Introdução ao R com Aplicações

Lupércio França Bessegato
Augusto Carvalho Souza
Dep. de Estatística/UFJF

Roteiro Geral

1. Fundamentos da linguagem R
2. Visualização e descrição de dados
3. Inferência estatística básica
4. Modelos de regressão
5. Análise de dados multivariados
6. Séries temporais
7. Referências

Análise de Dados Multivariados - Introdução

Objetivos

- Familiarização com os dados
- Detecção de estruturas interessantes
- Presença de valores atípicos (*outliers*)

Razões para Uso de AED

√ Identificação de erros e inconsistências
√ Verificação de pressupostos do modelo
√ Seleção preliminar de modelos apropriados
√ Determinação das relações entre as variáveis explicativas
√ Avaliação da direção e da intensidade das relações entre as variáveis explicativas e as variáveis respostas.

Análise Multivariada

- Para um conjunto de variáveis correlacionadas:
  √ Avaliar as relações entre as variáveis
  √ Considerar os efeitos dos "tratamentos" sobre essas relações
  √ Considerar como uma "resposta" depende dessas relações

- Métodos multivariados para redução de dados:
  √ Resumir as correlações entre variáveis
  √ Produzir um conjunto menor de variáveis (não correlacionadas) contendo as informações mais importantes
- Para um conjunto de objetos "relacionados"
  √ Identificar grupos de objetos semelhantes
  √ Identificar diferenças entre grupos de objetos semelhantes
    – (e o que faz com que os objetos sejam semelhantes)

Análise Exploratória de Dados Multivariados

- Sequência básica inicial:
  √ Medidas-resumo e gráficos:
    – Variabilidade para cada variável
    – Forma da distribuição de cada variável
  √ Grupos de observações:
    – Pré-determinados
    – (para encontrar diferenças potenciais)
  √ Diagrama de dispersão/correlações
    – Associações entre pares de variáveis

- Recomenda-se executar análise exploratória de dados univariados em cada um dos componentes, antes de realizar a AED multivariada.

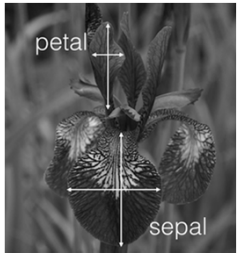
Importante

- A Análise Exploratória de Dados é um passo inicial crítico em qualquer análise de dados.

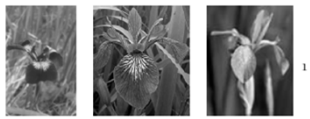
Conjunto de Dados

- Anderson (1935) e Fischer (1936)
- Conjunto de dados de flores de íris (gênero de iridácea)
  - √ Medidas morfológicas de 50 flores de cada espécie
  - √ Espécies:
    - Iris setosa (originária do Alasca)
    - Iris versicolor
    - Iris virginica
- Dados: *iris* {*datasets*}

- Morfologia iris:



- Espécies



√ Variáveis:

- Sepal.Length: comprimento da sépala, em cm.
- Sepal.Width: largura da sépala, em cm.
- Petal.Length: comprimento da pétala, em cm.
- Petal.Width: largura da pétala, em cm.
- Species: setosa, versicolor e virginica

• Carregamento do conjunto de dados

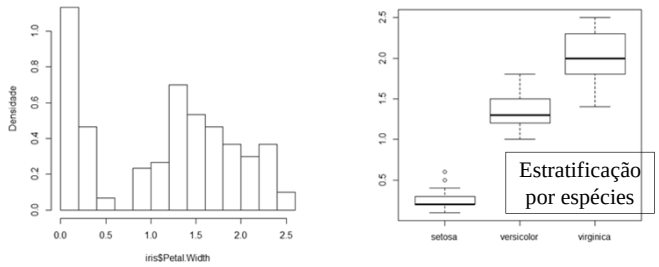
```
> dim(iris)
[1] 150  5
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 ...
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

• Estratos da categórica

```
> table(iris$Spcies)

    setosa versicolor  virginica
        50         50         50
```


• Histograma da variável `Petal.Width`:

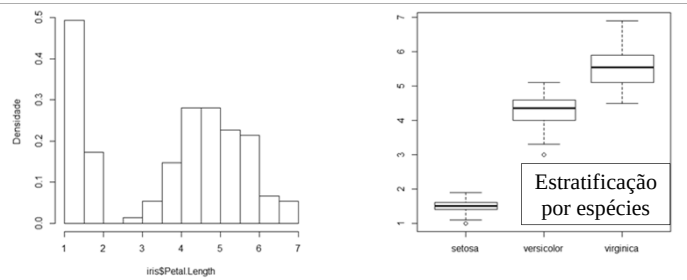
```
> # Petal.Width
> hist(iris$Petal.Width, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Petal.Width ~ Species, data = iris)
```



√ Mistura de populações

• Histograma da variável `Petal.Length`:

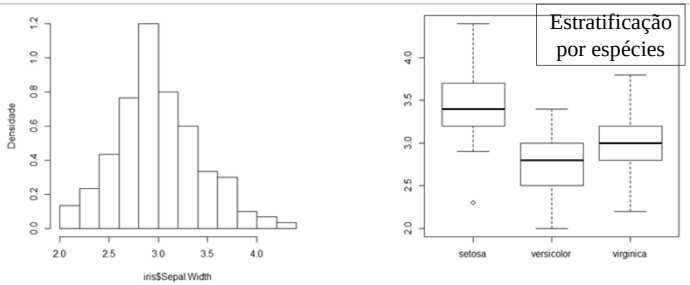
```
> # Petal.Length
> hist(iris$Petal.Length, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Petal.Length ~ Species, data = iris)
```



√ Mistura de populações

- Histograma da variável `Sepal.Width`:

```
> # Sepal.Width
> hist(iris$Sepal.Width, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Sepal.Width ~ Species, data = iris)
```



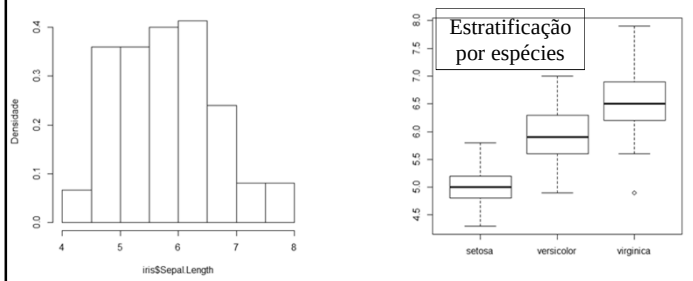
Estratificação por espécies

√ Mistura de populações menos acentuada

17

- Histograma da variável `Sepal.Length`:

```
> # Sepal.Length
> hist(iris$Sepal.Length, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Sepal.Length ~ Species, data = iris)
```



Estratificação por espécies

√ Mistura de populações mais próximas

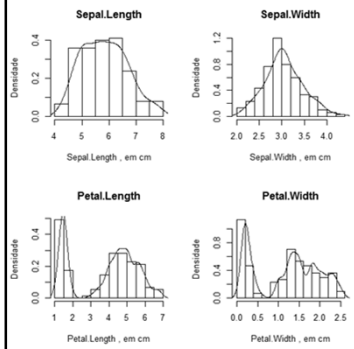
18

- Histogramas com suavizador:
  - √ Comando `density`: núcleo estimador

```
> variaveis <- names(iris[-5])
> par(mfrow = c(2, 2))
> for(i in 1: length(variaveis)) {
+ with(iris, {
+ dados <- eval(parse(text = variaveis[i]))
+ hist(dados, freq = F, main = variaveis[i], ylab = "Densidade",
+ xlab = paste(variaveis[i],", em cm"))
+ d <- density(dados, bw = "sj")
+ lines(d, lty = 1, col = "blue")
+ })
+ }
> par(mfrow = c(1, 1))
```

19

- Histogramas com suavizador:



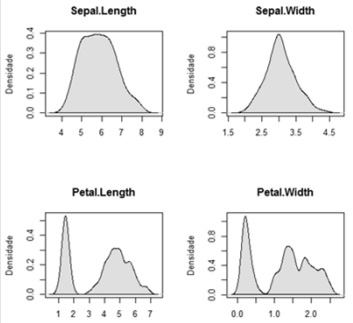
√ Facilita visualização das misturas

20

- Estimativas das densidades:

```
> variaveis <- names(iris[-5])
> par(mfrow = c(2, 2))
> for(i in 1: length(variaveis)) {
+ with(iris, {
+ dados <- eval(parse(text = variaveis[i]))
+ d <- density(dados, bw = "sj")
+ plot(d, type = "n", main = variaveis[i], ylab = "Densidade",
+ xlab = "")
+ polygon(d, col = "wheat")
+ })
+ }
> par(mfrow = c(1, 1))
```


- Estimativas suavizadas das densidades:



√ Todas as variáveis com estratificação por `Species`

- *Violin plot*:
  - √ Visualização da distribuição dos dados e de sua densidade.

```
> library(vioplot)
> nomes <- levels(iris$Species)
> with(iris,{
+ #for(i in 1:3) assign(paste0("x",i), Petal.Width[Species == nomes[i]])
+ for(i in 1:3) assign(paste("x",i, sep=""), Petal.Width[Species == nomes[i]])
+ vioplot(x1, x2, x3, names = nomes,  col = "lightblue", horizontal = TRUE) # col =
"gold"
+ title("Violin Plots de Petal.Width")
+ })
```

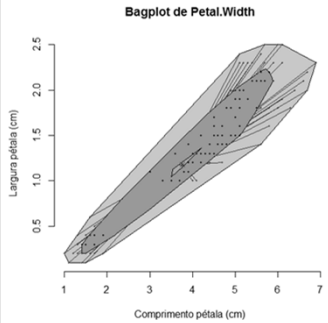

- *Violin plot* de `Petal.Width`:

- √ Semelhante box plot
- √ Apresenta densidade condicional
- √ Cuidado com o uso em variáveis discretas

- *Bag plot*:
  - √ Versão bivariada do box-plot.

```
> library(aplpack)
> #Fonte: http://www.statmethods.net/graphs/boxplot.html
> # Bagplot de Largura de pétala
> bagplot(iris$Petal.Length, iris$Petal.Width,
+ xlab = "Comprimento pétala (cm)", ylab = "Largura pétala (cm)",
+   main = "Bagplot de Petal.Width")
```

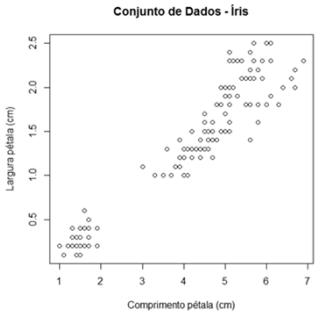

- *Bag plot* de `Petal.Width`:



- √ *Bag* contém 50% dos dados.
- √ Aponta outliers

- Diagrama de Dispersão - Pétalas

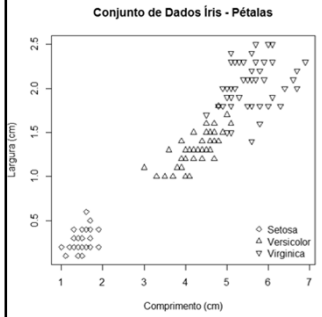
```
> # scatter plot simples
> plot(iris$Petal.Length, iris$Petal.Width, main="Conjunto de Dados - Íris",
+ xlab = "Comprimento pétala (cm)", ylab = "Largura pétala (cm)")
```



- √ Tendência linear
- √ Há dois agrupamentos de dados

- *Scatter plot* – Pétalas por `Species`:

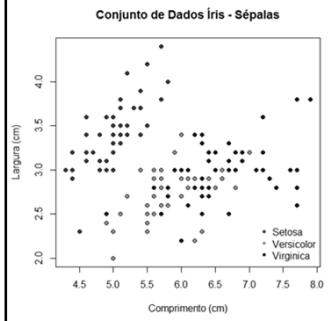
```
> # Scatter plot com o fator 'Species' - Pétalas
> plot(iris$Petal.Length, iris$Petal.Width, pch =
c(23,24,25)[unclass(iris$Species)],
+ main = "Conjunto de Dados Íris - Pétalas", xlab = "Comprimento (cm)",
+ ylab = "Largura (cm)")
> legend("bottomright", legend = c("Setosa", "Versicolor", "Virginica"),
+   pch = c(23,24,25), bty = "n")
```



- √ Tendência linear
- √ Percebe-se a discriminação dos 3 grupos

• *Scatter plot* – Sépalas por `Species`:

```
> # Scatter plot com fator 'Species' - Sépalas
> plot(iris$Sepal.Length, iris$Sepal.Width, pch = 21,
+ bg = c("red","green3","blue")[unclass(iris$Species)],
+ main = "Conjunto de Dados Íris - Sépalas", xlab = "Comprimento (cm)",
+ ylab = "Largura (cm)")
> legend("bottomright", legend = c("Setosa", "Versicolor", "Virginica"),
+   pch = rep(20,3), col = c("red","green3","blue"), cex = 1, bty = "n")
```

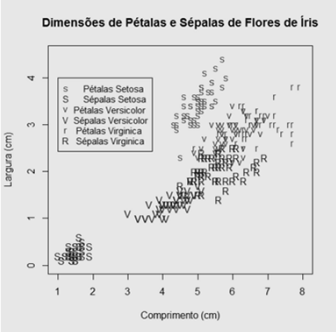


√ Grupo das setosas está bem discriminado
√ Discriminação entre os dois outros grupos não está tão clara

• *Scatter plot* com todas medidas e com o fator `Species`:

```
> # Scatter plot com o fator 'Species' - Todos comprimentos e larguras
> iS <- iris$Species == "setosa"
> iV <- iris$Species == "versicolor"
> iG <- iris$Species == "virginica"
> op <- par(bg = "bisque")
> matplot(c(1, 8), c(0, 4.5), type =  "n",
+   xlab = "Comprimento (cm)", ylab = "Largura (cm)",
+         main = "Dimensões de Pétalas e Sépalas de Flores de Íris")
> matpoints(iris[iS,c(1,3)], iris[iS,c(2,4)], pch = "sS", col = c(2,4))
> matpoints(iris[iV,c(1,3)], iris[iV,c(2,4)], pch = "vV", col = c(2,4))
> matpoints(iris[iG,c(1,3)], iris[iG,c(2,4)], pch = "rR", col = c(2,4))
> legend(1, 4, c("    Pétalas Setosa", "    Sépalas Setosa",
+               "Pétalas Versicolor", "Sépalas Versicolor",
+    " Pétalas Virginica", " Sépalas Virginica"), cex=0.9,
+        pch = "sSvVrR", col = rep(c(2,4), 3))
```


• *Scatter plot* – todos os comprimentos e larguras por `Species`:



√ Setosa é do Alasca
√ Há *clusters*?
√ Há *outliers*?

• *Plot* bivariado – pacote `xda`:

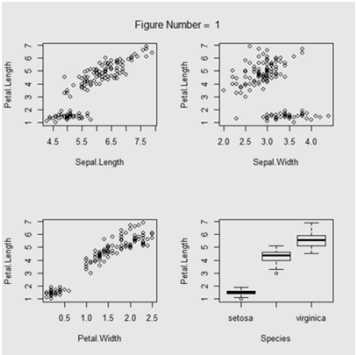
```
> library(devtools)
> install_github("ujjwalkarn/xda")
> library(xda)
> # resumo de todas as variáveis quantitativas
> numSummary(iris)
              n mean    sd max min range nunique nzeros  iqr lowerbound
Sepal.Length 150 5.84 0.828 7.9 4.3   3.6      35      0 1.30       3.15
Sepal.Width  150 3.06 0.436 4.4 2.0   2.4      23      0 0.50       2.05
Petal.Length 150 3.76 1.765 6.9 1.0   5.9      43      0 3.55      -3.72
Petal.Width  150 1.20 0.762 2.5 0.1   2.4      22      0 1.50      -1.95
             upperbound noutlier kurtosis skewness mode miss miss%   1%   5%
Sepal.Length       8.35        0   -0.606    0.309  5.0    0     0 4.40 4.60
Sepal.Width        4.05        4    0.139    0.313  3.0    0     0 2.20 2.34
Petal.Length      10.42        0   -1.417   -0.269  1.4    0     0 1.15 1.30
Petal.Width        4.05        0   -1.358   -0.101  0.2    0     0 0.10 0.20
             25%  50% 75%  95%  99%
Sepal.Length 5.1 5.80 6.4 7.25 7.70
Sepal.Width  2.8 3.00 3.3 3.80 4.15
Petal.Length 1.6 4.35 5.1 6.10 6.70
Petal.Width  0.3 1.30 1.8 2.30 2.50
> # resumo de todas as variáveis qualitativas
> charSummary(iris)
         n miss miss% unique                      top5levels:count
Species 150    0     0      3 setosa:50, versicolor:50, virginica:50
```

- *Plot* Tabela de dupla entrada entre `Sepal.Length` e `Species`:

```
> # análise bivariada entre 'Species' e 'Sepal.Length'
> bivariate(iris,'Species','Sepal.Length')
  bin_Sepal.Length setosa versicolor virginica
1        (4.3,5.2]     39          5         1
2        (5.2,6.1]     11         29        10
3          (6.1,7]      0         16        27
4          (7,7.9]      0          0        12
```


- *Plot* de todas as variáveis vs. `Petal.Length`:

```
> # plot de todas as variáveis contra Petal.Length
> Plot(iris,'Petal.Length')
```

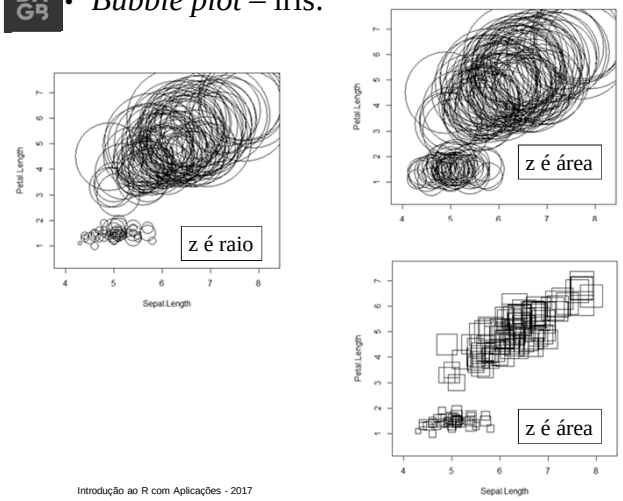


√ Gráficos bivariados com `Petal.Length` aparentam discriminar estratos de `Species`

- *Bubble plot*:
  - √ Extensão do diagrama de dispersão:
  - √ Usa dimensão adicional dos dados para determinar tamanho dos símbolos

```
> # variável z é raio
> with(iris, symbols(Sepal.Length, Petal.Length, circles = Petal.Width))
> # variável z é área
> raio <- sqrt(iris$Petal.Width/pi)
> with(iris, symbols(Sepal.Length, Petal.Length, circles = raio))
> # x = S.L; y = P.L, z = P.W
> with(iris, symbols(Sepal.Length, Petal.Length, circles = raio, inches=0.35,
+ fg = "white", bg = "darkgray", xlab = "Comprimento de sépala",
+ ylab = "Comprimento de pétala"))
> # quadrado com área Petal.Width
> with(iris, symbols(Sepal.Length, Petal.Length, squares = sqrt(Petal.Width),
+   inches=0.5))
```

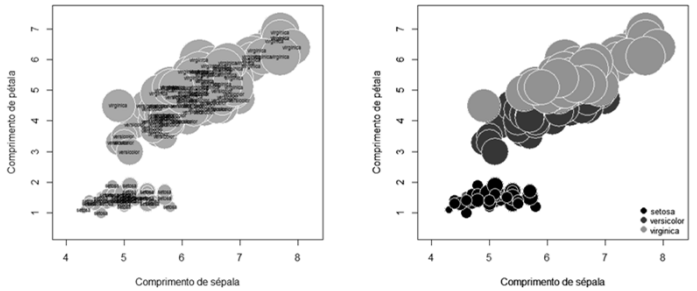

- *Bubble plot* – iris:



- *Bubble plot* – iris:
  - √ x = `Sepal.Length`.
  - √ y = `Petal.Length`.
  - √ z = `Petal.Width`.

```
> # x = S.L; y = P.L, z = P.W e fator
> with(iris, symbols(Sepal.Length, Petal.Length, circles = raio, inches=0.35,
+ fg = "white", bg = "darkgray", xlab = "Comprimento de sépala",
+ ylab = "Comprimento de pétala"))
> text(iris$Sepal.Length, iris$Petal.Length, iris$Species, cex=0.5)
> # x = S.L; y = P.L, z = P.W e fator em 3 cores
> with(iris, {
+ symbols(Sepal.Length, Petal.Length, circles = raio, inches=0.35,
+   fg = "white", bg = unclass(Species),
+   xlab = "Comprimento de sépala", ylab = "Comprimento de pétala")
+ legend("bottomright", levels(iris$Species), pch = rep(20, 3),
+   pt.cex = 2, bg = unique(unclass(iris$Species)),
+   col = unique(unclass(iris$Species)), bty = "n", cex = 0.8)
+ })
```

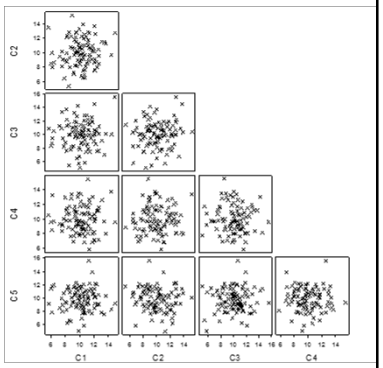

- *Bubble plot* – iris:



√ Espécie setosa têm pétalas mais estreitas

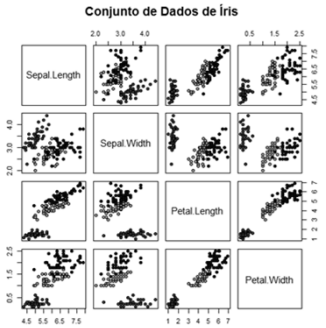
Scatter Plot Matrix

- Dados quantitativos:
  - √ Cuidado se houver muitos empates
- Diagrama de dispersão para os pares de variáveis
  - √ Apresentação em forma matricial
- Calcular coeficiente de correlação de cada par de variáveis



- *Scatter plot matrix* – iris:

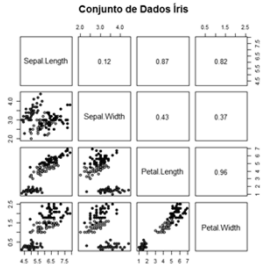
```
> pairs(iris[1:4], main = "Conjunto de Dados de Íris", pch = 21,
+ bg = c("red", "green3", "blue")[unclass(iris$Species)])
```



√ Quais gráficos aparentam discriminar melhor os grupos?
√ Há relações entre as medidas morfológicas?

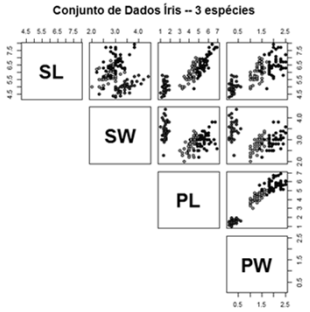
- *Scatter plot matrix* com correlações:

```
> # função para personalização do painel
> painel.pearson <- function(x, y, ...) {
+ horizontal <- (par("usr")[1] + par("usr")[2]) / 2;
+ vertical <- (par("usr")[3] + par("usr")[4]) / 2;
+ text(horizontal, vertical, format(abs(cor(x,y)), digits=2), cex = 1.2,
+ font = 1)
+ }
> # Scatter plot matrix com correlações
> pairs(iris[1:4], main = "Conjunto de Dados Íris", pch = 21,
+ bg = c("red","green3","blue")[unclass(iris$Species)],
+ upper.panel = painel.pearson)
```



- *Scatter plot matrix* com diagonal modificada:

```
> # Scatterplot matrix com diagonal modificada
> pairs(iris[1:4], main = "Conjunto de Dados Íris -- 3 espécies", pch = 21,
+ bg = c("red", "green3", "blue")[unclass(iris$Species)],
+ lower.panel = NULL, labels = c("SL", "SW", "PL", "PW"), font.labels = 2,
+ cex.labels = 3.0
```

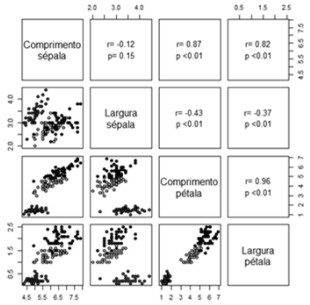


- *Scatter plot matrix* com correlação e p-valor:
  - √ Função para personalização do painel

```
> # Scatterplot matrix com correlação e p-valor
>
> # função para personalização do painel
> painel.cor <- function(x, y, digits = 2, cex.cor, ...){
+   usr <- par("usr"); on.exit(par(usr))
+   par(usr = c(0, 1, 0, 1))
+   # coeficiente de correlação
+  r <- cor(x, y)
+   txt <- format(c(r, 0.123456789), digits = digits)[1]
+   txt <- paste("r= ", txt, sep = "")
+   text(0.5, 0.6, txt, cex = 1.2)
+   # cálculo do p-valor
+   p <- cor.test(x, y)$p.value
+   txt2 <- format(c(p, 0.123456789), digits = digits)[1]
+  txt2 <- paste("p= ", txt2, sep = "")
+   if(p < 0.01) txt2 <- paste("p ", "<0.01", sep = "")
+   text(0.5, 0.4, txt2, cex = 1.2)
+ }
```


- √ *Scatter plot matrix* com correlação e p-valor:

```
> # scatter plot matrix
> pairs(iris[,1:4], pch = 21,
+ bg = c("red", "green3", "blue")[unclass(iris$Species)],
+ upper.panel = painel.cor,
+ labels = c("Comprimento\nsépala", "Largura\nsépala",
+ "Comprimento\npétala", "Largura\npétala"))
```



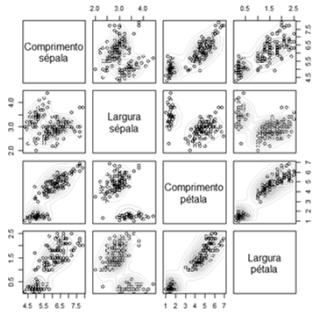
• *Scatter plot matrix* com estimativas de densidade bivariada:

√ Função para estimativas de densidade bivariada

```
> library(MASS)
> library(colorspace)
>
> # função para estimativas densidade bivariada por núcleo
> painel.dens <- function(x,y) {
+     points(x,y)
+     k <- kde2d(x,y)# package: MASS
+     cnt <- contourLines(k$x, k$y, k$z)
+     n <- length(cnt)
+     cols <- rev(sequential_hcl(n))# package: colorspace
+     for( i in seq_len(n) ) lines(cnt[[i]], col=cols[i])
+ }
```


√ *Scatter plot matrix* com densidade bivariada:

```
> # Scatter plot matrix
> pairs(iris[,1:4], panel = painel.dens,
+ labels = c("Comprimento\nsépala", "Largura\nsépala",
+ "Comprimento\npétala", "Largura\npétala"))
```

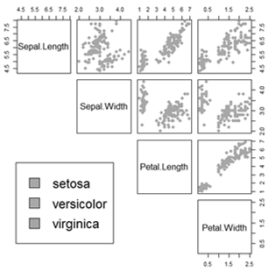


√ Estimativa densidade está codificada por cor

√ Pode ser conveniente quando houver muitos empates

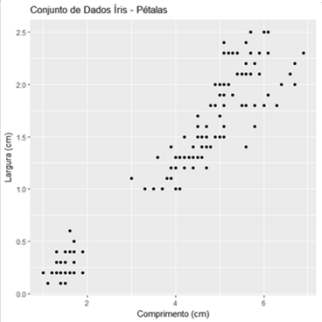
• *Scatter plot matrix* com legenda:

```
> library(colorspace) # cores melhores
> species_labels <- iris[,5]
> species.cor <- rev(rainbow_hcl(3))[as.numeric(iris$Species)]
> # Plot um SPloM:
> pairs(iris[-5], col = species.cor, lower.panel = NULL,
+ cex.labels = 1.7, pch = 19, cex = 1.2)
> par(xpd = TRUE)
> legend(x = 0.05, y = 0.4, cex = 1.5, legend = as.character(levels(iris$Species)),
+     fill = unique(species.cor))
> par(xpd = NA)
```



• *Scatter plot matrix* com pacote `ggplot2`:

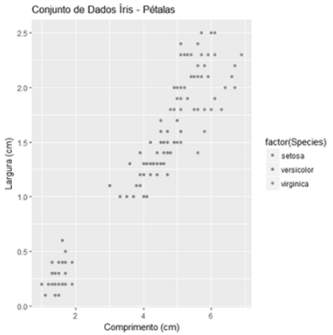
```
> library(ggplot2)
> library(gridExtra)
> # Plot com pontos default
> sp1 <- ggplot(iris, aes(x = Petal.Length, y = Petal.Width))
> sp1 + geom_point() +
+ xlab("Comprimento (cm)") +   ylab("Largura (cm)") +
+ ggtitle("Conjunto de Dados Íris - Pétalas")
```



√ Configuração default

• *Scatter plot matrix* com pacote `ggplot2`:

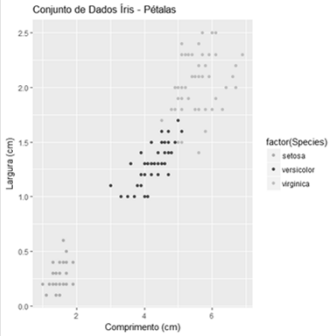
```
> # Mudança de cor dos pontos
> sp2 <- sp1 + geom_point(aes(color = factor(Species))) + # cor p/ nível fator
+  xlab("Comprimento (cm)") +
+    ylab("Largura (cm)") +
+  ggtitle("Conjunto de Dados Íris – Pétalas")
> sp2
```



√ Pontos com códigos de cores

• *Scatter plot matrix* com pacote `ggplot2`:

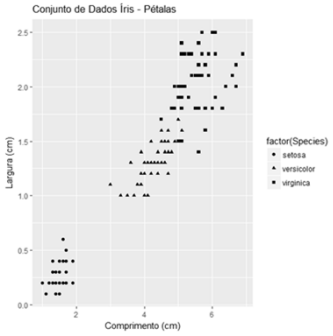
```
> # Cores conforme usuário
> sp3 <- sp1 + geom_point(aes(color=factor(Species))) +
+ scale_color_manual(values = c("orange", "purple", "gray")) +
+ xlab("Comprimento (cm)") +
+   ylab("Largura (cm)") +
+ ggtitle("Conjunto de Dados Íris – Pétalas")
> sp3
```



√ Códigos de cores conforme usuário

• *Scatter plot matrix* com pacote `ggplot2`:

```
> # Mudança forma e tamanho dos pontos
> sp4 <- sp1 + geom_point(aes(shape = factor(Species))) + # forma p/ nível fator
+  xlab("Comprimento (cm)") +
+    ylab("Largura (cm)") +
+  ggtitle("Conjunto de Dados Íris – Pétalas")
> sp4
```

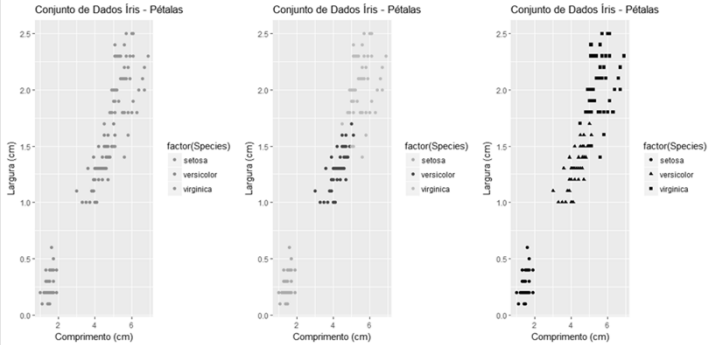


√ Modificação da forma e do tamanho dos pontos

• *Scatter plot matrix* com pacote `ggplot2`:

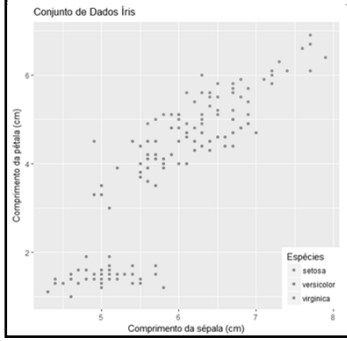
√ Painel com gráficos

```
> # painel com os gráficos
> grid.arrange(sp2, sp3, sp4, nrow=1)
```



• *Scatter plot matrix* com pacote `ggplot2`:

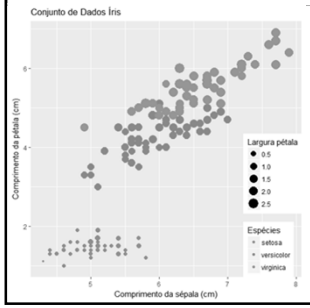
```
> # Scatterplot comprimentos + espécies
> ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
+ geom_point()+
+ xlab("Comprimento da sépala (cm)") +
+   ylab("Comprimento da pétala (cm)") +
+ ggtitle("Conjunto de Dados Íris") +
+ scale_color_discrete(name ="Espécies") +
+ theme(legend.position = c(1, 0), legend.justification = c(1,0))
```



√ Comprimentos de pétala e de sépala oferecem boa discriminação das espécies

• *Scatter plot matrix* com pacote `ggplot2`:

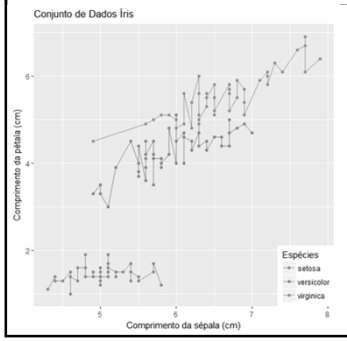
```
> # Scatterplot comprimentos vs. espécies vs.largura pétala (bubble)
> ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species,
+   size = Petal.Width), alpha = I(0.7)) +# alpha: reduz overplotting
+ geom_point() +
+ xlab("Comprimento da sépala (cm)") +
+   ylab("Comprimento da pétala (cm)") +
+ ggtitle("Conjunto de Dados Íris") +
+ scale_color_discrete(name ="Espécies") +
+ scale_size_continuous(name = "Largura pétala") +
+ theme(legend.position = c(1, 0), legend.justification = c(1,0))
```



√ Flores da espécie setosa têm as pétalas mais estreitas

• *Scatter plot* em linhas:

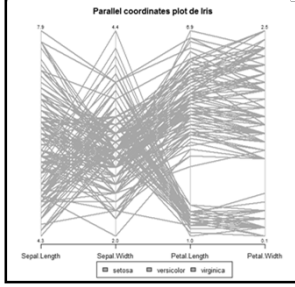
```
>   # Scatter plot em linhas
> ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
+ geom_line()+ geom_point() +
+ xlab("Comprimento da sépala (cm)") +
+   ylab("Comprimento da pétala (cm)") +
+ ggtitle("Conjunto de Dados Íris") +
+ scale_color_discrete(name ="Espécies") +
+ theme(legend.position = c(1, 0), legend.justification = c(1,0))
```



√ Gráfico não faz muito sentido, mas pode ajudar a "enxergar" grupos

• *Parallel coordinate plot*:

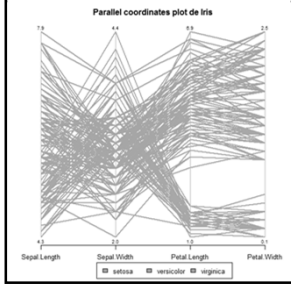
```
> library(MASS)
> library(colorspace) # get nice colors
> species.cor <- rev(rainbow_hcl(3))[as.numeric(iris$Species)]
> par(las = 1, mar = c(4.5, 3, 3, 2) + 0.1, cex = .8)
> parcoord(iris[-5], col = species.cor, var.label = TRUE, lwd = 2)
> title("Parallel coordinates plot de Iris")
> par(xpd = TRUE)
> legend(x = 1.75, y = -0.125, cex = 1,
+    legend = as.character(levels(iris$Species)),
+    fill = unique(species.cor), horiz = TRUE)
> par(xpd = NA)
```



√ Flores da espécie setosa têm as pétalas mais estreitas

• *Parallel coordinate plot*:

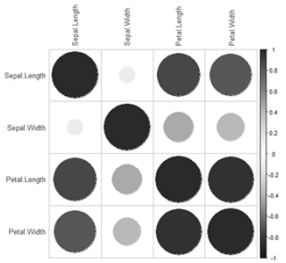
```
> library(MASS)
> library(colorspace) # get nice colors
> species.cor <- rev(rainbow_hcl(3))[as.numeric(iris$Species)]
> par(las = 1, mar = c(4.5, 3, 3, 2) + 0.1, cex = .8)
> parcoord(iris[-5], col = species.cor, var.label = TRUE, lwd = 2)
> title("Parallel coordinates plot de Iris")
> par(xpd = TRUE)
> legend(x = 1.75, y = -0.125, cex = 1,
+    legend = as.character(levels(iris$Species)),
+    fill = unique(species.cor), horiz = TRUE)
> par(xpd = NA)
```



√ Flores da espécie setosa têm as pétalas mais estreitas

• Correlograma:

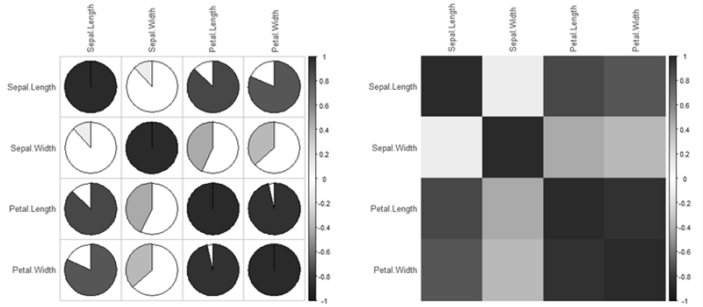
```
        # Matriz de correlações
(iris.cor <- cor(iris[-5]))
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
> library(corrplot)
> # correlograma - círculo
> corrplot(iris.cor, method = "circle")
```



√ círculos

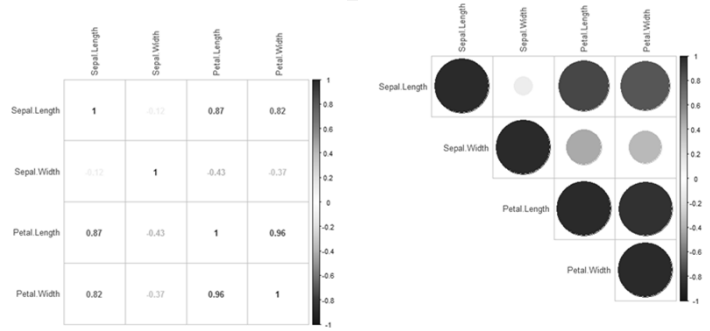
• Correlograma:

```
> # correlograma - pizza
> corrplot(iris.cor, method = "pie")
> # coorelograma - cor
> corrplot(iris.cor, method = "color")
```



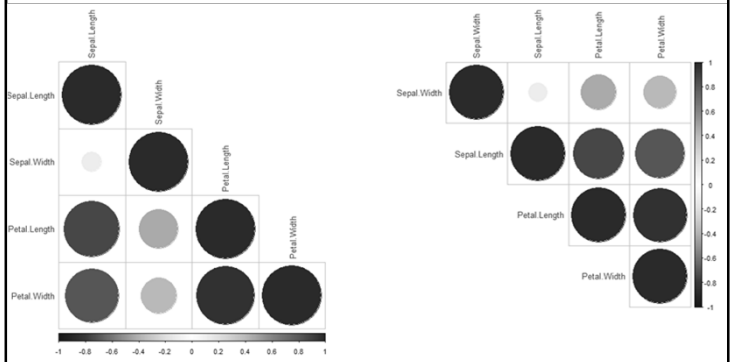
• Correlograma:

```
> # correlograma - valores
> corrplot(iris.cor, method = "number")
> # correlograma - superior
> corrplot(iris.cor, type = "upper")
```



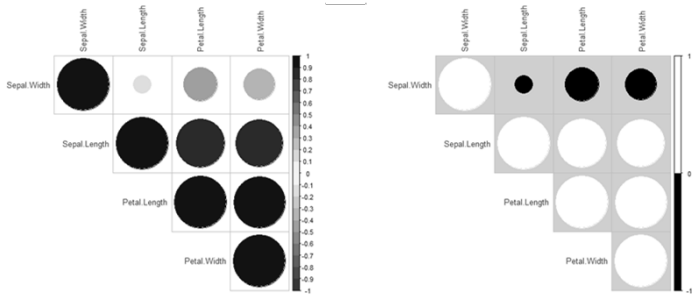
Correlograma:

```
> # correlograma - inferior
> corrplot(iris.cor, type = "lower")
> # correlograma c/ reordenação por hclust
> corrplot(iris.cor, type="upper", order = "hclust")
```



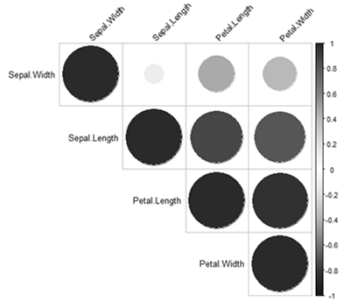
Correlograma:

```
> # usando espectro de cores diferente
> col <- colorRampPalette(c("red", "white", "blue"))(20)
> corrplot(iris.cor, type = "upper", order = "hclust", col = col)
> # Mudando cor de fundo para lightblue
> corrplot(iris.cor, type = "upper", order = "hclust", col = c("black", "white"),
+          bg = "lightblue")
```



Correlograma:

```
> # Mudando a cor e a rotação dos rótulos
> corrplot(iris.cor, type = "upper", order = "hclust", tl.col = "black",
+ tl.srt = 45)
> #tl.col (cor do texto) e tl.srt (rotação texto)
```



Correlograma:

√ Função para cálculo de p-valor

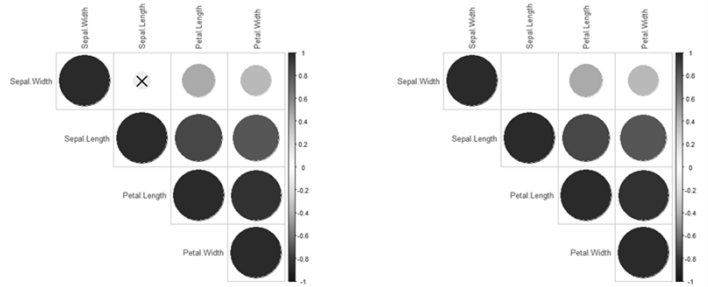
```
> # Função para cálculo do p-valor das correlações
> cor.mteste <- function(mat, ...) {
+     mat <- as.matrix(mat)
+     n <- ncol(mat)
+     p.mat<- matrix(NA, n, n)
+     diag(p.mat) <- 0
+     for (i in 1:(n - 1)) {
+         for (j in (i + 1):n) {
+             tmp <- cor.test(mat[, i], mat[, j], ...)
+             p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
+         }
+     }
+   colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
+   p.mat
+ }
```

√ Matriz dos p-valores das correlações

```
> # matriz dos p-valores das correlações
> p.mat <- cor.mteste(iris[-5])
> head(p.mat)
 Sepal.Length  Sepal.Width Petal.Length  Petal.Width
Sepal.Length 0.000000e+00 1.518983e-01 1.038667e-47 2.325498e-37
Sepal.Width  1.518983e-01 0.000000e+00 4.513314e-08 4.073229e-06
Petal.Length 1.038667e-47 4.513314e-08 0.000000e+00 4.675004e-86
Petal.Width  2.325498e-37 4.073229e-06 4.675004e-86 0.000000e+00
```

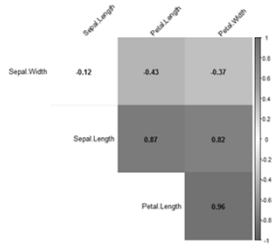
• Correlograma:

```
> # Agregando nível de significância ao correlograma
> corrplot(iris.cor, type="upper", order="hclust", p.mat = p.mat,
+ sig.level = 0.01)
> # Deixando em branco coeficiente não significante
> corrplot(iris.cor, type = "upper", order = "hclust", p.mat = p.mat,
+ sig.level = 0.01, insig = "blank")
```



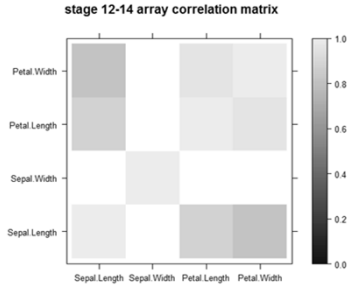
• Correlograma:

```
> # Customizando o correlograma
> col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD",
+ "#4477AA"))
> corrplot(iris.cor, method="color", col=col(200), type="upper", order="hclust",
+          addCoef.col = "black", # Adiciona coeficiente de correlação
+          tl.col="black", tl.srt=45, # Rotação e cor de texto rótulo
+          # Combinação com significância
+          p.mat = p.mat, sig.level = 0.01, insig = "blank",
+          diag=FALSE # elimina valores da diagonal principal
+          )
```



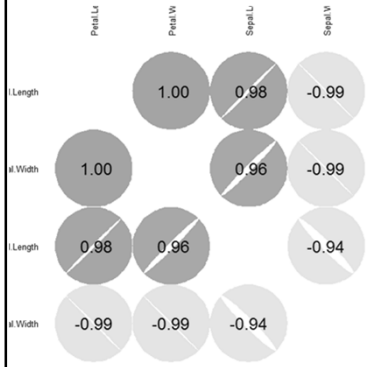
• Matriz de Correlações – pacote `lattice`:

```
> library(lattice)
> rgb.palette <- colorRampPalette(c("blue", "yellow"), space = "rgb")
> levelplot(iris.cor, main = "stage 12-14 array correlation matrix",
+ xlab = "", ylab = "", col.regions = rgb.palette(120),
+ cuts = 100, at = seq(0, 1, 0.01)
+ )
```



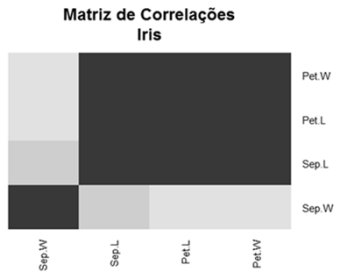
• Matriz de Correlações – pacote `lattice`:

```
> source("https://github.com/JVAdams/jvamisc/blob/master/R/plotcor.R")
> library(plotrix)
> library(seriation)
> library(MASS)
> plotcor(cor(iris.cor), mar = c(0.1, 4, 4, 0.1))
```



- Mapa de Calor:

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(iris.cor, col = brewer.pal(9, "GnBu"), trace ="none",
+     key = FALSE, dend = "none", cexCol = 1.1, cexRow = 1.1, srtCol = 90,
+     labRow = c("Sep.L", "Sep.W", "Pet.L", "Pet.W"),
+     labCol = c("Sep.L", "Sep.W", "Pet.L", "Pet.W"),
+     main = "\n\n\nMatriz de Correlações\nIris")
```



- Gráfico em html:
  - √ Gráfico 1:

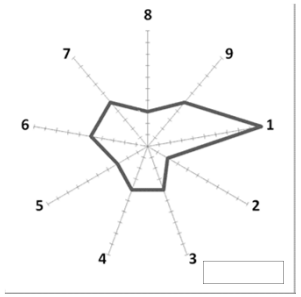
```
> library(plotly)
> p1 <- plot_ly(data = iris, x =~Sepal.Length, y =~Sepal.Width, split =~Species,
+   showlegend = F)
> p2 <- plot_ly(data = iris, x =~Sepal.Length, y =~Sepal.Width, split =~Species,
+   showlegend = T)
> subplot(p1,p2)
```

  - √ Gráfico 2:

```
> p1 <-
+   iris %>%
+   group_by(Species) %>%
+   plot_ly(x =~Sepal.Length, color = ~Species) %>%
+   add_markers(y = ~Sepal.Width)
> p2 <-
+   iris %>%
+   group_by(Species) %>%
+   plot_ly(x=~Sepal.Length, color= ~Species) %>%
+   add_markers(y =~Sepal.Width, showlegend = F)
> subplot(p1,p2)
```


Star Plot

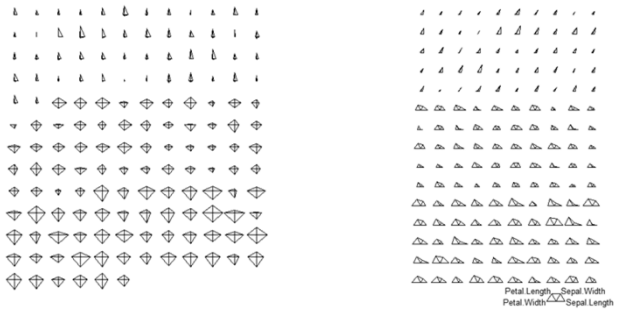
- Estrelas para visualização de dados
- Formação da estrela:
  - √ Raio para cada variável
  - √ Comprimento é proporcional à variável
- Útil para visualização de itens com número arbitrário de variáveis



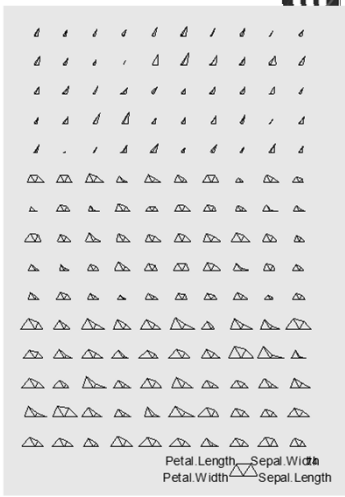
- Pode ser usado para responder as seguintes perguntas:
  - √ Quais variáveis são dominantes para uma determinada observação?
  - √ Quais observações são similares?
    (Existem agrupamentos de observações?)
  - √ Existem valores discrepantes?

- *Star plot*:

```
> #default
> stars(iris[, -5])
> # posicionamento legenda
> stars(iris[, -5], key.loc = c(17,0), full = F, ncol = 10)
```



- Exemplo: Flores de íris
  - √ Você vê diamantes?
    - Alguns são grandes, alguns são pequenos
  - √ Dados em sequência
    - Setosa, versicolar e virginica
  - √ Valores iniciais pequenos
    - Setosa é do Alasca!
  - √ Há outliers?



Componentes Principais

Introdução

- Objetivo:
  - √ Explicar a estrutura de variância e covariância de conjunto de variáveis através de algumas combinações lineares das mesmas
  - √ Busca-se:
    - Redução de dados
    - Interpretação

Componentes Principais Exatas

- Algebricamente:
  - √ Combinações lineares particulares das *p* variáveis aleatórias X_1, X_2, ..., X_p.
- Geometricamente:
  - √ Representam a seleção de um novo sistema de coordenadas obtidas por rotação do sistema original
  - √ Os novos eixos representam as direções com maior variabilidade
  - √ Fornecem descrição mais simples e mais parcimoniosa da estrutura de covariâncias

77

- Componentes principais:
  - √ São necessárias *p* componentes para reproduzir a variabilidade total do sistema
  - √ As componentes são não correlacionadas entre si
    - – Ortogonalidade entre as componentes
  - √ Variabilidade das p variáveis é aproximada pela variabilidade das k principais componentes
    - – Buscam-se situações em que haja quase tanta informação nas k componentes principais quanto nas p variáveis originais

78

- Análise de componentes principais:
  - √ Não pressupõe normalidade
    - – Componentes principais derivadas de populações normais têm interpretações úteis
  - √ Com frequência, revela relações insuspeitadas
    - – Pode permitir interpretações que não seriam obtidas preliminarmente
  - √ Em geral, é um passo intermediário para a aplicação de outras técnicas

79

Componentes Principais Exatas Extraídas da Matriz de Covariâncias

- Sejam o vetor aleatório

$$\mathbf{X}' = [X_1, X_2, \ldots, X_p].$$

com matriz de covariâncias é $\mathbf{\Sigma}$, cujos autovalores são $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$.

- Componentes principais de $\mathbf{\Sigma}$:

$$Y_1, Y_2, ..., Y_p.$$

  - √ Combinações lineares não correlacionadas do vetor aleatório, cujas variâncias são as maiores possíveis

80

- Definição – Componente principal:
  - √ A j-ésima componente principal da matriz **Σ** é definida como:

$$Y_j = \mathbf{e}_j'\mathbf{X} = e_{j1}X_1 + e_{j2}X_2 + \cdots + e_{jp}X_p.$$

  - √ $\mathbf{e}_j$: autovetor correspondente ao j-ésimo autovalor
- Esperança e variância de Y_j:

$$\mathrm{E}[Y_j] = \mathrm{E}[\mathbf{e}_j'\mathbf{X}] = \mathbf{e}_j'\boldsymbol{\mu} = e_{j1}\mu_1 + e_{j2}\mu_2 + \cdots + e_{jp}\mu_p.$$

$$\mathrm{Var}[Y_j] = \mathrm{Var}[\mathbf{e}_j'\mathbf{X}] = \mathbf{e}_j'\boldsymbol{\Sigma}\mathbf{e}_j = \mathbf{e}_j'\left(\sum_{i=1}^{p}\mathbf{e}_i\mathbf{e}_i'\right)\mathbf{e}_j' = \lambda_j.$$

- Covariância entre duas componentes principais:

$$\mathrm{Cov}[Y_j, Y_k] = 0,\ j \neq k$$

- Comentário:
  - √ Cada autovalor λ_j representa a variância de uma componente principal Y_j.
  - √ Autovalores estão ordenados em ordem decrescente
    - A primeira componente é a de maior variabilidade
    - A p-ésima componente é a de menor variabilidade

- Variâncias total e generalizada de **Σ**:
  - √ Total: $\mathrm{tr}(\boldsymbol{\Sigma}) = \sum_{i=1}^{p}\sigma_{ii} = \sum_{i=1}^{p}\lambda_i$
  - √ Generalizada de **Σ**: $|\boldsymbol{\Sigma}| = \prod_{i=1}^{p}\lambda_i$
  - √ Em termos dessas duas medidas globais de variação, os vetores **X** e **Y** são equivalentes

- Proporção da variância total que é explicada pela j-ésima componente principal:

$$\frac{\mathrm{Var}[Y_j]}{\text{Variância total de } \mathbf{X}} = \frac{\lambda_j}{\mathrm{tr}(\boldsymbol{\Sigma})} = \frac{\lambda_j}{\sum_{i=1}^{p}\lambda_i}$$

  - √ 1ª componente tem a maior proporção de explicação
- Proporção da variância total que é explicada pelas k primeiras componentes principais

$$\frac{\sum_{j=1}^{k}\mathrm{Var}[Y_j]}{\text{Variância total de } \mathbf{X}} = \frac{\sum_{j=1}^{k}\lambda_j}{\mathrm{tr}(\boldsymbol{\Sigma})} = \frac{\sum_{j=1}^{k}\lambda_j}{\sum_{i=1}^{p}\lambda_i}$$

  - √ Busca-se analisar um conjunto menor de variáveis sem perder muita informação sobre a estrutura de variabilidade original

- Aproximação de **Σ**:
  - √ Analisando as k primeiras componentes principais

$$\mathbf{\Sigma}_{p\times p} \approx \sum_{i=1}^{k} \lambda_i \mathbf{e}_j \mathbf{e}_j'$$

  - √ Cada parcela da soma envolve uma matriz de dimensão pxp correspondente apenas à informação da j-ésima componente principal

Correlação entre Componente Principal e Variável Aleatória

- Os coeficientes de correlação entre a componente principal Y_i de S e a variáve1 X_k é

$$\rho_{Y_i,X_k} = \frac{e_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$$

  - √ A magnitude de e_{ik} mede a contribuição da k-ésima variável na i-ésima componente (a despeito das outras variáveis).
    - Não medem a importância de X_k na presença das outras variáveis.
    - Alguns estatísticos recomendam que somente os valores e_{ik} (e não as correlações) sejam consideradas na interpretação dos componentes

Estimação das Componentes Principais – Matriz de Covariâncias

- Em geral, **Σ** é estimada por **S**:

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1p} \\ S_{12} & S_{22} & \dots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ S_{1p} & S_{2p} & \dots & S_{pp} \end{bmatrix}.$$

  - √ Autovalores de **S**: $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_p$
  - √ Autovetores de **S**: $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \dots, \hat{\mathbf{e}}_p$

- j-ésima componente principal de **S**:

$$\hat{Y}_j = \hat{\mathbf{e}}_j'\mathbf{X} = \hat{e}_{j1}X_1 + \hat{e}_{j2}X_2 + \cdots + \hat{e}_{jp}X_p,\ j = 1, 2, \dots, p.$$

- Componentes principais amostrais – Propriedades
  i. Variância: $\mathrm{Var}[\hat{Y}_j] = \hat{\lambda}_j.$
  ii. Covariância entre as componentes: $\mathrm{Cov}(\hat{Y}_j, \hat{Y}_k) = 0,\ j \neq k$
  iii. Variância total estimada explicada pela componente:

$$\frac{\mathrm{Var}[\hat{Y}_j]}{\text{Variância total estimada de } \mathbf{X}} = \frac{\hat{\lambda}_j}{\mathrm{tr}(\mathbf{S})} = \frac{\hat{\lambda}_j}{\sum_{i=1}^{p}\hat{\lambda}_i}$$

  iv. Correlação estimada entre componente e variável:

$$r_{\hat{Y}_j,X_k} = \frac{\hat{e}_{jk}\sqrt{\hat{\lambda}_j}}{\sqrt{S_{kk}}}$$

- Decomposição espectral de **S**:

$$\mathbf{S} = \sum_{j=1}^{p} \hat{\lambda}_j \mathbf{e}_j \mathbf{e}_j'.$$

  √ Aproximação de S pelas primeiras k componentes

$$\mathbf{S}_{p\times p} \approx \sum_{i=1}^{k} \hat{\lambda}_i \hat{\mathbf{e}}_j \hat{\mathbf{e}}_j'$$

- Scores das componentes
  √ Valor das componentes para cada elemento amostral
  √ Na prática, o uso das componentes relevantes de dá através dos scores

- Componente principal das variáveis padronizadas:
  √ A j-ésima componente principal da matriz **Σ** :

$$Y_j = \mathbf{e}_j'\mathbf{Z} = \mathbf{e}_j' \left(\mathbf{V}^{1/2}\right)^{-1} (\mathbf{X} - \boldsymbol{\mu}) = e_{j1}Z_1 + e_{j2}Z_2 + \cdots + e_{jp}Z_p.$$

  √ $\mathbf{e}_j$: autovetor da matriz de correlações **P**.
- Variância total de **P**:

$$\sum_{j=1}^{p} \mathrm{Var}[Y_j] = \sum_{j=1}^{p} \mathrm{Var}[Z_j] = p$$

  √ Proporção de variância populacional (padronizada) devido à j-ésima componente

$$\frac{\mathrm{Var}[Y_j]}{\text{Variância total de } \mathbf{Z}} = \frac{\lambda_j}{\mathrm{tr}(\mathbf{P})} = \frac{\lambda_j}{p},\ k = 1, 2, \ldots, p$$

  √ Correlação entre Y_j e X_k: $\rho_{Y_j, X_k} = e_{jk}\sqrt{\lambda_j},\ i, k = 1, 2, \ldots, p$

Componentes Principais de Variáveis Padronizadas

- Padronização do vetor aleatório **X**:

$$\mathbf{Z} = \left(\mathbf{V}^{1/2}\right)^{-1} (\mathbf{X} - \boldsymbol{\mu})$$

  √ $\mathbf{V}^{1/2}$: matriz diagonal de desvios-padrão
  √ Variável padronizada: $Z_i = \frac{X_i - \mu_i}{\sqrt{\sigma_{ii}}}$
  √ Matriz de covariâncias de **Z**:

$$\mathrm{Cov}(\mathbf{Z}) = \left(\mathbf{V}^{1/2}\right)^{-1} \boldsymbol{\Sigma} \left(\mathbf{V}^{1/2}\right)^{-1} = \mathbf{P}$$

  √ Componentes principais de **Z**:
    – Obtidas dos autovalores e autovetores de **P**.

Comentários

- As componentes principais de **Σ** são diferentes daquelas obtidas de **P**.
  √ Seus autovalores e autovetores são diferentes
  √ Um conjunto de componentes principais não é simplesmente uma função do outro conjunto
- A padronização traz consequências
  √ Variáveis deveriam ser padronizadas se elas são medidas em escalas com amplitudes muito diferentes
    – Ex. Vendas anuais e razão entre lucro/ativos

Padronização dos Componentes Principais Amostrais

- Frequentemente são padronizadas:
  - √ Variáveis medidas em diferentes escalas
  - √ Na mesma escala, mas com amplitudes bastante diferentes
- As componentes principais não são invariantes às mudanças na escala

Padronização dos Componentes Principais Amostrais

- Frequentemente são padronizadas:
  - √ Variáveis medidas em diferentes escalas
  - √ Na mesma escala, mas com amplitudes bastante diferentes
- As componentes principais não são invariantes às mudanças na escala

Análise de Componentes Principais – Matriz de Correlações

- As componentes principais obtidas a partir da matriz de covariâncias são influenciadas pelas variáveis de maior variância
  - √ A padronização das variáveis ameniza esse problema
- Análise de componentes principais de variáveis padronizadas é equivalente a obter as componentes principais através da matriz de correlações

Importante

- √ Um valor pequeno incomum para o último autovalor da matriz de covariâncias (ou correlação) amostral pode indicar uma dependência linear não detectada no conjunto de dados
- √ Valores grande de autovalores (e correspondentes autovetores são importantes em uma análise
- √ Autovalores próximos de zero não devem ser ignorados
  - Autovetores associados podem apontar dependências lineares no conjunto de dados

  (problemas computacionais ou de interpretação)

Gráfico dos Componentes Principais

- Podem:
  - √ revelar observações suspeitas
  - √ fornecer verificações da hipótese de normalidade

- São combinações das variáveis originais:
  - √ Se as observações provém de população normal multivariada, é razoável esperar que as componentes sejam aproximadamente normais
  - √ Se forem usadas como entrada em análises adicionais
    - – Verificar se as 1ª.s componentes são aproximadamente normais
- As últimas componentes principais podem ajudar a apontar observações suspeitas

Resumo

- Procedimento auxiliar na verificação de normalidade
  - √ Construir diagrama de dispersão para os pares dos primeiros componentes principais
  - √ Construir Q-Q plots para os valores amostrais gerados por cada componente principal
- Identificação de observações suspeitas:
  - √ Construir diagramas de dispersão e Q-Q plots para as últimas componentes principais.

Exemplo

- Pesquisa de percepção de marcas:
  - √ Avaliação de características relacionadas à marca
  - √ Pergunta:
    - – Quão [atributo] é a [marca]?
  - √ Variáveis:
    - – Atributos: *perform*, *leader*, *latest*, *fun*, *serious*, *bargain*, *value*, *trendy*, *rebuy*
      - ▪ Níveis : 1 (menos) a 10 (mais)
    - – brand:
      - ▪ Níveis: *a* a *j*
  - √ Respondentes*: 100*
  - √ Dados*: BD_multivariada.xls/brand*

- Características das marcas – Perguntas:

Atributo	Exemplo de pergunta
perform	Marca tem um forte desempenho?
leader	Marca é líder no mercado?
latest	Marca tem os produtos mais recentes?
fun	Marca é divertida?
serious	Marca é séria?
bargain	Produtos da marca são uma pechincha
value	Produtos da marca possuem um bom valor?
trendy	Marca está na moda?
rebuy	Eu compraria a marca novamente?

- Fonte: Chapman, C.; Feit, E. M. *R for marketing research and analytics*, Springer, 2015

103

- Conjunto de dados:

```
> brand.ratings <- read.csv("rintro-chapter8.csv")
> head(brand.ratings)

 perform leader latest fun serious bargain value trendy rebuy brand
1       2      4      8   8       2       9     7      4     6     a
2       1      1      4   7       1       1     1      2     2     a
3       2      3      5   9       2       9     5      1     6     a
4       1      6     10   8       3       4     5      2     1     a
5       1      1      5   8       1       9     9      1     1     a
6       2      8      9   5       3       8     7      1     2     a
```

√ Estrutura:

```
> str(brand.ratings)

'data.frame':   1000 obs. of  10 variables:
 $ perform: int  2 1 2 1 1 2 1 2 2 3 ...
 $ leader : int  4 1 3 6 1 8 1 1 1 1 ...
 $ latest : int  8 4 5 10 5 9 5 7 8 9 ...
 $ fun    : int  8 7 9 8 8 5 7 5 10 8 ...
 $ serious: int  2 1 2 3 1 3 1 2 1 1 ...
 $ bargain: int  9 1 9 4 9 8 5 8 7 3 ...
 $ value  : int  7 1 5 5 9 7 1 7 7 3 ...
 $ trendy : int  4 2 1 2 1 1 1 7 5 4 ...
 $ rebuy  : int  6 2 6 1 1 2 1 1 1 1 ...
 $ brand  : Factor w/ 10 levels "a","b","c","d",..: 1 1 1 1 1 1 1 1 1 1 ...
```


- Resumo dos dados:

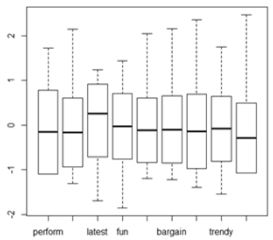
```
> summary(brand.ratings)

 perform          leader           latest            fun        
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 4.000   1st Qu.: 4.000  
 Median : 4.000   Median : 4.000   Median : 7.000   Median : 6.000  
 Mean   : 4.488   Mean   : 4.417   Mean   : 6.195   Mean   : 6.068  
 3rd Qu.: 7.000   3rd Qu.: 6.000   3rd Qu.: 9.000   3rd Qu.: 8.000  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000  

    serious          bargain          value           trendy      
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.00  
 1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 3.00  
 Median : 4.000   Median : 4.000   Median : 4.000   Median : 5.00  
 Mean   : 4.323   Mean   : 4.259   Mean   : 4.337   Mean   : 5.22  
 3rd Qu.: 6.000   3rd Qu.: 6.000   3rd Qu.: 6.000   3rd Qu.: 7.00  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.00  

     rebuy           brand     
 Min.   : 1.000   a      :100  
 1st Qu.: 1.000   b      :100  
 Median : 3.000   c      :100  
 Mean   : 3.727   d      :100  
 3rd Qu.: 5.000   e      :100  
 Max.   :10.000   f      :100  
                  (Other):400  
```

105

- Padronização dos dados:

√ Melhora a comparabilidade entre indivíduos

```
> brand.sc <- brand.ratings
> brand.sc[, 1:9] <- scale(brand.ratings[, 1:9])
> summary(brand.sc)

  perform            leader            latest             fun         
 Min.   :-1.0888   Min.   :-1.3100   Min.   :-1.6878   Min.   :-1.84677  
 1st Qu.:-1.0888   1st Qu.:-0.9266   1st Qu.:-0.7131   1st Qu.:-0.75358  
 Median :-0.1523   Median :-0.1599   Median : 0.2615   Median :-0.02478  
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000  
 3rd Qu.: 0.7842   3rd Qu.: 0.6069   3rd Qu.: 0.9113   3rd Qu.: 0.70402  
 Max.   : 1.7206   Max.   : 2.1404   Max.   : 1.2362   Max.   : 1.43281  
```



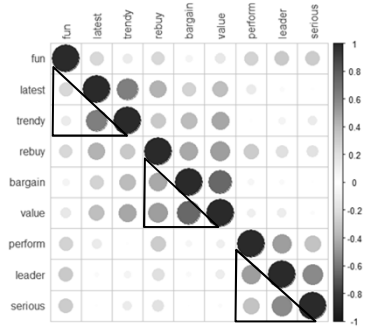
106

- Matriz de correlação dos dados originais:
  - √ Há grupos de variáveis mais fortemente correlacionadas?

```
> cor(brand.ratings[,1:9], use = "complete.obs")

             perform      leader      latest         fun      serious
perform   1.000000000 0.50020206 -0.122445813 -0.2563323 0.359172206
leader    0.500202058 1.00000000  0.026890447 -0.2903576 0.571215126
latest   -0.122445813 0.02689045  1.000000000  0.2451545 0.009951527
fun      -0.256332316 -0.29035764 0.245154457 1.0000000 -0.281097443
serious   0.359172206 0.57121513  0.009951527 -0.2810974 1.000000000
bargain   0.057129372 0.03309405 -0.254419016 -0.0665528 -0.002655590
value     0.101946104 0.11831017 -0.342713717 -0.1452185 0.023756556
trendy    0.008733494 0.06651244  0.627627667  0.1279736 0.121009377
rebuy     0.306658801 0.20870036 -0.397180225 -0.2371607 0.180702720
             bargain       value      trendy      rebuy
perform   0.05712937  0.10194610  0.008733494  0.3066588
leader    0.03309405  0.11831017  0.066512436  0.2087004
latest   -0.25441902 -0.34271372  0.627627667 -0.3971802
fun      -0.06655280 -0.14521849  0.127973639 -0.2371607
serious  -0.00265559  0.02375656  0.121009377  0.1807027
bargain   1.00000000  0.73962672 -0.350533746  0.4673811
value     0.73962672  1.00000000 -0.434534536  0.5059617
trendy   -0.35053375 -0.43453454  1.000000000 -0.2982462
rebuy     0.46738109  0.50596166 -0.298246195  1.0000000
```


- *Correlation plot*:
  - √ Auxilia visualização das correlações

```
> library(corrplot)
> corrplot(cor(brand.sc[, 1:9]), order = "hclust")
```



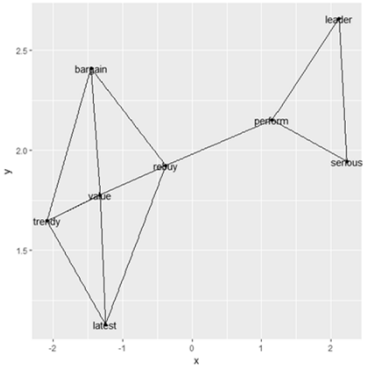
- √ Dados aparentam se agrupar em três grupos:
  - *fun*/*latest*/*trendy*
  - *rebuy*/*bargain*/*value*
  - *perform*/*leader*/*serious*

- *Correlation network plots*:
  - √ Preparação e criação dos objetos:

```
> library(tidyverse)
> library(corrr)
> library(igraph)
> library(ggraph)
> tidy_cors <- brand.sc[,1:9] %>%
+   correlate() %>%
+   stretch()
> tidy_cors
> graph_cors <- tidy_cors %>%
+   filter(abs(r) > .3) %>%
+   graph_from_data_frame(directed = FALSE)
> graph_cors
```


- *Correlation network plot*:

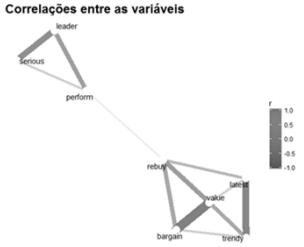
```
> ggraph(graph_cors) +
+   geom_edge_link() +
+   geom_node_point() +
+   geom_node_text(aes(label = name))
```



- √ Variáveis aparentam se agrupar em dois grupos:

- *Correlation network plot*:

```
> ggraph(graph_cors) +
+   geom_edge_link(aes(edge_alpha = abs(r), edge_width = abs(r), color =
r)) +
+   guides(edge_alpha = "none", edge_width = "none") +
+   scale_edge_colour_gradientn(limits = c(-1, 1), colors =
c("firebrick2", "dodgerblue2")) +
+   geom_node_point(color = "white", size = 5) +
+   geom_node_text(aes(label = name), repel = TRUE) +
+   theme_graph() +   labs(title = "Correlações entre as variáveis")
```



111

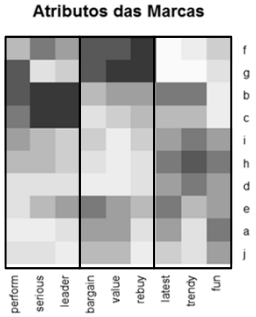
- Qual a média da marca em cada atributo?

```
> brand.mean <- aggregate(. ~brand, data = brand.sc, mean)
> rownames(brand.mean) <- brand.mean[, 1] # use brand for the row names
> brand.mean <- brand.mean[, -1] # remove brand name column
> brand.mean

      perform     leader     latest        fun     serious     bargain
a -0.88591874 -0.5279035  0.4109732  0.6566458 -0.91894067  0.21409609
b  0.93087022  1.0707584  0.7261069 -0.9722147  1.18314061  0.04161938
c  0.64992347  1.1627677 -0.1023372 -0.8446753  1.22273461 -0.60704302
d -0.67989112 -0.5930767  0.3524948  0.1865719 -0.69217505 -0.88075605
e -0.56439079  0.1928362  0.4564564  0.2958914  0.04211361  0.55155051
f -0.05868665  0.2695106 -1.2621589 -0.2179102  0.58923066  0.87400696
g  0.91838369 -0.1675336 -1.2849005 -0.5167168 -0.53379906  0.89650392
h -0.01498383 -0.2978802  0.5019396  0.7149495 -0.14145855 -0.73827529
i  0.33463879 -0.3208825  0.3557436  0.4124989 -0.14865746 -0.25459062
j -0.62994504 -0.7885965 -0.1543180  0.2849595 -0.60218870 -0.09711188
       value      trendy       rebuy
a  0.18469264 -0.52514473 -0.59616642
b  0.15133957  0.74030819  0.23697320
c -0.44067747  0.02552787 -0.13243776
d -0.93263529  0.73666135 -0.49398892
e  0.41816415  0.13857986  0.03654811
f  1.02268859 -0.81324496  1.35699580
g  1.25616009 -1.27639344  1.36092571
```

112

- *Heat map*:
  - √ Pontos coloridos pela intensidade

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(as.matrix(brand.mean), col = brewer.pal(9, "GnBu"), trace =
"none",
+ key = FALSE, dend = "none",
+ main = "\n\n\n\n\nAtributos das Marcas")
```



- √ Ordenação para enfatizar similaridades e padrões
- √ Há grupos e relações de atributos e marcas:
  - *rebuy*/*bargain*/*value*

  (Marca com valor alto em um, tende a ter valor alto no outro)

113

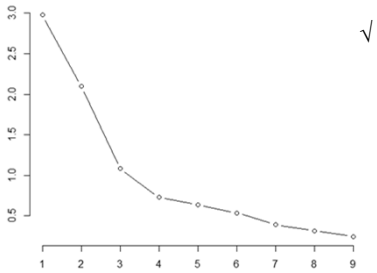
- Componentes principais:
  - √ Reduzir a complexidade dos dados
    - Retenção e análise de apenas um subconjunto das componentes que expliquem grande parte da variabilidade dos dados

```
> > brand.pc <- prcomp(brand.sc[, 1:9])
> summary(brand.pc)

Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     1.726 1.4479 1.0389 0.8528 0.79846 0.73133 0.62458
Proportion of Variance 0.331 0.2329 0.1199 0.0808 0.07084 0.05943 0.04334
Cumulative Proportion  0.331 0.5640 0.6839 0.7647 0.83554 0.89497 0.93831
                          PC8     PC9
Standard deviation     0.55861 0.49310
Proportion of Variance 0.03467 0.02702
Cumulative Proportion  0.97298 1.00000
```

114

√ *Scree plot*:

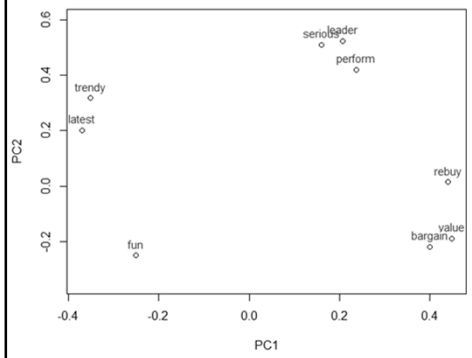
```
> plot(brand.pc, type = "l")
```



√ As 2 ou 3 primeiras componentes explicam a maior parte da variabilidade dos dados

116

- *Plot* dos coeficientes das duas primeiras componentes:



Regiões:
√ Liderança:
- *serious*, *leader* e *perform*

√ Valor:
- *rebuy*, *value* e *bargain*

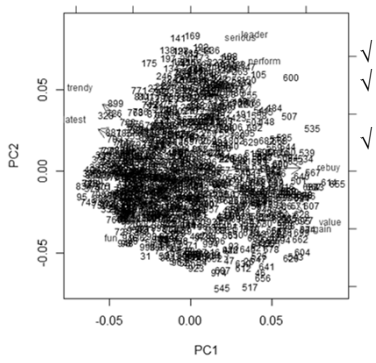
√ Tendência:
- *trendy* e *latest*

√ Isolado:
- *fun*

117

- *Biplot* das duas primeiras componentes:
  √ Auxilia visualização das correlações

```
> biplot(brand.pc, cex = 0.75, expand = 1, arrow.len = 0.15)
```



√ 4 regiões
√ Plot muito denso
- todos os respondentes

√ Solução:
- Executar ACP usando avaliações agregadas por marca

118

- Médias dos atributos por marca:

```
> brand.mean <- aggregate(. ~brand, data = brand.sc, mean)
> rownames(brand.mean) <- brand.mean[, 1] # use brand for the row names
> brand.mean <- brand.mean[, -1] # remove brand name column
> brand.mean

      perform     leader     latest        fun     serious     bargain
a -0.88591874 -0.5279035  0.4109732  0.6566458 -0.91894067  0.21409609
b  0.93087022  1.0707584  0.7261069 -0.9722147  1.18314061  0.04161938
c  0.64992347  1.1627677 -0.1023372 -0.8446753  1.22273461 -0.60704302
d -0.67989112 -0.5930767  0.3524948  0.1865719 -0.69217505 -0.88075605
e -0.56439079  0.1928362  0.4564564  0.2958914  0.04211361  0.55155051
f -0.05868665  0.2695106 -1.2621589 -0.2179102  0.58923066  0.87400696
g  0.91838369 -0.1675336 -1.2849005 -0.5167168 -0.53379906  0.89650392
h -0.01498383 -0.2978802  0.5019396  0.7149495 -0.14145855 -0.73827529
i  0.33463879 -0.3208825  0.3557436  0.4124989 -0.14865746 -0.25459062
j -0.62994504 -0.7885965 -0.1543180  0.2849595 -0.60218870 -0.09711188
       value      trendy       rebuy
a  0.18469264 -0.52514473 -0.59616642
b  0.15133957  0.74030819  0.23697320
c -0.44067747  0.02552787 -0.13243776
d -0.93263529  0.73666135 -0.49398892
e  0.41816415  0.13857986  0.03654811
f  1.02268859 -0.81324496  1.35699580
g  1.25616009 -1.27639344  1.36092571
h -0.78254646  0.86430070 -0.60402622
i -0.80339213  0.59078782 -0.20317603
j -0.07379367 -0.48138267 -0.96164748
```

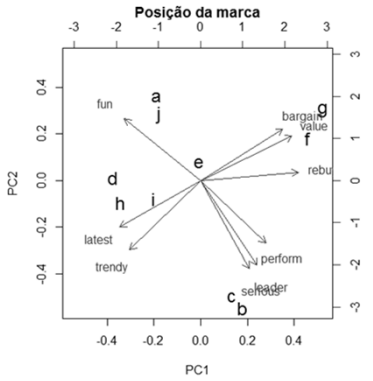
119

- ACP dos dados agregados por média:
  - √ Realizada nova padronização:
    - – Médias agregadas têm escala um pouco diferente que os dados padronizados

```
> brand.mu.pc <- prcomp(brand.mean, scale = TRUE)
> summary(brand.mu.pc)

Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     2.1345 1.7349 0.7690 0.61498 0.50983 0.36662 0.21506
Proportion of Variance 0.5062 0.3345 0.0657 0.04202 0.02888 0.01493 0.00514
Cumulative Proportion  0.5062 0.8407 0.9064 0.94842 0.97730 0.99223 0.99737
                           PC8     PC9
Standard deviation     0.14588 0.04867
Proportion of Variance 0.00236 0.00026
Cumulative Proportion  0.99974 1.00000
```

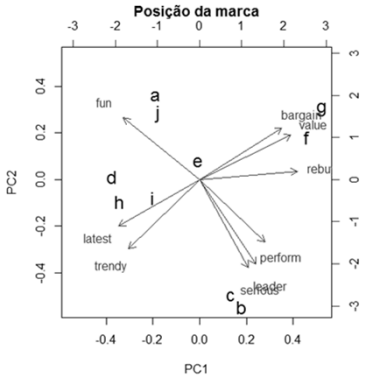
  - √ Primeiras duas componentes com 84% da variabilidade total das avaliações de média

- Mapa de percepção para médias agregadas



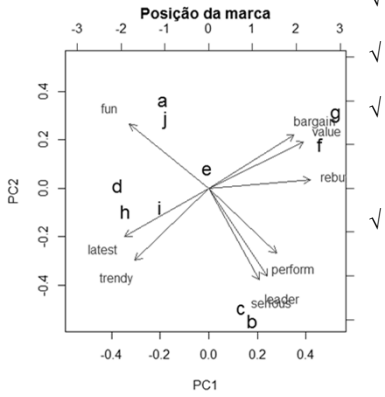
  - √ Rotação diferente dos atributos
    - – Posição espacial é arbitrária
  - √ Mesmo agrupamento global de atributos e estrutura de associações
  - √ Posição das variáveis nas componentes consistente com ACP com todas as observações
    - – Pode-se prosseguir com a interpretação do gráfico

- Grupos de atributos



  - √ *serious* e *leader* estão próximos
    - – Posição espacial é arbitrária
  - √ *fun* está distante das outras variáveis e em posição oposta aos atributos de liderança (*serious* e *leader*)

- Grupos de marcas



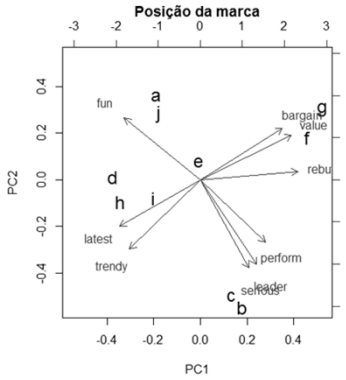
  - √ Marcas **f** e **g**:
    - – Fortes em *value*
  - √ Marcas **a** e **j**:
    - – Relativamente fortes em *fun*
  - √ Marca **e**:
    - – Aparenta não estar bem diferenciada em qualquer das dimensões
  - √ Pode ser bom ou mau:
    - – Busca atender muitos consumidores (marca segura)
    - – Não tem forte percepção de diferenciação
      - Movimentar a marca em alguma direção do mapa

- Diferenças entre a marca **c** e **e**:

```
> brand.mean["c",] - brand.mean["e",]

   perform    leader     latest       fun  serious   bargain      value
c 1.214314 0.9699315 -0.5587936 -1.140567 1.180621 -1.158594 -0.8588416
     trendy      rebuy
c -0.113052 -0.1689859
```

  - √ **e** é mais forte que **c** em *value* e *fun*
  - √ **c** é mais forte que **e** em *perform* e *serious*
    - Aspectos do produto ou da mensagem para **e** reforçar

- Outra opção:
  - √ Não seguir outra marca, mas obter espaço diferenciado



  - √ Gap entre grupo **b** e **c** e **f** e **g**
    - Área *value-leader* ou similar
  - √ Como se posicionar nessa nova área?

- Gap *value-leader*:
  - √ Assumindo que o gap reflete aproximadamente a média dessas 4 marcas

```
> colMeans(brand.mean[c("b", "c", "f", "g"), ]) - brand.mean["e", ]

   perform    leader     latest        fun   serious    bargain      value
e 1.174513 0.3910396 -0.9372789 -0.9337707 0.5732131 -0.2502787 0.07921355
      trendy     rebuy
e -0.4695304 0.6690661
```

  - √ Para marca **e** posicionar-se no gap:
    - Poderia focar *performance* e reduzir ênfase em *latest* e *fun*

- Comentário:
  - √ Mapas de percepção podem também ser usados em:
    - Pesquisa de avaliação das marcas
    - Utilizar dados objetivos:
      - Preço, medidas físicas ou combinações de ambos

Análise Fatorial

Análise Fatorial

- Objetivo:
  - √ Descrever as relações de covariância entre muitas variáveis em termos de poucas quantidades aleatórias subjacentes e não observáveis
- Motivação:
  - √ Variáveis de um grupo altamente correlacionadas entre si, mas com pequenas correlações de outros grupos
  - √ É concebível que cada grupo de variáveis represente um fator (ou construto) que seja o responsável pelas correlações observadas

- Análise fatorial:
  - √ Pode ser considerada uma extensão da Análise de Componentes Principais
    - – Ambas são tentativas de aproximar S.
    - – A aproximação baseada em Análise Fatorial é mais elaborada
  - √ Questão principal:
    - – Dados são consistentes com a estrutura prescrita?

- Análise Fatorial Exploratória:
  - √ Busca encontrar os fatores subjacentes às variáveis originais amostradas
  - √ Em geral, efetuada quando não se tem noção clara da quantidade de fatores do modelo e nem do que representam
- Análise Fatorial Confirmatória:
  - √ Tem-se em mãos um modelo fatorial pré-especificado (modelo hipotético) e deseja-se verificar se é aplicável ou consistente com os dados amostrais de que dispõe

Modelo Fatorial Ortogonal via Matriz de Correlações

- Seja o vetor aleatório

$$\mathbf{X}' = [X_1, X_2, \ldots, X_p].$$

com vetor de médias **μ**, matriz de covariâncias é **Σ**, e matriz de correlações **P**.

- Sejam as variáveis originais padronizadas: $Z_i = \frac{X_i - \mu_i}{\sqrt{\sigma_{ii}}}$

√ **P** é a matriz de covariâncias do vetor aleatório **Z**, cujos componentes são as variáveis padronizadas

- Modelo Fatorial Ortogonal
  - √ Construído via a matriz de correlação populacional
  - √ Relaciona linearmente as variáveis padronizadas e os m fatores comuns (que são desconhecidos)
  - √ Fatores são variáveis independentes

- Equações do modelo:

$$\begin{aligned} Z_1 &= l_{11}F_1 + l_{12}F_2 + \cdots + l_{1m}F_m + \epsilon_1 \\ Z_2 &= l_{21}F_1 + l_{22}F_2 + \cdots + l_{2m}F_m + \epsilon_2 \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ Z_p &= l_{p1}F_1 + l_{p2}F_2 + \cdots + l_{pm}F_m + \epsilon_p \end{aligned}$$

√ Em notação matricial:

$$\mathbf{V}(\mathbf{X} - \boldsymbol{\mu}) = \mathbf{LF} + \boldsymbol{\epsilon}.$$

$$\mathbf{V} = \text{diagonal}[\sigma_1, \sigma_2, \ldots, \sigma_p].$$

$$\mathbf{L}_{p\times m} = \begin{bmatrix} l_{11} & l_{12} & \ldots & l_{1m} \\ l_{21} & l_{22} & \ldots & l_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ l_{p1} & l_{p2} & \ldots & l_{pm} \end{bmatrix}. \quad \mathbf{F}_{m\times 1} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}. \quad \boldsymbol{\epsilon}_{p\times 1} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{bmatrix}.$$

- Modelo fatorial:

$$\mathbf{V}(\mathbf{X} - \boldsymbol{\mu}) = \mathbf{LF} + \boldsymbol{\epsilon}.$$

√ **F**: vetor aleatório contendo m fatores
  - Essas variáveis latentes precisam ser identificadas

√ **ε**: vetor dos erros aleatórios
  - Erros de medida e variação de Z_i que não é explicada pelos fatores comuns

√ L: matriz de loadings fatoriais
  - l_{ij}: representa o grau de relacionamento entre Z_i e F_j.

√ O modelo de análise fatorial assume que as variáveis Z_i estão relacionadas linearmente com os fatores
  - Variáveis originais padronizadas são representadas por p+m variáveis não observáveis

Modelo de Fatores Ortogonais

- Suposições:
  i. Todos os fatores tem média zero $\mathrm{E}[\mathbf{F}] = \mathbf{0}$.
  ii. Todos os fatores são não correlacionados e tem variância um. $\mathrm{Cov}[\mathbf{F}] = \mathbf{I}_m$.
  iii. Todos os erros tem média igual a zero $\mathrm{E}[\boldsymbol{\epsilon}] = \mathbf{0}$.
  iv. Erros são não correlacionados entre si e não necessariamente tem a mesma variância

$$\mathrm{Cov}[\boldsymbol{\epsilon}] = \mathrm{diagonal}(\psi_1, \psi_2, \ldots, \psi_p). \qquad \begin{array}{l} \mathrm{Var}[\epsilon_j] = \psi_j \\ \mathrm{Cov}(\epsilon_i, \epsilon_j) = 0, \quad \forall i \neq j. \end{array}$$

v. Os vetores **e** e **F** são independentes

$$\mathrm{Cov}(\boldsymbol{\epsilon}_{p\times 1}, \mathbf{F}_{m\times 1}) = \mathrm{E}[\boldsymbol{\epsilon}\mathbf{F}'] = \mathbf{0}.$$

√ **F** e **ε** são duas fontes de variação distintas, relacionadas às variáveis padronizadas Z_i, não havendo qualquer relacionamento entre estas fontes de informação.

- Assumido o modelo, **P** pode ser reparametrizada

$$\mathbf{P}_{p\times p} = \mathbf{L}\mathbf{L}' + \boldsymbol{\Psi}.$$

√ O objetivo é encontrar as matrizes $\mathbf{L}_{pxm}$ e $\boldsymbol{\psi}_{pxp}$ que possam representar a matriz $\mathbf{P}_{pxp}$.

– Há matrizes de correlação que não podem ser decompostas na forma do modelo

- Consequências da decomposição fatorial de **P**:

√ Variância de Z_i é decomposta em duas partes:

$$\mathrm{Var}[Z_i] = h_i^2 + \psi_i$$
$$\text{onde } h_i^2 = l_{i1}^2 + l_{i2}^2 + \cdots + l_{im}^2.$$

– h_i^2: comunalidade
  ▪ variabilidade explicada pelos m fatores que é uma fonte comum de variação de Z_i.

– ψ_i: variância específica
  ▪ Parte da variabilidade de Zi associada apenas ao erro aleatório

√ Covariâncias entre variáveis e fatores

$$\mathrm{Cov}(Z_i, Z_k) = l_{i1}l_{k1} + l_{i2}l_{k1} + \cdots + l_{im}l_{km},\ i, k = 1, 2, \ldots, p, i \neq k.$$
$$\mathrm{Cov}(Z_i, F_j) = \mathrm{Corr}(Z_i, F_j) = l_{ij},\ i = 1, 2, \ldots, p \text{ e } i = 1, 2, \ldots, m.$$

√ Proporção da variância total explicada pelo fator F_j:

$$\text{Proporção explicada}_{F_j} = \frac{\sum_{i=1}^{p} l_{ij}^2}{p}.$$

Métodos de Estimação de L e ψ

- Escolhe-se o valor de m
- Métodos de estimação das matrizes **L** e **ψ**:
  - √ Método de componentes principais
    - – Em geral, utilizado como um análise exploratória dos dados, em termos dos fatores subjacentes
  - √ Método de fatores principais
    - – Refinamento do método das componentes principais
  - √ Método da máxima verossimilhança
    - – Indicado apenas quando **Z** tem distribuição normal

Método das Componentes Principais

- Matrizes **L** e **ψ** serão estimadas por:

$$\hat{\mathbf{L}} = \left[\sqrt{\hat{\lambda}_1}\hat{\mathbf{e}}_1, \sqrt{\hat{\lambda}_2}\hat{\mathbf{e}}_2, \ldots, \sqrt{\hat{\lambda}_m}\hat{\mathbf{e}}_m\right].$$

$$\hat{\boldsymbol{\psi}} = \text{diagonal}\left(\mathbf{R}_{p\times p} - \hat{\mathbf{L}}_{p\times m}\hat{\mathbf{L}}'_{p\times m}\right).$$

  - √ Aproximação de R

$$\mathbf{R}_{p\times p} \approx \hat{\mathbf{L}}_{p\times m}\hat{\mathbf{L}}'_{p\times m} + \hat{\boldsymbol{\psi}}.$$

- Matriz residual:

$$\mathbf{MRes} = \mathbf{R}_{p\times p} - \left(\hat{\mathbf{L}}_{p\times m}\hat{\mathbf{L}}'_{p\times m} + \hat{\boldsymbol{\psi}}\right).$$

  - √ Pode servir como critério de avaliação do modelo
    - – Seus valores deveriam ser próximos de zero
    - – Matriz é nula somente quando o valor de m é igual a p
  - √ Os elementos da diagonal da matriz R são reproduzidos exatamente pela reprodução do modelo
    - – O mesmo não ocorre para os outros elementos da matriz **R** (covariâncias das variáveis Z_i e Z_j)

- Método das componentes principais na estimação de **LL'** e **ψ**.

$$\text{Proporção explicada}_{F_j} = \frac{\sum_{i=1}^{p} l_{ij}^2}{p}.$$

  - √ Representa o quanto cada fator consegue captar da variabilidade original das variáveis Z_i.

Método da Máxima Verossimilhança

- Só pode ser utilizado quando a forma da distribuição de probabilidades é conhecida
- Suposição:
  - √ Vetor aleatório **X** tem distribuição normal p-variada
  - √ Consequência:
    - Vetor das variáveis padronizadas é normal p-variado
    - Fatores tem distribuição normal multivariada com vetor de médias zero e matriz de covariâncias $\mathbf{I}_m$
    - Erros tem distribuição normal p-variada com vetor de médias zero e matriz de covariâncias $\boldsymbol{\psi}$.

144

- A função de verossimilhança é expressa como

$$L(\mathbf{0},\mathbf{P}) = \frac{1}{(2\pi)^{np/2}|\mathbf{LL}' + \boldsymbol{\psi}|^{n/2}} \exp\left\{-\frac{1}{2}\sum_{j=1}^{n}\mathbf{z}_j'\left(\mathbf{LL}' + \boldsymbol{\psi}\right)^{-1}\mathbf{z}_j\right\}.$$

  - √ A função de verossimilhança depende da matrizes **L** e **ψ**, através da matriz de correlação **P**.
  - √ As estimativas de máxima verossimilhança de $\hat{\mathbf{L}}$ e $\hat{\boldsymbol{\psi}}$ são as matrizes **L** e **ψ** que maximizam a função de verossimilhança.
  - √ Maximização é feita por métodos numéricos
  - √ Método mais sofisticado que os métodos de componentes e fatores principais
    - Produz estimativas mais precisas

145

- Cuidados:
  - √ Está fundamentado na suposição de normalidade multivariada dos vetores **Z**, **F** e **ε**.
    - Apenas a normalidade do vetor **Z** pode ser investigada a priori a partir dos dados amostrais
    - Fatores e erros são variáveis aleatórias não observáveis

146

- Valor de m:
  - √ Método de máxima verossimilhança
    - Mudança de valor de m altera as estimativas dos loadings
  - √ Método de componentes principais
    - Aumento no valor de m não altera os loadings para os fatores obtidos anteriormente
  - √ Quando os dados provêm de distribuição normal multivariada
    - Usar método de componentes principais como análise exploratória dos fatores e estimação do valor provável de m
    - Posteriormente, qualidade da solução inicial poderá ser melhorada pelo uso do método de máxima verossimilhança

147

- Dados omissos:
  - √ São considerados apenas os elementos amostrais com observações completas
    (Análise de componentes principais e análise fatorial)
  - √ Caso haja muitas observações com dados omissos em algumas variáveis, deve-se avaliar até que ponto as análises são válidas.

Rotação dos Fatores

- A matriz de covariância Σ é reproduzida pelos loadings fatoriais obtidos por transformação ortogonal, da mesma maneira que os loadings iniciais.
  - √ Matriz de covariâncias estimada

$$\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{\mathbf{\Psi}} = \hat{\mathbf{L}}\mathbf{T}\mathbf{T}'\hat{\mathbf{L}}' + \hat{\mathbf{\Psi}} = \hat{\mathbf{L}}^*\hat{\mathbf{L}}^{*'} + \hat{\mathbf{\Psi}}$$

  - √ TT' = T'T = I
  - √ $\hat{L}$*: matriz de loadings rotacionados
  - √ A matriz de resíduos permanece a mesma ($\hat{h}_i^2$ e $\hat{\Psi}_i$)

$$\mathbf{S}_n - \hat{\mathbf{L}}\hat{\mathbf{L}}' - \hat{\mathbf{\Psi}} = \mathbf{S}_n - \hat{\mathbf{L}}^*\hat{\mathbf{L}}^{*'} - \hat{\mathbf{\Psi}}$$

  - √ Do ponto de vista estatístico é irrelevante obter $\hat{L}$ ou $\hat{L}$*

- Comentários:
  - √ Com a rotação, busca-se uma estrutura mais simples
    - loadings originais podem não ter fácil interpretação
  - √ Ideal: encontrar um padrão de loadings tais que cada variável carregue-se fortemente em um único fator
    (com loadings moderados nos outros fatores)
  - √ Nem sempre é possível obter esta estrutura mais simples

Critérios de Rotação

- Ideal:
  - √ Transformação que fizesse os loadings de cada $\mathbf{Z}_i$ ter valor grande em apenas um dos fatores e valores pequenos (ou moderados) nos outros
    - Para facilitar a interpretação dos fatores
- Alguns critérios para encontrar matriz ortogonal:
  - √ Varimax
  - √ Quartimax
  - √ Orthomax

- Qualidade de ajuste
  - √ A rotação não acrescenta nenhuma melhoria em relação ao ajuste original
    - – Matriz residual original não é alterada pela transformação ortogonal
    - – Valores estimados de comunalidade e variâncias específicas permanecem inalterados
- Interpretação:
  - √ Novos fatores podem ser de mais fácil interpretação
- Quando a solução sem rotação já é de boa qualidade, não se recomenda rotação
  - √ Solução rotacionada pode ser pior

- Critério Varimax:
  - √ É um dos mais utilizados
  - √ Em geral, produz soluções mais simples
- Critério Quartimax
  - √ Tem tendência de gerar fatores, onde todas as variáveis têm loadings elevados
- Critério Orthomax
  - √ É uma média ponderada dos dois outros métodos

Matriz de Resíduos

- A observação da matriz de resíduos:
  - √ Muitas vezes, pode indicar quando o número de fatores está superdimensionado
  - √ Ex.:
    - – Se *m* não for muito pequeno e a matriz de resíduos estiver próxima de zero, recomenda-se testar outras soluções para *m* menores que o valor já especificado

- Importante:
  - √ Análise fatorial deve ser utilizada apenas se utilizada em situações em que as variáveis originais são correlacionadas
  - √ Consequência:
    - – Evitar soluções com *m* elevado tal que determinados fatores fiquem relacionados com uma única variável original
  - √ Em situações em que aparecem fatores relacionados a uma única variável Z_i é recomendável retirar a variável Z_i e reestimar o modelo de análise fatorial

Exemplo

- Pesquisa de percepção de marcas:
  - √ Avaliação de características relacionadas à marca
  - √ Pergunta:
    - Quão [atributo] é a [marca]?
  - √ Variáveis:
    - Atributos: *perform*, *leader*, *latest*, *fun*, *serious*, *bargain*, *value*, *trendy*, *rebuy*
      - Níveis : 1 (menos) a 10 (mais)
    - brand:
      - Níveis: *a* a *j*
  - √ Respondentes*: 100*
  - √ Dados*: BD_multivariada.xls/brand*

Características das marcas – Perguntas:

Atributo	Exemplo de pergunta
perform	Marca tem um forte desempenho?
leader	Marca é líder no mercado?
latest	Marca tem os produtos mais recentes?
fun	Marca é divertida?
serious	Marca é séria?
bargain	Produtos da marca são uma pechincha
value	Produtos da marca possuem um bom valor?
trendy	Marca está na moda?
rebuy	Eu compraria a marca novamente?

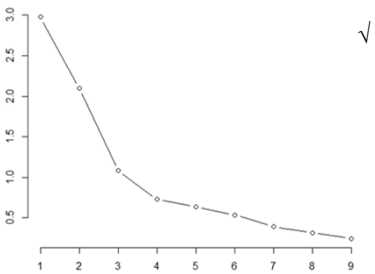
- Fonte: Chapman, C.; Feit, E. M. *R for marketing research and analytics*, Springer, 2015

Determinação da Quantidade de Fatores

- *Scree plot*
- Reter fatores associados a autovalores maiores que 1
  - √ Quantidade de variância que pode ser atribuída a uma única variável
  - √ Fator que captura variância menor que a de uma variável é considerado desprezível

√ *Scree plot*:

```
> plot(brand.pc, type = "l")
```



√ As 2 ou 3 primeiras componentes explicam a maior parte da variabilidade dos dados

- Testes para determinação de m:

```
> # scree tests
> library(nFactors)
> nScree(brand.sc[, 1:9])

 noc naf nparallel nkaiser
1   3   2         3       3
```

√ Aplicando 4 métodos, 3 sugerem que os dados têm 3 fatores

- Autovalores:

```
> # autovalores
> eigen(cor(brand.sc[, 1:9]))$values

[1] 2.9792956 2.0965517 1.0792549 0.7272110 0.6375459 0.5348432 0.3901044
[8] 0.3120464 0.2431469
```

√ Os 3 primeiros autovalores são maiores que 1.

- Escolha final:
  - √ Depende da utilidade da análise
- Verificar algumas soluções com 2 e 3 fatores

- Solução com 2 fatores:

```
> # Solução com 2 fatores
> factanal(brand.sc[, 1:9], factors = 2) # perform maximum-likelihood AF
>    # default: varimax

Uniquenesses:
perform  leader  latest     fun serious bargain   value  trendy   rebuy
  0.635   0.332   0.796   0.835   0.527   0.354   0.225   0.708   0.585

Loadings:
         Factor1 Factor2
perform            0.600
leader             0.818
latest  -0.451
fun     -0.137  -0.382
serious            0.686
bargain  0.803
value    0.873   0.117
trendy  -0.534
rebuy    0.569   0.303

                Factor1 Factor2
SS loadings       2.245   1.759
Proportion Var    0.249   0.195
Cumulative Var    0.249   0.445
```

√ Fator 1: Valor
- Loadings fortes em *bargain* e *value*.

√ Fator 2: Liderança
- Cargas fatoriais fortes em *perform*, *leader* e *serious*.

Não parece ser uma má solução

- Solução com 3 fatores:

```
> # Solução com 3 fatores
> factanal(brand.sc[, 1:9], factors = 3)

Uniquenesses:
perform  leader  latest     fun serious bargain   value  trendy   rebuy
  0.624   0.327   0.005   0.794   0.530   0.302   0.202   0.524   0.575

Loadings:
         Factor1 Factor2 Factor3
perform            0.607
leader             0.810   0.106
latest  -0.163             0.981
fun             -0.398   0.205
serious            0.682
bargain  0.826          -0.122
value    0.867          -0.198
trendy  -0.356           0.586
rebuy    0.499   0.296  -0.298

                Factor1 Factor2 Fac
SS loadings       1.853   1.752   1.510
Proportion Var    0.206   0.195
Cumulative Var    0.206   0.401
```

√ Fator 1: Valor
- Cargas fortes em *bargain* e *value*.

√ Fator 2: Liderança no mercado
- Cargas fatoriais fortes em *perform*, *leader* e *serious*.

√ Fator 3: Atualidade
- Cargas fatoriais fortes em *latest* e *trendy*.

Fator adicionado é interpretável

- Comparação dos modelos:

```
> # Solução com 2 fatores

Loadings:
        Factor1 Factor2
perform          0.600
leader           0.818
latest  -0.451
fun     -0.137  -0.382
serious          0.686
bargain  0.803
value    0.873   0.117
trendy  -0.534
rebuy    0.569   0.303
```

```
> # Solução com 3 fatores

Loadings:
        Factor1 Factor2 Factor3
perform          0.607
leader           0.810   0.106
latest  -0.163           0.981
fun             -0.398   0.205
serious          0.682
bargain  0.826          -0.122
value    0.867          -0.198
trendy  -0.356           0.586
rebuy    0.499   0.296  -0.298
```

  - √ Modelo com 3 fatores:
    - Acrescenta na compreensão dos dados conceito claramente interpretável
    - Está consistente com sugestões:
      - (*scree plot*, autovalores , *scree tests*, mapas de percepção)
    - Aparenta ser superior ao de 2 fatores porque os fatores são melhor interpretáveis

Rotação

- Objetivo:
  - √ Obter novas cargas fatoriais com a mesma proporção de variabilidade
- Tipos:
  - √ Ortogonal:
    - Construtos são independentes
  - √ Oblíqua:
    - Construtos podem estar correlacionados
- Questão:
  - √ Você deseja permitir que os fatores estejam correlacionados ou não

Rotação Oblíqua

- Permitir correlação entre fatores relaciona-se mais com nosso conceito da estrutura latente subjacente e menos com os dados
- Os eixos dimensionais não são perpendiculares, mas assimétricos pelas correlações entre os fatores

- No exemplo:
  - √ Podemos julgar que os construtos valor e liderança estejam correlacionados
  - √ O líder pode colocar um preço especial e, portanto podemos esperar que esses dois construtos sejam correlacionados negativamente
    (ao invés de independentes)

Rotação Oblimin (oblíqua):

```
> library(GPArotation)
> (brand.fa.ob <- factanal(brand.sc[, 1:9], factors = 3, rotation = "oblimin"))

Loadings:
        Factor1 Factor2 Factor3
perform           0.601
leader            0.816
latest                    1.009
fun             -0.381    0.229
serious           0.689
bargain   0.859
value     0.880
trendy   -0.267    0.128   0.538
rebuy     0.448    0.255  -0.226

Factor Correlations:
        Factor1 Factor2 Factor3
Factor1  1.0000 -0.388  0.0368
Factor2 -0.3884  1.000 -0.1091
Factor3  0.0368 -0.109  1.0000
```

√ Não há mudança substancial na interpretação dos fatores
- Loadings ligeiramente diferentes

Resultados apresentam matriz de correlações

Varimax e Oblimin – Diferenças:

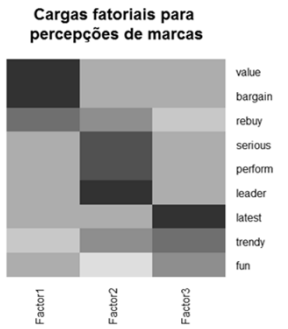
```
> # Rotação Varimax

Loadings:
        Factor1 Factor2 Factor3
perform           0.607
leader            0.810   0.106
latest  -0.163            0.981
fun             -0.398    0.205
serious           0.682
bargain  0.826           -0.122
value    0.867           -0.198
trendy  -0.356            0.586
rebuy    0.499   0.296  -0.298
```

```
> # Rotação Oblimin

Loadings:
        Factor1 Factor2 Factor3
perform           0.601
leader            0.816
latest                    1.009
fun             -0.381    0.229
serious           0.689
bargain  0.859
value    0.880
trendy  -0.267    0.128   0.538
rebuy    0.448    0.255  -0.226
```

√ Mostra separação distinta dos atributos entre os fatores
√ F1 é correlacionado com F2 (r = –0,39)
√ Decisão entre as rotações:
- Basear-se no conhecimento e domínio interpretativo, em vez da estatística

√ Mapa de calor dos loadings:

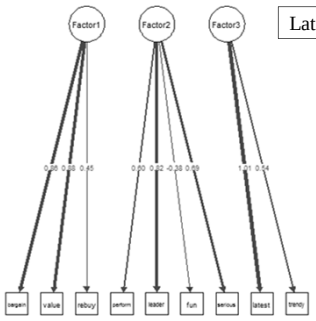
```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(brand.fa.ob$loadings, col = brewer.pal(9, "Greens"),
+           trace = "none", key = FALSE, dend = "none",
+           Colv = FALSE, cexCol = 1.2,
+           main = "\n\n\n\nCargas fatoriais para \npercepções de marcas")
```



√ Separação clara das atributos nos 3 fatores
√ *Rebuy*:
- Carrega em F1 (value) e F2(leader)
- Consumidores recompram ou pelo valor da marca ou por ela ter liderança

√ *Path diagram*:

```
> library(semPlot)
> semPaths(brand.fa.ob, what = "est", residuals = FALSE,
+     cut = 0.3, posCol = c("white", "darkgreen"),
+     negCol = c("white", "red"), edge.label.cex = 0.75, nCharNodes = 7)
```



Latentes

√ Loading +: green
√ Loading –: red

√ Ao invés de usar as 9 variáveis observadas, os dados poderiam ser representados com os 3 fatores latentes subjacentes

Variáveis observáveis

- Scores dos fatores para as marcas:
  - √ Estimativa da variável latente para cada observação

```
> # Bartlett scores
> brand.fa.ob <- factanal(brand.sc[, 1:9], factors = 3, rotation = "oblimin",
+                         scores = "Bartlett")
> brand.scores <- data.frame(brand.fa.ob$scores)    # get the factor scores
> brand.scores$brand <- brand.sc$brand              # get the matching brands
> head(brand.scores)

     Factor1    Factor2    Factor3 brand
1  1.6521364 -0.6886749  0.5256104     a
2 -1.4005333 -1.6681901 -0.6764121     a
```

  - √ Útil em modelos como os de regressão porque pode-se reduzir sua complexidade (número de dimensões)
  - √ Permite visualizar os dados em um espaço com quantidade menor de dimensões

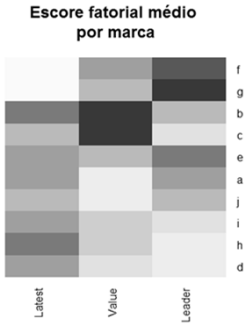
- Uso dos escores para Determinar a posição das marcas nos construtos:

```
> # Determinação da psoição da marca nos fatores
> brand.fa.mean <- aggregate(. ~ brand, data = brand.scores, mean)
> rownames(brand.fa.mean) <- brand.fa.mean[, 1]             # brand names
> brand.fa.mean <- brand.fa.mean[, -1]
> names(brand.fa.mean) <- c("Leader", "Value", "Latest")    # factor names
> brand.fa.mean

 Leader       Value      Latest
a  0.23158792 -1.06993703  0.39326652
b  0.09686823  1.51913070  0.72391174
c -0.58937138  1.45069457 -0.07690784
...
```

  - √ Média de cada marca por construto

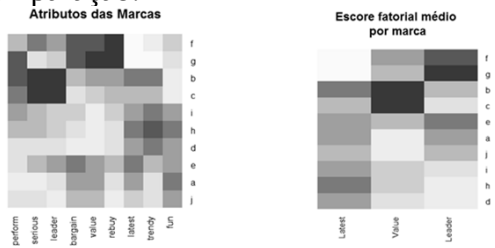
- √ Mapa de calor das médias das marcas:

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(brand.fa.ob$loadings, col = brewer.pal(9, "Greens"),
+           trace = "none", key = FALSE, dend = "none",
+           Colv = FALSE, cexCol = 1.2,
+           main = "\n\n\n\nCargas fatoriais para \npercepções de marcas")
```



  - √ Média de cada marca por construto

- Comparação:



  - √ Mapa com scores fatoriais é mais simples que a matriz completa das percepções
  - √ As similaridades entre as marcas são evidenciadas novamente
    - **f**-**g**, **b**-**c**, ...

Usos da Análise Fatorial

- Examinar a estrutura subjacente e as relações das variáveis
- Reduzir a complexidade dos dados em construtos mais simples e melhor interpretáveis

Análise Discriminante

Agrupamento e Classificação

- Agrupar:
  - √ Processo de alocar item em grupo
  - √ Não há suposições sobre o número de grupos ou sobre a estrutura dos grupos
    - Técnica mais primitiva
- Classificar:
  - √ Predição de pertinência a grupo
  - √ Número de grupos é conhecido e o objetivo é alocar novas observações a um desses grupos
  - √ Usa status conhecido para encontrar preditores, aplicando-os a uma nova observação

Conjunto de Dados

- Partição do conjunto de dados:
  - √ Conjunto de treinamento
    - Usado para desenvolver modelo de classificação
  - √ Conjunto de teste
    - Usado para determinar desempenho do modelo
  - √ Importante não avaliar desempenho com as mesmas observações usadas para desenvolver o modelo

Passos para Classificação

1. Conjunto de dados coletado, com alocações de item em grupo já conhecidas (ou atribuídas)
   - √ Observação, julgamento de especialista, procedimentos de agrupamento
2. Dados são divididos em conjunto de treinamento e teste
   - √ Treinamento: de 50% a 80% (comum: 67%)
   - √ Restante atribuído ao conjunto de teste

3. Construção do modelo de predição
   - √ Predizer alocação dos dados de treinamento tão bem quanto possível
4. Avaliação do desempenho do modelo usando os dados do conjunto de teste

Métodos de Classificação

- Há inúmero métodos de classificação:
  - √ Análise discriminante
  - √ Regressão logística
  - √ Naive Bayes Classification
  - √ Random Forest Classifiers
  - √ Método do vizinho mais próximo
  - √ Classification and Regression Trees – CART
  - √ Support Vector Machine – SVM
  - √ Método dos núcleos estimadores
  - √ Redes neurais artificiais

Análise de Agrupamento e Análise Discriminante

- Análise de Agrupamentos
  - √ Dividir os elementos da amostra (ou população) em grupos, de maneira que:
    - – Elementos de um grupo são similares entre si
    - – Elementos de grupos diferentes sejam heterogêneos em relação a essas características

- Análise discriminante:
  - √ Classificação de elementos de amostra (população)
    - Grupos são pré-definidos
  - √ Procedimento:
    - Regra de classificação

Análise Discriminante

- Caso especial de correlações canônicas
  - √ Variáveis dependentes são categóricas por natureza
- Objetivo:
  - √ Usar informações das ariáveis independentes para a separação (discriminação) mais clara possível entre os grupos

- Abordagens:
  - √ Fischer
  - √ Mahalanobis

Aplicações Potenciais

- Perfil:
  - √ Compreender como cada variável independente (X) influencia a variável dependente (Y: grupo)
  - √ Descrição, em análise de regressão
  - √ Quando os objetivos do estudo são principalmente exploratórios

√ Como os grupos são discriminados pelas variáveis subjacentes?
- Exame dos perfis de segmentos do mercado para entender como consumidores diferem com relação a variáveis demográficas e psicológicas
- Diferenças entre usuários de categoria de produto em relação ao tamanho da família, renda, educação, etc.

√ Como potenciais consumidores de marca diferem da população em geral em relação ao seu envolvimento com a mídia?

188

- Diferenciação:
  - √ Capacidade de afirmar, com certo nível de confiança, se a relação entre X e Y se deve ao acaso
  - √ Inferência, em análise de regressão
  - √ Traçados os perfis dos grupo, pode ser importante verificar se as diferenças aparentes entre eles dão de fato significativas
  - √ Exemplo:
    - Entender e controlar as variações associadas a certos processos de produção

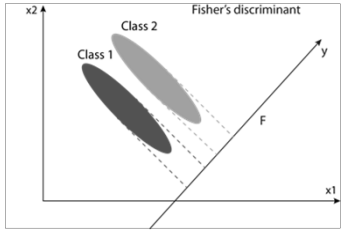
189

- Classificação:
  - √ Usar o modelo para avaliar o valor da variável dependente, com observações fora da amostra de treinamento
    - Predizer a pertinência a grupo
  - √ Predição, em análise de regressão
  - √ Exemplos:
    - *Credit scoring*
      - Traçar o perfil dos clientes de empréstimo e julgar se novos candidatos oferecem risco ao crédito
    - Marketing direto
      - Que perfil de clientes devem receber oferta de mala direta?

190

Fisher – Intuição

- Baseia-se na noção de pontuação discriminante

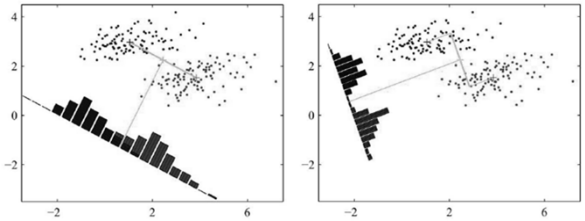


√ Encontrar combinação linear das variáveis independente que produza pontuações discriminantes maximamente diferentes

191

Função objetivo:

√ Quantifica a noção de “maximamente diferente”



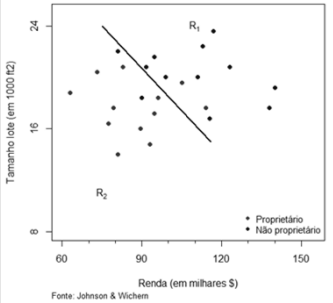
√ Função linear que melhor aloca as observações

- Eixo que descreve diferença entre centróides
- Ajusta de acordo com o padrão de covariância

192

Mahalanobis – Intuição

- Encontrar o ‘locus’ dos pontos equidistantes das médias dos 2 grupos



√ 2 variáveis explicativas:
- ‘locus’ dos pontos é uma linha

√ 3 variáveis explicativas:
- ‘locus’ dos pontos é um plano ou hiperplano

√ ‘locus’ serve para discriminar os dois grupos

193

- Medida de distância ajustada

$$D_i^2 = (\mathbf{x} - \bar{\mathbf{x}}_{(i)})' \mathbf{C}_W^{-1} (\mathbf{x} - \bar{\mathbf{x}}_{(i)}),\ i = 1, 2.$$

√ Distância ao quadrado da covariância ajustada de qualquer ponto ***x*** à média do grupo i

√ Dados seguem normal multivariada:

- Distância ajustada reflete com mais precisão a probabilidade de pertinência ao grupo do que a distância euclidiana

194

- Por definição, ‘locus’ dos pontos descritos por Malahanobis é ortogonal ao eixo da função discriminante proposta por Fisher

195

Análise Discriminante – Abordagens

- São complementares:
  - √ Fisher:
    - – Reduz os dados em uma única dimensão de modo a maximizar a separação entre grupos
  - √ Mahalanobis:
    - – Determina linha divisória (ou plano) que separa mais precisamente os dois grupos
    - – Ortogonal à dimensão discriminante

Regras de Alocação e Classificação

- Em geral, são desenvolvidas a partir de amostras de treinamento:
  - √ Examinadas diferenças das medidas características de objetos selecionados
  - √ Todos os resultados amostrais possíveis são dividido em duas regiões (R_1 e R_2)
    - – Se uma nova observação pertencer à região R_1 ela é alocada à população π_1.
    - – Se uma nova observação pertencer à região R_2 ela é alocada à população π_2.

Problema da Classificação

- Como saber se algumas observações pertencem a uma particular população?
  - √ Incerteza na classificação

Paradoxos da Classificação

- Informação incompleta sobre desempenho futuro:
  - √ Classificação de candidato como capaz de concluir ou não um mestrado
- Informação perfeita exige destruição objeto:
  - √ Classificação de itens como bons ou defeituosos
- Informação cara ou indisponível:
  - √ Problemas médicos que podem ser identificados conclusivamente apenas com procedimentos caros

Erros de Classificação

- Caso médico:
  - √ Em geral, deseja-se diagnosticar um mal a partir de sintomas externos facilmente observáveis
- Erro de classificação:
  - √ A distinção entre as características medidas das duas populações pode não ser clara.

Critérios para Classificação

- Bom procedimento de classificação:
  - √ Poucos erros de classificação
- Probabilidades a priori deveriam integrar regra ótima:
  - √ Classe (ou população) com verossimilhança de ocorrência maior que outra
  - √ Classe é relativamente maior que outra
  - √ Ex.:
    - – Há muito mais empresas solventes que insolventes

- Outro aspecto a considerar:
  - √ Custo associado ao erro de classificação
  - √ Ex.:
    - – Classificar um objeto π_1 como π_2 é mais sério que classificar um objeto π_2 como π_1.

Critérios para Classificação

- Bom procedimento de classificação:
  - √ Poucos erros de classificação
- Probabilidades a priori deveriam integrar regra ótima:
  - √ Classe (ou população) com verossimilhança de ocorrência maior que outra
  - √ Classe é relativamente maior que outra
  - √ Ex.:
    - – Há muito mais empresas solventes que insolventes

- Outro aspecto a considerar:
  - √ Custo associado ao erro de classificação
  - √ Ex.:
    - Classificar um objeto π_1 como π_2 é mais sério que classificar um objeto π_2 como π_1.

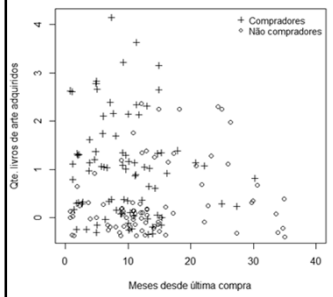
Exemplo

- Clube de livro *'Books by Mail'*:
  - √ Oferta de livro de arte
    - Correspondência de teste enviada para 1.000 clientes escolhidos aleatoriamente
    - 83, responderam à oferta
  - √ Informações de compras passadas:
    - X_1: tempo desde a última compra, meses
    - X_2: quantidade de livros de artes adquiridos
  - √ Objetivo:
    - Discriminar compradores e não compradores
  - √ Dados: *BOOKS_1*.txt e *BOOKS_2.txt*

- Importação e tratamento dos dados:

```
> # Carregamento e tratamento conjunto de dados
>
> books <- read.table("BOOKS_1.txt")
> books <- books[-1]
> colnames(books) <- c("tempo", "livros", "compra")
> books$compra <- factor(books$compra, labels=c("N","Y"))
> levels(books$compra) <- c("N", "Y")
> books$tmpc <- cut(books$tempo, breaks = c(0, seq(2.5, 37.5, 5)),
+   labels = c(1, seq(5, 35, 5)))
> medias.livros <- aggregate(livros ~ tmpc, data = books, mean)
> head(books)
  tempo livros compra tmpc
1    24      0      N   25
2    16      0      N   15
3    15      0      N   15
4    22      0      N   20
5    15      0      Y   15
6     6      2      N    5
```


- Análise descritiva do conjunto de dados:
  - √ 83 compradores e 83 não compradores (ao acaso)



- √ Pontos plotados com perturbação
- √ Distribuição dos compradores deslocada em relação aos não compradores
  - − A sobreposição é substancial

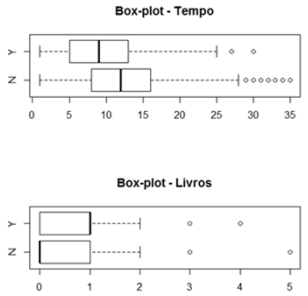
Média das variáveis por grupo:

```
> # Carregamento e tratamento conjunto de dados
> setNames(aggregate(. ~ compra, data = books, mean),
+   c("Compra", "Tempo", "Qte. Livros"))
  Compra     Tempo Qte. Livros
1      N 12.731734   0.3336968
2      Y  9.409639   1.0000000
```

√ Compradores tendem a apresentar;

- Intervalo médio mais curto desde a última compra
- Número médio mais alto de livros de arte adquiridos
  - Maior interesse pela categoria

Diferenças univariadas entre os grupos:



√ Há diferenças entre as médias individuais
√ As diferenças entre as médias conjuntas (centróides) são significativas?
√ Como visualizar

Scatter plot matrix das variáveis:

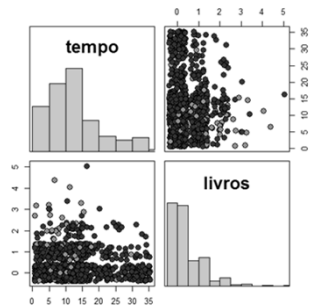
√ Função customizada

```
> # Matrix plot variáveis livros e tempo
> # Customização plot
> panel.hist = function(x, ...) {
+   usr <- par("usr"); on.exit(par(usr))
+   par(usr = c(usr[1:2], 0, 1.5) )
+   h <- hist(x, plot = FALSE)
+   breaks <- h$breaks; nB <- length(breaks)
+   y <- h$counts; y <- y/max(y)
+   rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
+ }
```

√ Gráfico de todos os pontos:

```
> # Gráfico tempo e livros - todos os pontos
> with(books, pairs(jitter(cbind(tempo, livros), ruido), cex = 1.5,
pch = 21,
+       bg = c("red","green3")[unclass(compra)], diag.panel =
panel.hist,
+       cex.labels = 2, font.labels = 2)
+ )
```


Matrix plot das variáveis:

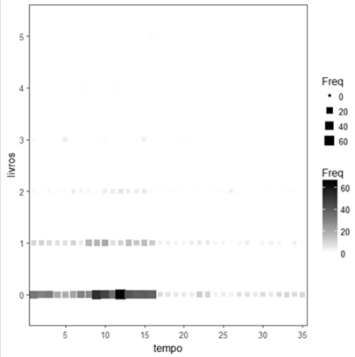


√ Difícil visualização dos grupos
  - Não compradores dominam
  - Há muito empates
√ Variáveis são discretas
√ Distribuições marginais

- Alternativa – *Fluctuation plot*:
  - √ Tamanho das células por frequência:

```
> # Alternativa 1 - Fluctiation plot
> theme_nogrid <- function (base_size = 12, base_family = "") {
+   theme_bw(base_size = base_size, base_family = base_family)
%+replace%
+     theme(panel.grid = element_blank())
+ }
> contagens.df <- with(books, as.data.frame(table(tempo, livros)))
> ggplot(contagens.df, aes(tempo, livros)) +
+   geom_point(aes(size = Freq, color = Freq, stat = "mean",
+ position = "identity"), shape = 15) +
+   scale_size_continuous(range = c(1,5)) +
+   scale_color_gradient(low = "white", high = "black") +
+   scale_x_discrete(breaks = seq(0, 35, 5)) +
+   theme_nogrid()
```

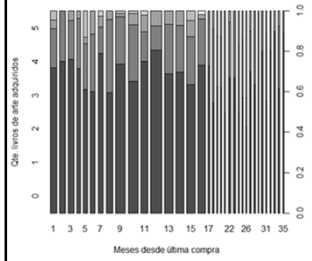

- Alternativa – *Fluctuation plot*:



  - √ Muitos valores zero para tempos abaixo de 16 meses

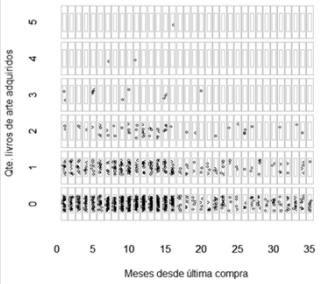
- Alternativa – *Spine plot*:
  - √ Larguras e alturas ponderadas por frequência

```
> # Alternativa 2 - spine plot
> with(books, spineplot(factor(livros)~ factor(tempo),
+ xlab = "Meses desde última compra",
+ ylab = "Qte. livros de arte adquiridos"))
> with(books, spineplot(factor(livros)~ factor(tempo),
+ xlab = "Meses desde última compra",
+ ylab = "Qte. livros de arte adquiridos"))
```



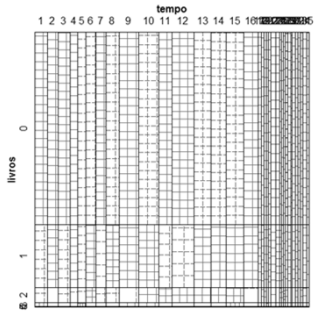
  - √ Valores concentrados abaixo de 16 meses

- Alternativa – *Scatter plot* modificado:
  - √ Visualização dos valores observados em cada par de valores da variáveis



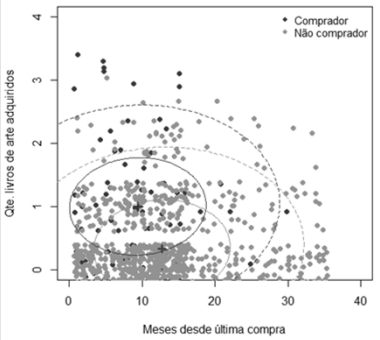
  - √ Valores concentrados abaixo de 16 meses e 1 livro

- Alternativa – *Sive plot*:
  - √ Visualização com contagem por célula



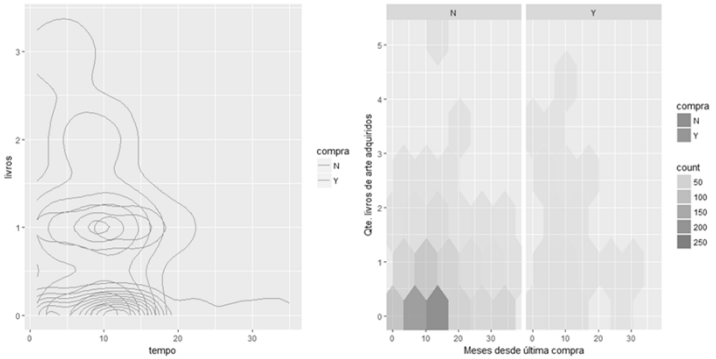
- √ Concentração nos valores de zero livros comprados

- Scatter plot por grupos:
  - √ Há diferenças entre os centróides?

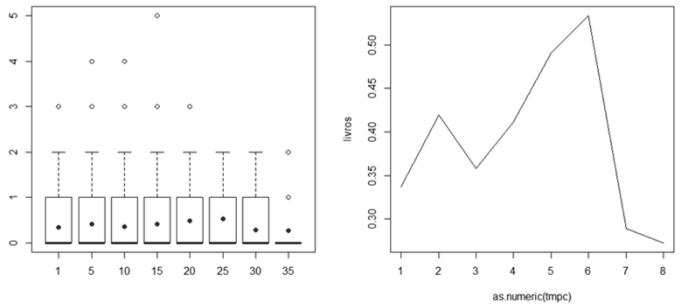



```
> # centróides
  compra     Tempo Qte. Livros
1      N 12.731734    0.3336968
2      Y  9.409639    1.0000000

> # Matriz covariâncias - grupo
> cov.lista
[[1]]
             tempo    livros
tempo  65.7270814 0.2391806
livros  0.2391806 0.3688742

[[2]]
             tempo     livros
tempo  35.4155157 -0.6707317
livros -0.6707317  1.1219512

> # Matriz covariâncias combinada
> sigma.pol
             tempo    livros
tempo  63.2365519 0.1644183
livros  0.1644183 0.4307503
```

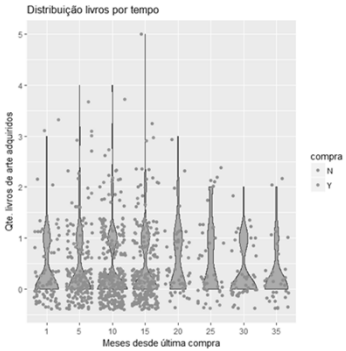

- *Scatter plot* por grupos – Estimado:



- Box-plot: livros por intervalo entre compras:
  - √ Tempo categorizado



- *Violin plot*:
  - √ Visualização da distribuição dos dados e de sua densidade.



  - √ Semelhante box plot
  - √ Apresenta densidade condicional
  - √ Cuidado com o uso:
    - No caso, as variáveis são discretas

- Centróides:

```
> centroide <- aggregate(cbind(tempo, livros) ~ compra, data = books, mean)
> centroide
  compra     tempo    livros
1      N 12.731734 0.3336968
2      Y  9.409639 1.0000000
```

- Distância entre os centróides:

$$\bar{\mathbf{x}}_{(2)} - \bar{\mathbf{x}}_{(1)} = \begin{bmatrix} 9,40 \\ 1,00 \end{bmatrix} - \begin{bmatrix} 12,70 \\ 0,33 \end{bmatrix} = \begin{bmatrix} -3,30 \\ 0,67 \end{bmatrix}.$$

- √ Matrizes das somas de quadrados – *within*:

```
> cov(books[books$compra == "N", 1:2])*(1000 - 83 -1) # SS within N
           tempo   livros
tempo  60206.0065 219.0894
livros   219.0894 337.8888

> cov(books[books$compra == "Y", 1:2])*(83-1)# SS within Y
           tempo livros
tempo  2904.072    -55
livros  -55.000     92
```

- √ Matriz de covariâncias combinada – *between*:

```
> books.aov <- manova(cbind(tempo, livros) ~ compra, data = books)
> estVar(books.aov)# Matriz de covariâncias combinada (entre grupos)
           tempo    livros
tempo  63.2365519 0.1644183
livros  0.1644183 0.4307503
```

- √ Inversa da matriz de covariâncias combinada:

```
> solve(estVar(books.aov))     # Inversa da matrz de covariâncias combinada
             tempo       livros
tempo   0.015829349 -0.006042095
livros -0.006042095  2.323837045
```


- Função discriminante:

```
> # Função Discriminante de Fisher
> library(MASS) # comando lda
> ajuste.df <- lda(books[, 1:2], books$compra, data = books)
> ajuste.df
Call:
lda(books[, 1:2], books$compra, data = books)

Prior probabilities of groups:
    N     Y
0.917 0.083

Group means:
      tempo    livros
N 12.731734 0.3336968
Y  9.409639 1.0000000

Coefficients of linear discriminants:
                LD1
tempo  -0.05098078
livros  1.41242601
```

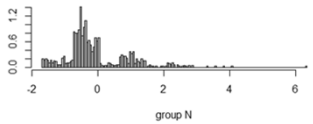
Centróides

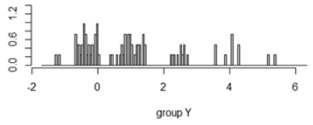
Proporcional a [–0,056; 1,577]

  - √ R padroniza variável discriminante:
    - Média zero e desvio-padrão 1

Escores discriminantes dos grupos:

```
> predicao <- predict(ajuste.df, books[, 1:2])
> GrupoPrevisto <- predicao$class
> ldahist(data = predicao$x, g = books$compra, h = 0.05)
```





- Compradores:
  - √ Em média mais positivos
- Não compradores:
  - √ Em média mais negativos

227

Referências

Bibliografia Recomendada

- ALBERT, J.; RIZZO, M. *R by Example*. Springer, 2012.
- CHAPMAN, C.; FEIT, E. M. *R for marketing research and analytics*. Springer, 2015.
- KLEIBER, C.; ZEILEIS, A. *Applied econometrics with R*. Springer, 2008.
- DALGAARD, P. *Introductory statistics with R*. Springer, 2008.

229