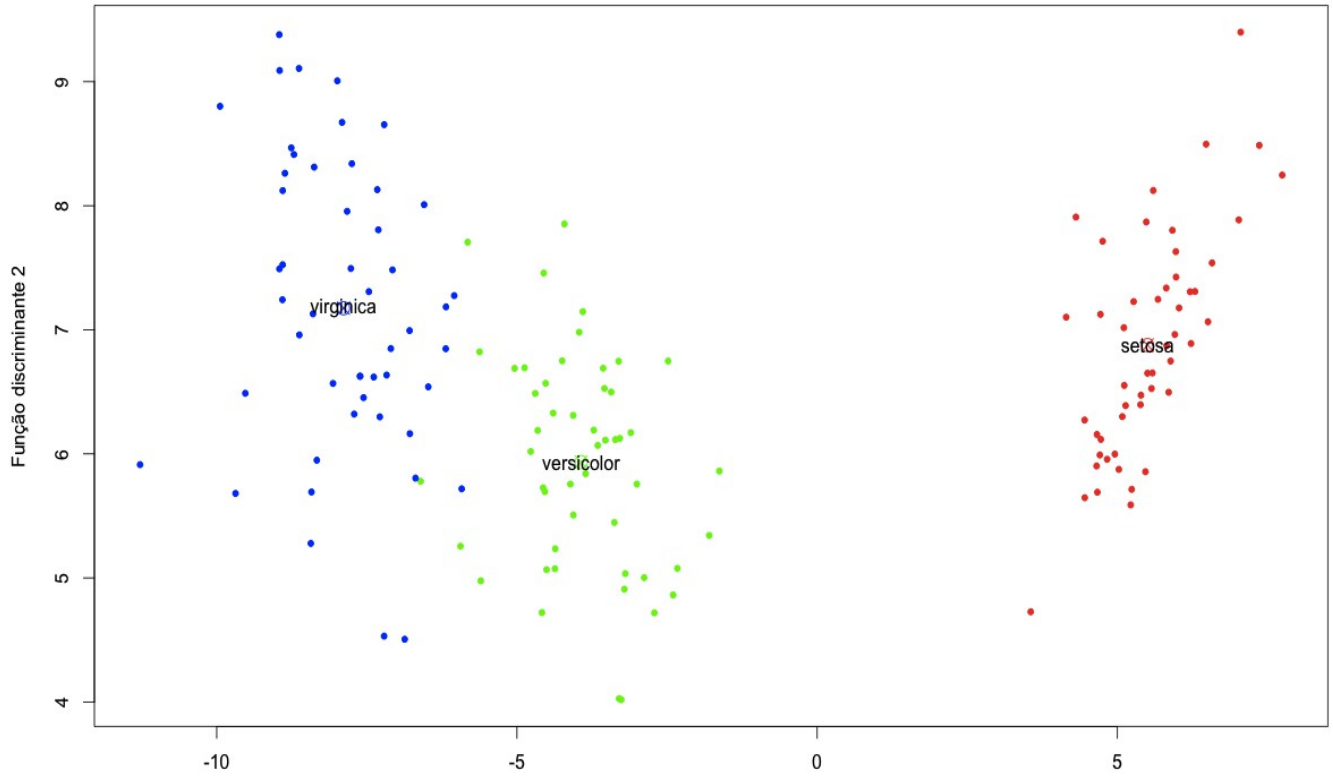
```
# Histograma de FD1
ldahist(FD[, 1], dados$Species)
```




```
# Histograma de FD2
ldahist(FD[, 2], dados$Species, col = "red")
```



```
# Gráfico de dispersão de FD1 e FD2
plot(FD[, 1], FD[, 2], pch = 20, col = cores[dados$Species],
     xlab = "Função discriminante 1", ylab = "Função discriminante 2")
points(FDb[, 1], FDb[, 2], pch = 13, col = cores, cex = 1.5)
text(FDb[, 1], FDb[, 2], m2$lev)
```



Nota 5. Refaça o exemplo aplicando funções discriminantes quadráticas (função `qda` em R).

Nota 6. Aplique um método de seleção de variáveis.

Nota 7. Procure obter todos os resultados deste exemplo utilizando outros pacotes estatísticos (SAS, SPSS, Minitab e Statistica, por exemplo).