Introdução à Análise e Modelagem de Dados Multivariados com o R

Lupércio França Bessegato
Dep. de Estatística/UFJF

Roteiro Geral

1. Fundamentos gráficos em R
2. Visualização de dados multivariados
   - Componentes principais, análise fatorial, escalonamento multidimensional; análise de agrupamentos
3. Técnicas de interdependência
   - Discriminação e classificação; análise discriminante linear; modelos logit.
4. Referências

Visualização e Descrição de Dados

Resumos Numéricos

- Principais medidas resumo para exploração de conjunto de dados:
  - √ Medidas de posição:
    - Média
    - Mediana
  - √ Medidas de dispersão:
    - Desvio-padrão (variância)
    - Distância interquartílica

Funções Mais Usadas

Símbolo	Função
`sum(x)`	Soma dos elementos de *x*
`prod(x)`	Produtório dos elementos de *x*
`max(x)`	Elemento máximo de *x*
`min(x)`	Elemento mínimo de *x*
`range(x)`	Elementos máximo e mínimo de *x*
`length(x)`	Quantidade de elementos do vetor *x*
`mean(x)`	Média dos elementos de *x*
`median(x)`	Mediana dos elementos de *x*
`var(x)`	Variância dos elementos de *x*
`sd(x)`	Desvio padrão dos elementos de *x*
`quantile(x,p)`	Quantil dos elementos de *x*, correspondente a *p*
`cor(x, y)`	Correlação entre os elementos de *x* e *y*

Exemplo

- Conjunto de dados de turma de alunos com as variáveis:
  - √ Sexo
  - √ Peso
  - √ altura

```
            # Carregando e conhecendo o banco

Dados <- read.csv2(file = "turma.csv")
head(dados)
dim(dados)            # tamanho do conjunto de dados
attach(dados)
head(Altura)
head(Peso)
head(Sexo)
is.factor(Sexo)       # verifica se categórica está como fator
```


Função length(x)

- Calcula quantidade de elementos de vetor
- Verifica a quantidade de variáveis:

```
length(Peso)          # Calcula o tamanho da amostra

# usada no conjunto de dados
length(dados)         # Informa quantidade de variáveis
```


Valores Extremos

- Funções
  - √ `min(x)` : determina o menor valor da variável
  - √ `max(x)` : determina o maior valor da variável
  - √ `range(x)` : determina o menor e o maior valor da variável

```
min(Peso)             # Menor peso observado
max(Peso)             # Maior peso observado
range(Peso)           # Menor e maior peso observado (vetor)
```

Outras Funções

- Soma e produto:
  - √ `sum(x)` : soma todos os elementos de x
  - √ `prod(x)` : multiplica todos os elementos de x.

```
sum(Peso)               # soma todos os pesos observados
prod(Peso)              # multiplica todos os pesos observados
sum(Peso)/length(Peso)  # cálculo do peso médio
```

- Média
  - √ `mean(x)` : médias dos elementos de x

```
mean(Peso)              # média dos pesos observados
mean(Altura)            # média das alturas observada
```


Função Aplicada a Grupos da Variável

- Determinação da média de alguns valores da variável.
  - √ Aplicando diretamente o comando `mean`

```
mean(Peso[Sexo=="F"])   # média dos pesos das alunas
mean(Peso[Sexo=="M"])   # média dos pesos das alunas
```

  - √ Comando `tapply` e `aggregate`
    - Aplica função a cada grupo de valores dado por uma combinação única dos níveis de certos fatores.

```
      # média da variável Peso por Sexo
tapply(Peso, Sexo, FUN = mean)
      # média de todas as variáveis por Sexo
aggregate(dados[, -1], list(Sexo), mean)
```


Mediana

- `median(x)` : calcula mediana da variável observada.

```
median(Peso)              # mediana dos pesos de todos os alunos
median(Altura)            # mediana das alturas de todos os alunos
median(Peso[Sexo=="M"])   # mediana dos pesos dos alunos
```


Dispersão

- √ `var(x)` : variância dos elementos de x
- √ `sd(x)` : desvio padrão dos elementos de x.

```
var(Peso)              # variância do peso de todos os alunos
sd(Peso)               # desvio padrão do peso de todos os alunos
var(Peso[Sexo=="F"])         # variância do peso das alunas
sd(Altura[Sexo=="M"])        # desvio padrão da altura dos alunos
```

- √ Criação de função para coeficiente de variação:

```
            # Coeficiente de variação - criação de função
cv <- function(x) sd(x)/mean(x)
cv(Peso)
```

Quantis

- `quantile(x, p)`: determina quantil, onde x é a variável observada e *p* é uma probabilidade.

```
quantile(Peso, 0.7)                 # Percentil 70 dos pesos
quantile(Peso, c(0.25, 0.75))       # 1° e 3° quartis dos pesos
quantile(Peso[Sexo=="F"], 0.7)      # Percentil 70 das alunas
quantile(Peso, 0.5)                 # mediana de todos os pesos
```


Correlação

- Relação linear entre duas variáveis quantitativas
  - √ `corr(x, y)`: coeficiente de correlação linear entre as variáveis *x* e *y*.

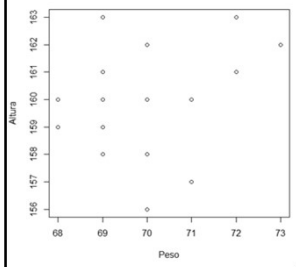
```
cor(Peso, Altura)           # correlação linear entre peso e altura
cor.test(Peso, Altura)      # teste de significância da correlação
```

- Opções do comando:
  - √ `cor(x,y,method='pearson')`: default
  - √ `cor(x,y,method='spearman')`
  - √ `cor.test(x,y,method='pearson')`: default
  - √ `cor.test(x,y,method='spearman')`

Gráfico de dispersão

√ `plot(x,y)`: gráfico da relação das variáveis quantitativas *x* e *y*.

```
plot(Peso, Altura)   # gráfico de dispersão entre Peso e Altura
```



Resumo de dos Dados

- Variáveis quantitativas:
  - √ Resumo de 5 números e média
    - `summary(x)`: fornece o mínimo, 1º quartil. Mediana, 3º quartil, máximo e média dos elementos de `x`.
- Variáveis categóricas:
  - √ Tabela de frequências

```
summary(Peso)       # resumos da variável Peso
summary(Altura)     # resumos da variável Altura
```

Tabelas

- Resumo da frequência dos níveis de variável categórico (ou variável discreta).
- `table(x)`:

```
table(Sexo)                 # tabela de contingência de Sexo
prop.table(table(Sexo))     # tabela de frequência relativa
```


Tabela de Frequência – Variável Contínua

- Não há comando específico no R. É necessário construí-la:
  - √ Exemplo com o conjunto de dados `faithful`.

```
duracao <- faithful$eruptions
range(duracao)
# sequencia para intervalo dos dados (aproximado)
breaks <- seq(1.5, 5.5, by=0.5)
# aloca elementos em sub-intervalos de tamanho 0.5
duracao.cut <- cut(duracao, breaks, right=FALSE)
# calcula a frequencia de erupções em cada sub-intervalo
duracao.freq <- table(duracao.cut)
# tabela com os resultados
cbind(duracao.freq)
```


Histograma

- Visualizando a variável `duracao`:

```
hist(duracao)
hist(duracao, label = T)        # histograma com frequências

duracao.hist <- hist(duracao)   # cria objeto com o histograma
str(duracao.hist)               # estrutura do objeto histograma
# limites dos sub-intervalos do histograma
duracao.hist$breaks
# frequência de valores em cada ub-intervalo
duracao.hist$counts)
```


Geração de um Gráfico Aleatório

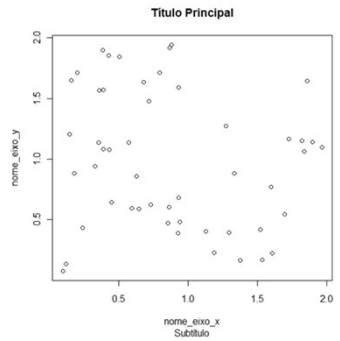
- Geração de 50 pontos ao acaso entre 0 e 2:

```
x <- runif(50, 0, 2)
y <- runif(50, 0, 2)
```

- Gráfico dos 50 pontos com título, subtítulo, rótulos eixos *x* e *y*:

```
plot(x, y, main = "Título Principal", sub = "Subtítulo",
       xlab = "nome_eixo_x", ylab = "nome_eixo_y")
```

Gráfico Gerado



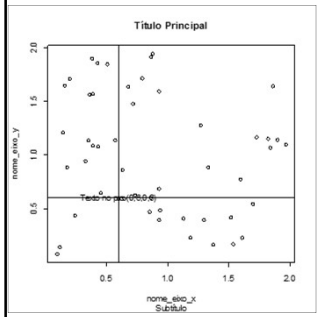
√ O gráfico de cada um será diferente

– Se rodar de novo o resultado também será outro

Adição de Dados

- Adicionando texto e linhas ao gráfico

```
text(0.6, 0.6, "texto no pto (0.6,0.6)")
# linhas pelo ponto (0.6, 0.6)
abline(h = 0.6, v = 0.6)
```



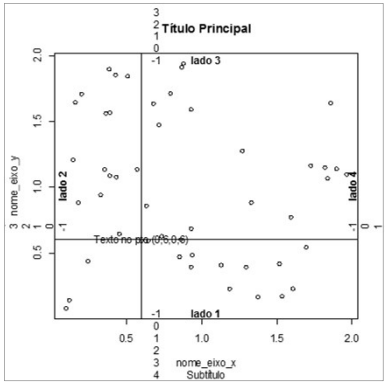
- `abline(a,b)` plota a reta $y=a+bx$

Coordenadas das Margens

- Coordenadas das margens através função `mtext`

```
       # coordenadas da margem
mtext(-1:4,side=1,at=0.7,line=-1:4)# coordenadas da margem
       # loop para as coordenadas das margens
for(lado in 1:4) mtext(-1:4, side = lado, at = 0.7, line = -1:4)
       # lado das margens
mtext(paste("lado", 1:4), side = 1:4, line = -1, font = 2)
```


Gráfico c/ Coordenadas Margens



- Layout de um gráfico padrão

Construindo um Gráfico por Partes

- Permite controle fino de cada elemento do gráfico
  - √ Desenha-se primeiro o gráfico sem os elemento

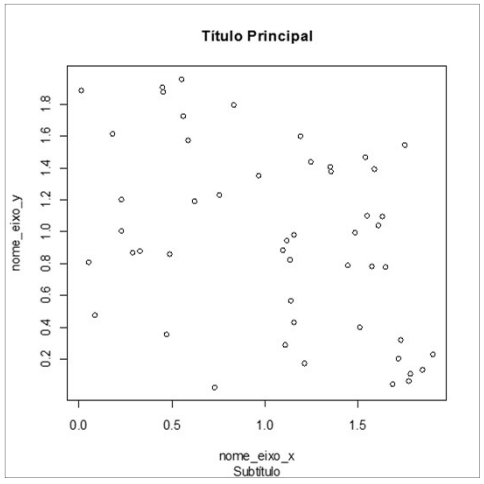
```
plot(x,y,type="n",xlab="",ylab="",axes=F) # plota-se nada!
```

  - √ Os elementos serão adicionados subsequentemente

Montagem do Gráfico

- O gráfico pode ser montado executando cada comando por vez
  - √ Verifique o que acontecerá

```
points(x, y)                        # plota os pontos do gráfico
axis(1)                             # plota o eixo x
axis(2, at = seq(0.2, 1.8, 0.2))    # plota o eixo y
box()                               # caixa do gráfico
# Título, sub-título, nomes dos eixos
title(main = "Título Principal", sub = "Subtítulo",
      xlab = "nome_eixo_x", ylab = "nome_eixo_y")
```

Argumentos para Gráficos

- Argumentos mais usados

Table 2.1. A selective list of arguments to `par()`.

Argument	Description
`axes`	should axes be drawn?
`bg`	background color
`cex`	size of a point or symbol
`col`	color
`las`	orientation of axis labels
`lty, lwd`	line type and line width
`main, sub`	title and subtitle
`mar`	size of margins
`mfcol, mfrow`	array defining layout for several graphs on a plot
`pch`	plotting symbol
`type`	types (see text)
`xlab, ylab`	axis labels
`xlim, ylim`	axis ranges
`xlog, ylog, log`	logarithmic scales

Histogramas

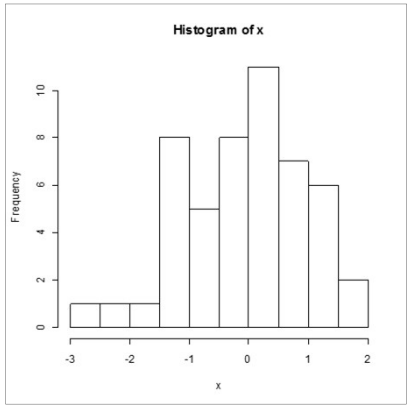
Construção de Histograma

- Geração de uma amostra aleatória com distribuição de freqüências com simetria:

```
x <- rnorm(50)
```

- Construção do histograma (default)

```
hist(x)
```

√ Repetir com outras amostras e verifique

Exemplo

- Levantamento de revistas econômicas.
  - √ Amostra aleatória com 180 observações
  - √ 10 variáveis (quantitativas e categóricas)
  - √ Período: 2000
  - √ Dados: *Journals*{*AER*} ou *Journals.csv*

√ Variáveis:

- title: título do periódico
- publisher: nome do editor (fator com 52 níveis)
- society: periódico é publicado por uma sociedade acadêmica? ('no' = não, 'yes' = sim
- price: preço da assinatura da biblioteca
- pages: número de páginas
- charpp: caracteres por página
- citations: número total de citações
- foundingyear: ano de fundação do jornal
- subs: número de assinaturas da biblioteca
- field: descrição do campo da Economia (fator, com 24 níveis).

• Importação dos dados – pacote AER:

```
> # carregamento direto do pacote
> data("Journals", package = "AER")
> help(Journals, package = "AER")
> revistas <- Journals
> str(revistas)
'data.frame':   180 obs. of  10 variables:
 $ title       : chr  "Asian-Pacific Economic Literature" "South African Journal of
Economic History" "Computational Economics" "MOCT-MOST Economic Policy in
Transitional Economics" ...
 $ publisher   : Factor w/ 52 levels "ANU Press","Academic Press",..: 11 45 28 28
18 18 13 18 28 11 ...
 $ society     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ price       : int  123 20 443 276 295 344 90 242 226 262 ...
 $ pages       : int  440 309 567 520 791 609 602 665 243 386 ...
 $ charpp      : int  3822 1782 2924 3234 3024 2967 3185 2688 3010 2501 ...
 $ citations   : int  21 22 22 22 24 24 24 27 28 30 ...
 $ foundingyear: int  1986 1986 1987 1991 1972 1994 1995 1968 1987 1949 ...
 $ subs        : int  14 59 17 2 96 15 14 202 46 46 ...
 $ field       : Factor w/ 24 levels "General","Economic History",..: 1 2 3 4 5
> dim(revistas)
[1] 180  10
```


• Importação pelo arquivo Journals.csv:

```
> # carregamento do arquivo csv
> revistas <- read.csv("Journals.csv")
> str(revistas)
'data.frame':   180 obs. of  11 variables:
 $ X           : Factor w/ 180 levels "AE","AEJ","AEL",..: 10 171 14 140 130 136 32
166 45 144 ...
 $ title       : Factor w/ 180 levels "Agricultural Economics",..: 8 174 18 144 129
137 47 168 43 143 ...
 $ publisher   : Factor w/ 52 levels "Academic Press",..: 11 45 28 28 18 18 13 18
28 11 ...
 $ society     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ price       : int  123 20 443 276 295 344 90 242 226 262 ...
 $ pages       : int  440 309 567 520 791 609 602 665 243 386 ...
 $ charpp      : int  3822 1782 2924 3234 3024 2967 3185 2688 3010 2501 ...
 $ citations   : int  21 22 22 22 24 24 24 27 28 30 ...
 $ foundingyear: int  1986 1986 1987 1991 1972 1994 1995 1968 1987 1949 ...
 $ subs        : int  14 59 17 2 96 15 14 202 46 46 ...
 $ field       : Factor w/ 24 levels "Agricultural Economics",..: 10 8 22 2 14 1
> dim(revistas)
[1] 180  11
```


• Conhecendo o conjunto de dados

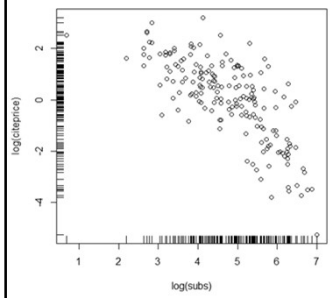
```
> # carregamento direto do pacote
> data("Journals", package = "AER")
> help(Journals, package = "AER")
> revistas <- Journals
> names(revistas)
 [1] "title"        "publisher"    "society"      "price"        "pages"
 [6] "charpp"       "citations"    "foundingyear" "subs"         "field"
> head(revistas)
                                                title
APEL                        Asian-Pacific Economic Literature
SAJoEH             South African Journal of Economic History
                  publisher society price pages charpp citations foundingyear
APEL                Blackwell      no   123   440   3822        21         1986
SAJoEH So Afr ec history assn      no    20   309   1782        22         1986
       subs             field
APEL     14           General
SAJoEH   59  Economic History
```

- Preparação do banco
  - √ Preço unitário por citação

```
> revistas$citeprice <- revistas$price/revistas$citations
> # Anexando o conjunto de dados para trabalho
> attach(revistas)
```


- *Plot* com distribuições marginais:

```
> plot(log(subs), log(citeprice))
> rug(log(subs))
> rug(log(citeprice), side = 2)
```

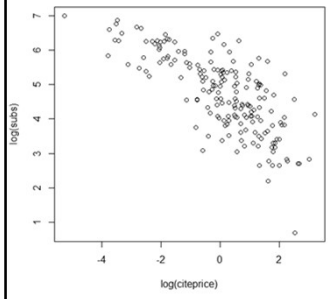


- √ Comando `rug`: marcas para cada dado

- *Plot* – comando alternativo
  - √ Sem o arquivo estar anexado

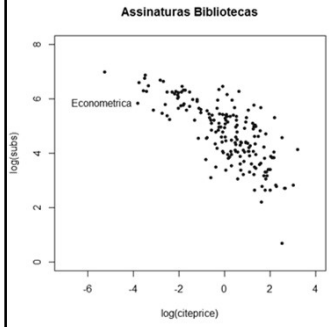
```
> detach(revistas)
> plot(log(subs) ~ log(citeprice), data = revistas)
```

- √ Formato Y ~ X



- *Plot* – parâmetros gráficos:

```
> plot(log(subs) ~ log(citeprice), data = revistas, pch = 20, col = "blue",
+ ylim = c(0, 8), xlim = c(-7, 4), main = "Assinaturas Bibliotecas")
> text(-3.798, 5.846, "Econometrica", pos = 2)
```

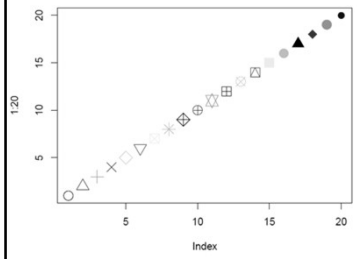


- √ Comando `text`
- √ Parâmetros:
  - `Pch, col, xlim, ylim, main`

Exportação de Gráficos

```
> pdf("meuArquivo.pdf", height = 5, width = 6)
> plot(1:20, pch = 1:20, col = 1:20, cex = 2)
> dev.off()
windows
      2
```

√ Gráfico exportado para o pdf

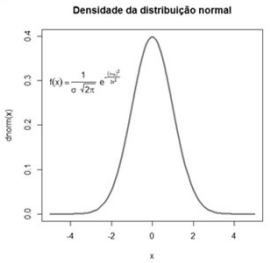


√ Formatos de caracteres
√ Cores:

Anotação Matemática

- Inserção de símbolos matemáticos:

```
> curve(dnorm, from = -5, to = 5, col = "slategray", lwd = 3,
+ main = "Densidade da distribuição normal")
> text(-5, 0.3, expression(f(x) == frac(1, sigma ~~ sqrt(2*pi)) ~~
+ e^{-frac((x - mu)^2, 2*sigma^2)}), adj = 0)
```



Sugestão

- Experimentar o comando:
  √ `demo("plotmath")`

Exemplo

- Experimento sobre influência de dieta no crescimento de pintos
  √ Amostra aleatória com 578 observações
  √ 4 variáveis (quantitativas e categóricas)
  √ Fonte: https://davetang.org/muse/2013/05/22/using-aggregate-and-apply-in-r/
  √ Dados: *ChickWeight*{*datasets*}

√ Variáveis:

- Time: número de dias entre o nascimento e a medição
- Chick: fator de identificação do pinto, com 50 níveis (18< ...)
- Diet: tipo de dieta recebida pelo pinto (níveis de 1 a 4)

• Importação dos dados:

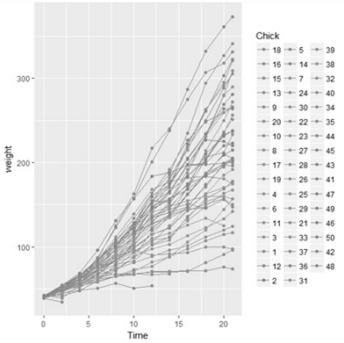
```
> # carregamento dos dados
> dados <- ChickWeight
> help(ChickWeight)
> str(dados)
Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':      578
obs. of  4 variables:
 $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
 $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
 $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15
...
 $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...

> head(dados)
  weight Time Chick Diet
1     42    0     1    1
2     51    2     1    1
3     59    4     1    1
4     64    6     1    1
5     76    8     1    1
6     93   10     1    1
```


• Explorando as variáveis:

```
> # dimensão do conjunto de dados
> dim(dados)
[1] 578   4
> # quantidade de pintos
> unique(dados$Chick)
 [1] 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
50 Levels: 18 < 16 < 15 < 13 < 9 < 20 < 10 < 8 < 17 < 19 < 4 < 6 < 11 < ... < 48
> length(unique(dados$Chick))
[1] 50
> #quantidade de dietas
> unique(dados$Diet)
[1] 1 2 3 4
Levels: 1 2 3 4
> # quantidade de instantes de tempo
> unique(dados$Time)
[1]  0  2  4  6  8 10 12 14 16 18 20 21
```


• Visualização gráfica dos crescimentos:

```
> library(ggplot2)
> ggplot(data = dados, aes(x = Time, y = weight, group = Chick, colour=Chick)) +
+         geom_line() +
+         geom_point()
```



√ Crescimento individual do peso dos pintos
√ Qual o efeito do tipo da dieta no crescimento?

- Estatísticas descritivas:
  - √ Uso do comando `aggregate`.

```
> # peso médio por dieta
> aggregate(dados$weight, list(diet = dados$Diet), mean)
  diet        x
1    1 102.6455
2    2 122.6167
3    3 142.9500
4    4 135.2627
```

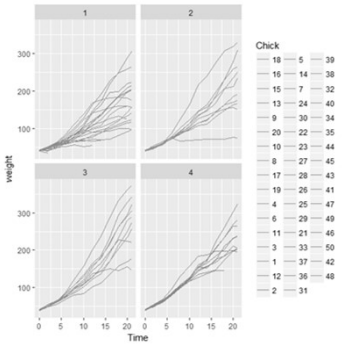
```
> # média por instante de medição
> aggregate(dados$weight, list(time = dados$Time), mean)
 time         x
1     0  41.06000
2     2  49.22000
3     4  59.95918
4     6  74.30612
5     8  91.24490
6    10 107.83673
7    12 129.24490
8    14 143.81250
9    16 168.08511
10   18 190.19149
11   20 209.71739
12   21 218.68889
```


- √ Agregação por várias variáveis:

```
> # médias agregadas por time e diet
> head(aggregate(dados$weight,
+                list(time = dados$Time, diet = dados$Diet),
+                mean
+               )
+     )
  time diet        x
1    0    1 41.40000
2    2    1 47.25000
3    4    1 56.47368
4    6    1 66.78947
> tail(aggregate(dados$weight,
+                list(time = dados$Time, diet = dados$Diet),
+                mean
+               )
+     )
   time diet        x
45   16    4 182.0000
46   18    4 202.9000
47   20    4 233.8889
48   21    4 238.5556
```


- Visualização dos crescimentos por dieta:

```
> ggplot(dados) + geom_line(aes(x = Time, y = weight, colour = Chick)) +
+              facet_wrap(~Diet) +
+              guides(col = guide_legend(ncol=3))
```



- √ Qual o efeito do tipo da dieta no crescimento?
  - Tendência
  - Dispersão

Análise Exploratória de Dados

Exemplo

- Current Population Survey, Maio/85.
  - √ Amostra aleatória com 534 observações
  - √ 11 variáveis (quantitativas e categóricas)
  - √ Dados: *CPS1985{AER}* ou *CPS1985.csv*

- √ Variáveis:
  - wage: salário, em US$ por hora
  - education: anos de escolaridade
  - experience: anos de experiência profissional potencial (age – education – 6).
  - age: idade, em anos
  - ethnicity: etnia. (cauc, hispanic, other)
  - region: mora no Sul? (south, other)
  - gender: sexo. (male, female).
  - occupation: ocupação. (worker, technical, services, office, sales, management.
  - sector: setor de ocupação. (manufacturing, construction, other).
  - union: trabalho sindicalizado? (no, yes).
  - married: É casado? (no, yes).

- Importação dos dados – pacote AER:

```
> # carregamento direto do pacote
> data("CPS1985", package = "AER")
> cps <- CPS1985
> head(cps)
 wage education experience age ethnicity region gender occupation
1     5.10         8         21  35  hispanic  other female     worker
1100  4.95         9         42  57      cauc  other female     worker
2     6.67        12          1  19      cauc  other   male     worker
           sector union married
1    manufacturing    no     yes
1100 manufacturing    no     yes
2    manufacturing    no      no
> str(CPS1985)
'data.frame':   534 obs. of  11 variables:
 $ wage      : num  5.1 4.95 6.67 4 7.5 ...
 $ education : num  8 9 12 12 12 13 10 12 16 12 ...
 $ experience: num  21 42 1 4 17 9 27 9 11 9 ...
 $ age       : num  35 57 19 22 35 28 43 27 33 27 ...
 $ ethnicity : Factor w/ 3 levels "cauc","hispanic",..: 2 1 1 1 1 1 1 1 1 1 ...
 $ region    : Factor w/ 2 levels "south","other": 2 2 2 2 2 2 1 2 2 2 ...
 $ gender    : Factor w/ 2 levels "male","female": 2 2 1 1 1 1 1 1 1 1 ...
 $ occupation: Factor w/ 6 levels "worker","technical",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ sector    : Factor w/ 3 levels "manufacturing",..: 1 1 1 3 3 3 3 3 1 3 ...
 $ union     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 1 1 ...
 $ married   : Factor w/ 2 levels "no","yes": 2 2 1 1 2 1 1 1 2 1 ...
```


- Importação pelo arquivo CPS1985.csv:

```
> # carregamento do arquivo csv
cps <- read.csv("CPS1985.csv")
> head(cps)
 wage education experience age ethnicity region gender occupation
1     5.10         8         21  35  hispanic  other female     worker
1100  4.95         9         42  57      cauc  other female     worker
2     6.67        12          1  19      cauc  other   male     worker
           sector union married
1    manufacturing    no     yes
1100 manufacturing    no     yes
2    manufacturing    no      no
> str(CPS1985)
'data.frame':   534 obs. of  11 variables:
 $ wage      : num  5.1 4.95 6.67 4 7.5 ...
 $ education : num  8 9 12 12 12 13 10 12 16 12 ...
 $ experience: num  21 42 1 4 17 9 27 9 11 9 ...
 $ age       : num  35 57 19 22 35 28 43 27 33 27 ...
 $ ethnicity : Factor w/ 3 levels "cauc","hispanic",..: 2 1 1 1 1 1 1 1 1 1 ...
 $ region    : Factor w/ 2 levels "south","other": 2 2 2 2 2 2 1 2 2 2 ...
 $ gender    : Factor w/ 2 levels "male","female": 2 2 1 1 1 1 1 1 1 1 ...
 $ occupation: Factor w/ 6 levels "worker","technical",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ sector    : Factor w/ 3 levels "manufacturing",..: 1 1 1 3 3 3 3 3 1 3 ...
 $ union     : Factor w/ 2 levels "no","yes": 1 1 1 1 1 2 1 1 1 1 ...
 $ married   : Factor w/ 2 levels "no","yes": 2 2 1 1 2 1 1 1 2 1 ...
```

- Abreviação níveis do fator occupation

√ Dados carregados do pacote

```
> levels(cps$occupation)
[1] "worker"     "technical"  "services"   "office"     "sales"
[6] "management"
> levels(cps$occupation)[c(2, 6)] <- c("mgmt", "techn")
> levels(cps$occupation)
[1] "worker"   "techn"    "services" "office"   "sales"    "mgmt"
```

√ Dados carregados do arquivo csv

```
> levels(cps$occupation)
[1] "management" "office"     "sales"      "services"   "technical"
[6] "worker"
> levels(cps$occupation)[c(1, 5)] <- c("mgmt", "techn")
> levels(cps$occupation)
[1] "mgmt"     "office"   "sales"    "services" "techn"    "worker"
```

√ Anexando conjunto de dados para trabalho:

```
> # anexando o arquivo
> attach(cps)
```


- Distribuição de `wage` na amostra:

√ Resumo dos 5 números e média

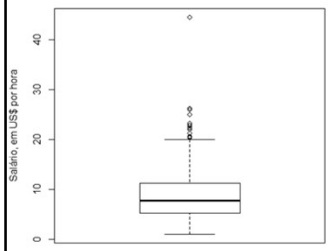
```
> summary(wage)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   5.250   7.780   9.024  11.250  44.500
> mean(wage)
[1] 9.024064
> median(wage)
[1] 7.78
```

√ Outros comandos para quantis

```
> fivenum(wage)
[1]  1.00  5.25  7.78 11.25 44.50
> quantile(wage)
   0%   25%   50%   75%  100%
 1.00  5.25  7.78 11.25 44.50
> quantile(wage, probs = c(0.05, 0.95))
   5%   95%
 3.50 19.98
> max(wage); min(wage)
[1] 44.5
[1] 1
```


- Box-plot da variável `wage`:

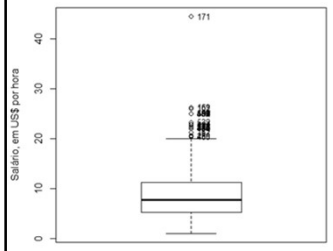
```
> boxplot(wage, ylab = "Salário, em US$ por hora")
```



√ Gráfico dos 5 números
√ Há *outliers*

- Identificação de *outliers* em `wage`:

```
> # identificação outliers
> wage.bxp <- boxplot(wage, ylab = "Salário, em US$ por hora", plot = F)
> outliers.row <- which(wage %in% wage.bxp$out)
> for(i in 1:length(wage.bxp$group)){
+     # adiciona texto no boxplot
+     text(wage.bxp$group[i], wage.bxp$out[i], which(wage==wage.bxp$out[i]),
+ pos = 4, cex = 0.8)
+ }
```



√ Há empates

• Detalhamento dos outliers

```
> outliers.row
 [1]  18  20 107 157 162 169 171 178 181 185 211 410 432 434 436 450 480 485 486
[20] 495 497 503 522 532
> length(outliers.row)
[1] 24
> cps[outliers.row,]
 wage education experience age ethnicity region gender occupation
17  22.20        12         26  44      cauc  other   male     worker
19  20.55        12         33  51      cauc  other   male     worker
106 26.00        14         21  41      cauc  other   male     worker
156 24.98        16         18  40      cauc  other   male       mgmt
161 21.25        13         32  51      cauc  other   male       mgmt
sector union married
17  manufacturing   yes     yes
19          other    no     yes
106         other   yes     yes
156         other    no     yes
161         other    no      no
```

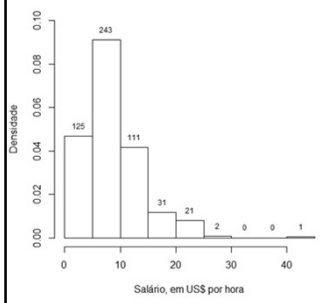

• Medidas de dispersão de `wage`:

√ Desvio-padrão, variância e distância interquartílica

```
> sd(wage)
 [1] 5.139097
> var(wage)
[1] 26.41032
> IQR(wage)
[1] 6
```


• Histograma da variável `wage`:

```
> wage.hst <- hist(wage, freq = FALSE, main = "", ylab = "Densidade",
+ xlab = "Salário, em US$ por hora", ylim = c(0, 0.105))
> text(wage.hst$mid, wage.hst$density + 0.005, wage.hst$counts, cex = 0.75)
```



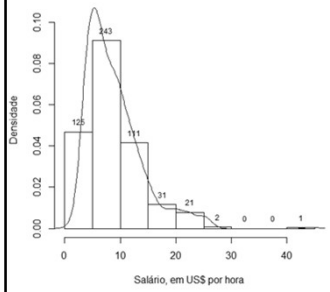
√ Fortemente assimétrica
√ Apenas um *outlier*?

• Assimetria e curtose de `wage`:

```
> library(moments)
> skewness(wage)
[1] 1.692514
> kurtosis(wage)
[1] 7.933936
```

• Histograma com suavização:

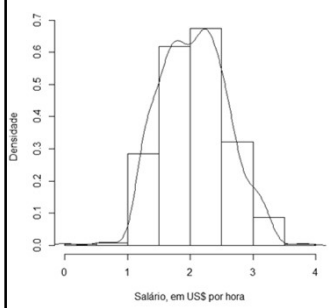
```
> # suavização por núcleo estimador
> lines(density(wage), col = 4)
```



√ Indício de bimodalidade?
- poucos dados para afirmar

• Histograma de `log(wage)`:

```
> hist(log(wage), freq = FALSE, main = "", ylab = "Densidade",
+ xlab = "Salário, em US$ por hora")
> lines(density(log(wage)), col = 4)
```



√ Distribuição de `log(wage)` é menos assimétrica

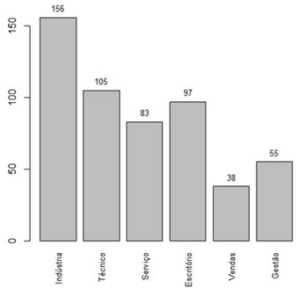
• Resumo da variável `occupation`:

√ Tabelas de frequências absolutas e relativas

```
> summary(occupation)
  worker    techn services   office    sales     mgmt
     156      105       83       97       38       55
> tab <- table(occupation)
> prop.table(tab)
occupation
    worker      techn   services     office      sales       mgmt
0.29213483 0.19662921 0.15543071 0.18164794 0.07116105 0.10299625
```


• *Barplot* da variável `occupation`:

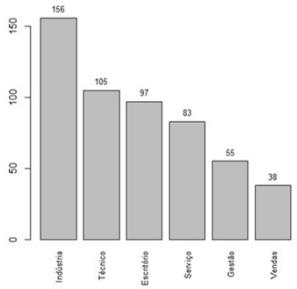
```
> nomes <- c("Indústria", "Técnico", "Serviço", "Escritório", "Vendas",
+ "Gestão")
> occup.bp <- barplot(tab, names.arg = nomes, cex.names = 0.85, las = 3,
+    ylim = c(0, 165))
> text(occup.bp, tab, labels = tab, cex = 0.85, pos = 3, offset = 0.5)
```



√ Ficaria melhor se as barras estivessem ordenadas por frequência?

- *Barplot* com as barras ordenadas:

```
> ordem <- order(tab, decreasing = T)
> occup.ord <- barplot(tab[ordem], names.arg = nomes[ordem], cex.names = 0.85,
+     las = 3, ylim = c(0, 165))
> text(occup.ord, tab[ordem], labels = tab[ordem], cex = 0.85, pos = 3,
+ offset = 0.5)
```



√ Pode ser conveniente quando houver muitos níveis

- *Pie chart* da variável `occupation`:

```
> pie(tab)
```

√ Eventualmente, podem ser úteis.

- Análise bivariada – categóricas

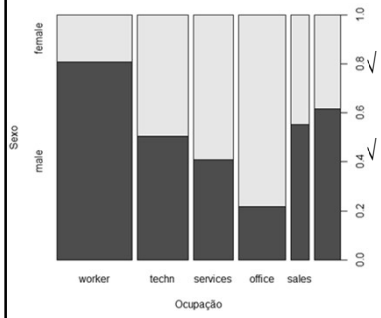
  (Variáveis: `gender` e `occupation`)

  √ Tabela de dupla entrada

```
> xtabs(~ gender + occupation, data = cps)
        occupation
gender   worker techn services office sales mgmt
  male      126    53       34     21    21   34
  female     30    52       49     76    17   21
```


- *Barplot* com duas variáveis categóricas:

```
> plot(gender ~ occupation, xlab = "Ocupação", ylab = "Sexo")
```



√ Proporção de homens e mulheres variam consideravelmente com a ocupação

√ Há mais pessoas trabalhando em "workers" que em "sales"

- Análise bivariada – quantitativas

(Variáveis: `log(wage)` e `education`)

√ Correlação de Pearson

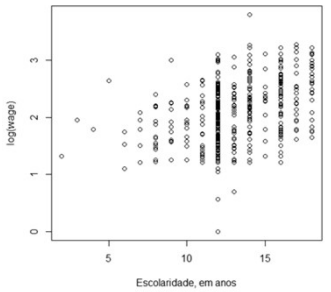
```
> cor(wage, education)
[1] 0.3819221
> cor(log(wage), education)
[1] 0.3803983
```

√ Correlação de Spearman – Não Paramétrico

```
> cor(wage, education, method = "spearman")
[1] 0.3813425
> cor(log(wage), education, method = "spearman")
[1] 0.3813425
```


- *Plot* para duas variáveis quantitativas:

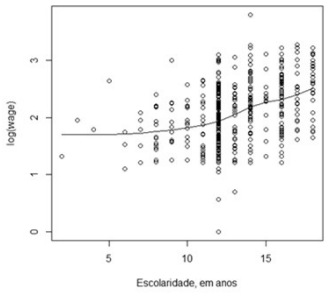
```
> plot(log(wage) ~ education, xlab = "Escolaridade, em anos")
```



√ Difícil perceber tendência
- Correlação linear baixa

- *Plot* com suavizador:

```
> plot(log(wage) ~ education, xlab = "Escolaridade, em anos")
> wage.lo <- loess(log(wage) ~ education)
> lines(sort(of.lo$x),sort(of.lo$fit))
```



√ Correlação baixa entre as variáveis

- Bivariada – quantitativa *vs*. categórica:

(Variáveis: `log(wage)` e `gender`)

√ Estatísticas descritivas por estrato

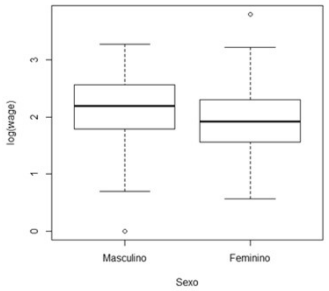
```
> # média por estrato
> tapply(log(wage), gender, mean)
    male   female
2.165286 1.934037
> # média e desvio padrão por estrato
> aggregate(log(wage) ~ gender, FUN = function(x) c(M=mean(x), SD=sd(x)))
gender log(wage).M log(wage).SD
1   male    2.165286     0.534453
2 female    1.934037     0.492118
```

√ Divisão da variável `wage` por `gender`:

```
> # Divisão wage por gender
> mwage <- subset(cps, gender == "male")$wage
> fwage <- subset(cps, gender == "female")$wage
```

• *Boxlot* wage por gender:

```
> plot(log(wage) ~ gender, xlab = "Sexo", xaxt = "n")
> axis(1, at = 1:2, labels = c("Masculino", "Feminino")
```

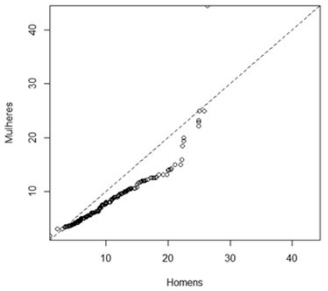


√ Similares as formas gerais de ambas as distribuições
√ Homens levam vantagem, principalmente na 'faixa média'

79

• *qq-plot* de wage por gender:

```
> # qq-plot de wage por gender
> qqplot(mwage, fwage, xlim = range(wage), ylim = range(wage), xaxs = "i",
+ yaxs = "i", xlab = "Homens", ylab = "Mulheres")
> abline(0, 1, lty = 2)
```

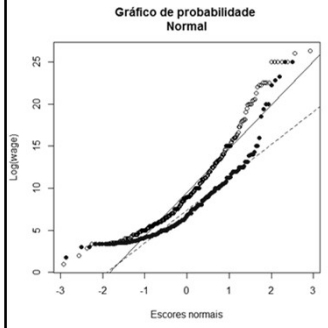


√ Na maioria dos quantis, salário dos homens é tipicamente mais alto

80

• Gráfico de probabilidade normal:

```
> qqnorm(mwage, ylab = "Log(wage)", xlab = "Escores normais",
+  main="Gráfico de probabilidade \nNormal")
> qqline(mwage)
> norm.fem <- qqnorm(fwage, plot.it = F)
> points(norm.fem$x, norm.fem$y, pch = 21, col = "blue", bg = "blue")
> qqline(fwage, lty = 2, col = "blue")
```



√ Distribuições similares com valores menores de quantis para Mulheres

81

Gráficos de Probabilidade

- Como saber se uma distribuição de probabilidades é um modelo razoável para os dados?
  - √ Pode-se fazer uma verificação de suposições:
    - Forma da distribuição, frequência esperada das observações
- Verificação gráfica:
  - √ Histogramas
    - Dão uma ideia da forma da distribuição,
    - Em geral não são indicadores confiáveis
      (a menos que o tamanho amostral seja grande)

- Gráfico de probabilidades:
  - √ Procedimento geral é simples
  - √ Mais confiável que histograma para tamanhos amostrais pequenos ou moderados
  - √ Usa eixos especiais, projetados para a distribuição hipotética

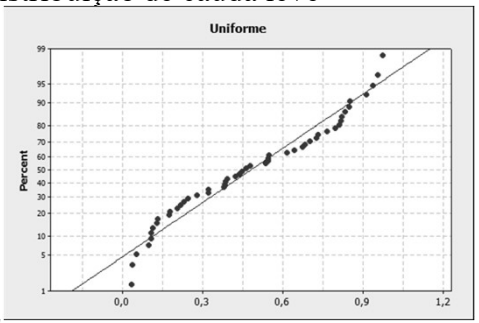
- Procedimento:
  - √ Ordenação das observações amostrais:
    - $x_{(1)}, x_{(2)}, ..., x_{(n)}$.
  - √ Plotam-se os pontos $(x_{(j)}; (j - 0,5)/n)$
    (observação, frequência acumulada observação)
  - √ Usa-se uma escala de probabilidade
  - √ Distribuição descreve adequadamente os dados:
    - pontos cairão, aproximadamente, ao longo de uma linha reta
  - √ Modelo hipotético não é apropriado
    - os pontos desviam-se significativamente de uma linha reta

- É subjetivo determinar se os pontos seguem ou não uma linha reta!

Gráfico de Probabilidades Normal

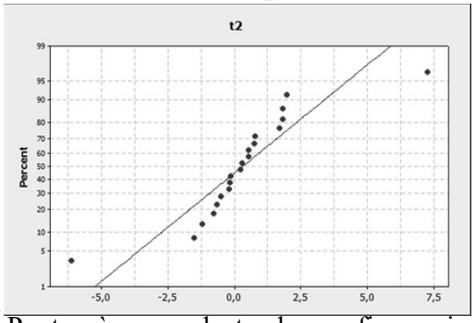
- Pode ser útil na identificação de distribuições que sejam simétricas mas que tenham caudas mais pesadas (ou mais leves) que a normal

- Distribuição de cauda leve



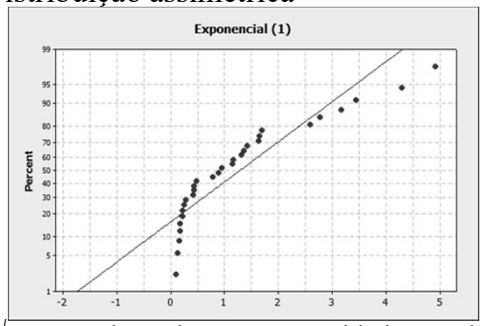
√ Pontos à esquerda tendem a ficar abaixo da linha e à direita tendem a ficar acima
  – As menores e maiores observações não serão tão extremas como se esperaria de uma normal

- Distribuição de cauda pesada



√ Pontos à esquerda tendem a ficar acima da linha e à direita tendem a ficar abaixo
√ Gráfico em forma de S

- Distribuição assimétrica



√ Pontos de ambas as extremidades tendem a estar abaixo da linha
√ Gráfico tem forma curvada

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.

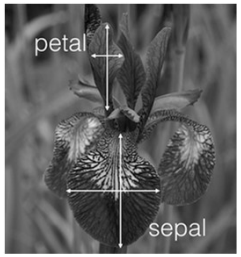
Importante

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

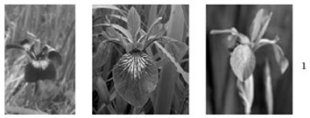
Conjunto de Dados

- 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

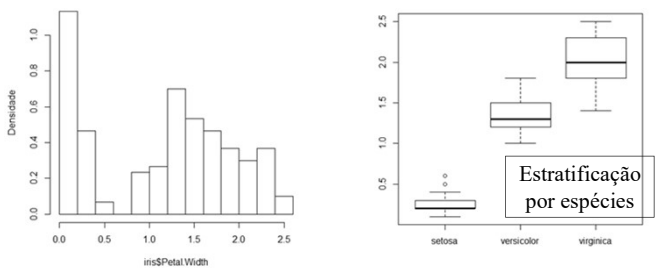
```
> dim(iris)
[1] 150   5
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 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`:

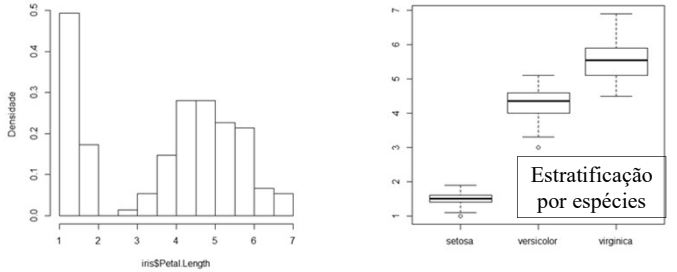
```
> # Petal.Width
> hist(iris$Petal.Width, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Petal.Width ~ Species, data = iris)
```



√ Mistura de populações

• Histograma da variável `Petal.Length`:

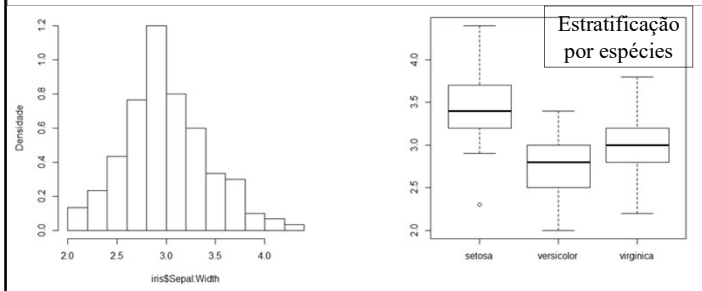
```
> # Petal.Length
> hist(iris$Petal.Length, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Petal.Length ~ Species, data = iris)
```



√ Mistura de populações

• Histograma da variável `Sepal.Width`:

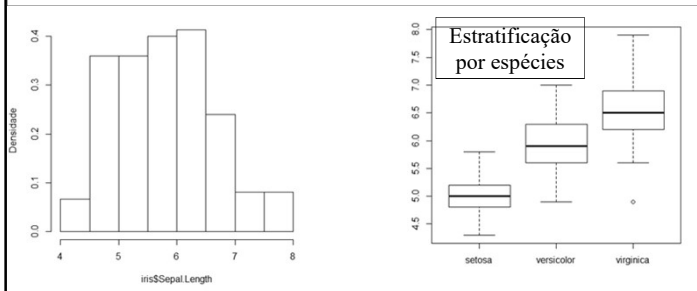
```
> # Sepal.Width
> hist(iris$Sepal.Width, freq = F, ylab = "Densidade", main = "")
> win.graph()
> boxplot(Sepal.Width ~ Species, data = iris)
```



√ Mistura de populações menos acentuada

• Histograma da variável `Sepal.Length`:

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

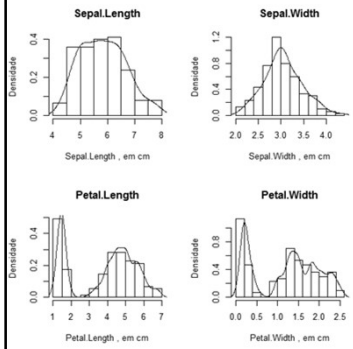


√ Mistura de populações mais próximas

- 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))
```


- Histogramas com suavizador:

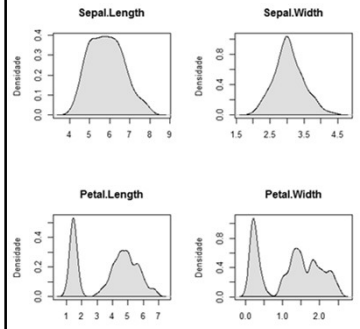


√ Facilita visualização das misturas

- 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.

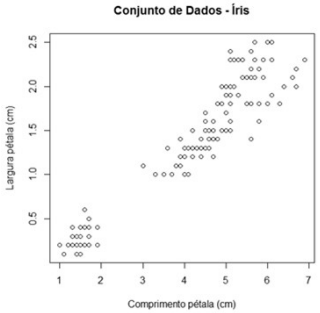
```
> library(aplpack)
> #Fonte: http://www.statmethods.net/graphs/boxplot.html
> # Bagplot de Largura de pétala
> bagplot(iris$Petal.Length, iris$Petal.Width,
+ xlab = "Comprimento pétala (cm)", ylab = "Largura pétala (cm)",
+   main = "Bagplot de Petal.Width")
```


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

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

• Diagrama de Dispersão - Pétalas

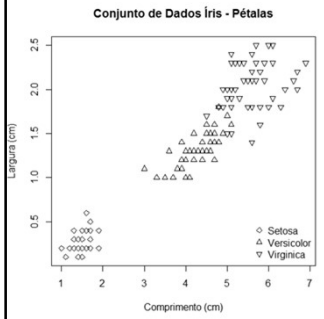
```
> # scatter plot simples
> plot(iris$Petal.Length, iris$Petal.Width, main="Conjunto de Dados - Íris",
+ xlab = "Comprimento pétala (cm)", ylab = "Largura pétala (cm)")
```



√ Tendência linear
√ Há dois agrupamentos de dados

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

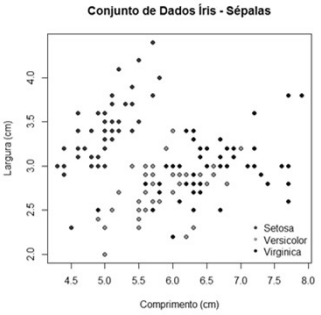
```
> # Scatter plot com o fator 'Species' - Pétalas
> plot(iris$Petal.Length, iris$Petal.Width, pch =
c(23,24,25)[unclass(iris$Species)],
+ main = "Conjunto de Dados Íris - Pétalas", xlab = "Comprimento (cm)",
+ ylab = "Largura (cm)")
> legend("bottomright", legend = c("Setosa", "Versicolor", "Virginica"),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

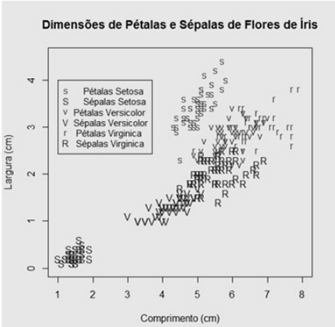


√ Grupo das setosas está bem discriminado
√ Discriminação entre os dois outros grupos não está tão clara

• *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`:

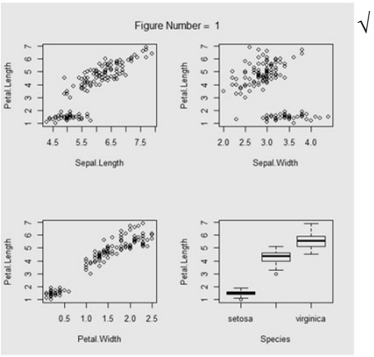
```
> library(devtools)
> install_github("ujjwalkarn/xda")
> library(xda)
> # resumo de todas as variáveis quantitativas
> numSummary(iris)
             n mean    sd max min range nunique nzeros  iqr lowerbound
Sepal.Length 150 5.84 0.828 7.9 4.3   3.6      35      0 1.30       3.15
Sepal.Width  150 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')
```

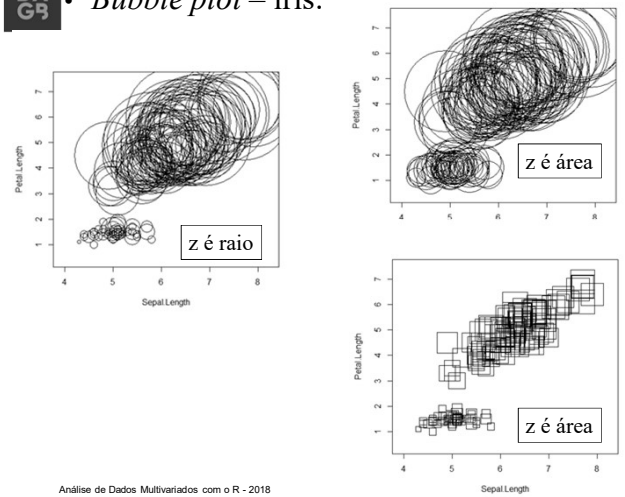


√ Gráficos bivariados com `Petal.Length` aparentam discriminar estratos de `Species`

- *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))
```

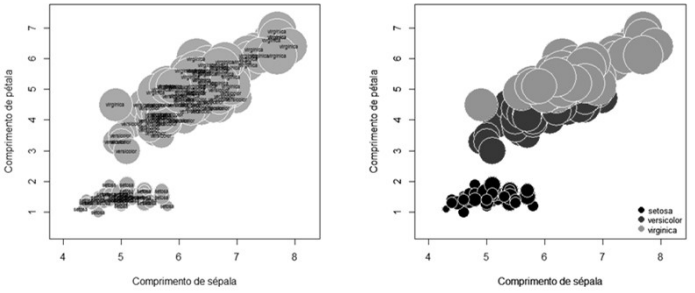

- *Bubble plot* – iris:



- *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)
+ })
```

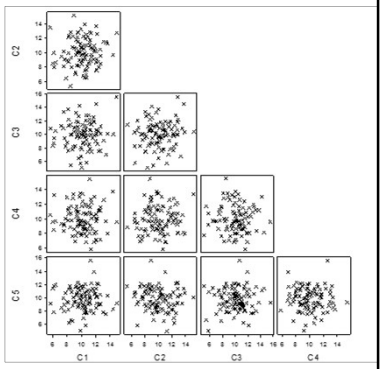

- *Bubble plot* – iris:



- √ Espécie setosa têm pétalas mais estreitas

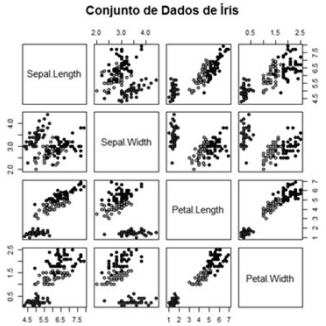
Scatter Plot Matrix

- 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:

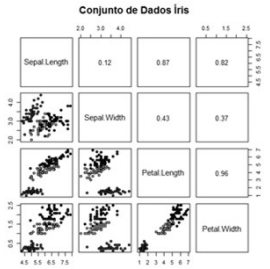
```
> pairs(iris[1:4], main = "Conjunto de Dados de Íris", pch = 21,
+ bg = c("red", "green3", "blue")[unclass(iris$Species)])
```



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

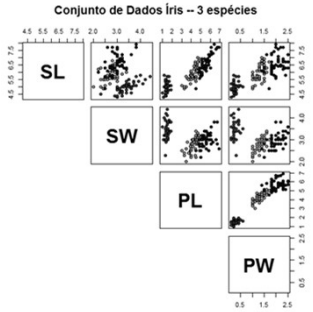
• *Scatter plot matrix* com correlações:

```
> # 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
```

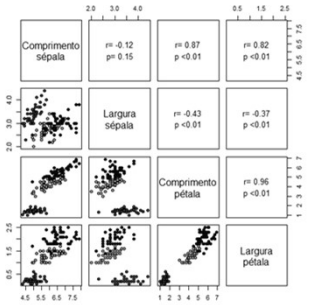


- *Scatter plot matrix* com correlação e p-valor:
  - √ Função para personalização do painel

```
> # 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"))
```

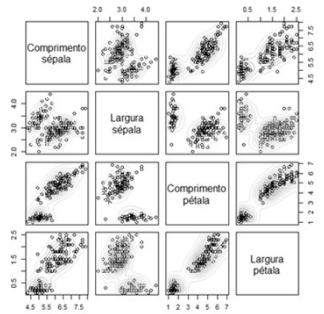


- *Scatter plot matrix* com estimativas de densidade bivariada:
  - √ Função para estimativas de densidade bivariada

```
> 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:

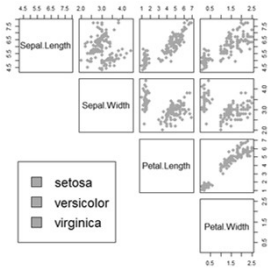
```
> # Scatter plot matrix
> pairs(iris[,1:4], panel = painel.dens,
+ labels = c("Comprimento\nsépala", "Largura\nsépala",
+ "Comprimento\npétala", "Largura\npétala"))
```



- √ 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)
```



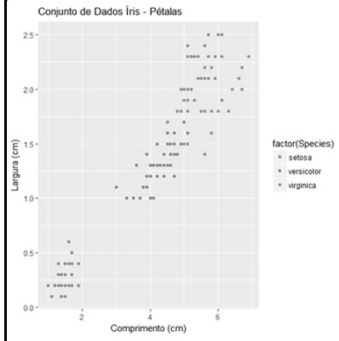
• *Scatter plot matrix* com pacote `ggplot2`:

```
> 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`:

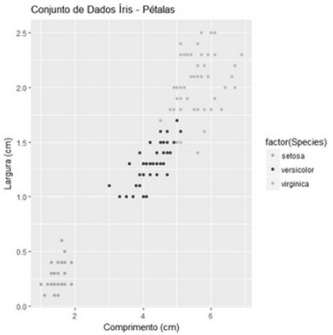
```
> # Mudança de cor dos pontos
> sp2 <- sp1 + geom_point(aes(color = factor(Species))) + # cor p/ nível fator
+  xlab("Comprimento (cm)") +
+    ylab("Largura (cm)") +
+  ggtitle("Conjunto de Dados Íris - Pétalas")
> sp2
```



√ Pontos com códigos de cores

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

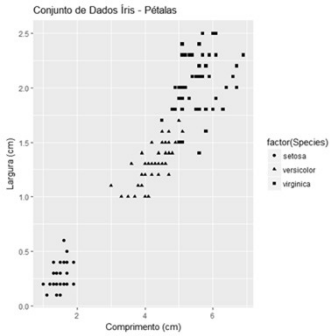
```
> # Cores conforme usuário
> sp3 <- sp1 + geom_point(aes(color=factor(Species))) +
+ scale_color_manual(values = c("orange", "purple", "gray")) +
+ xlab("Comprimento (cm)") +
+   ylab("Largura (cm)") +
+ ggtitle("Conjunto de Dados Íris - Pétalas")
> sp3
```



√ 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
```

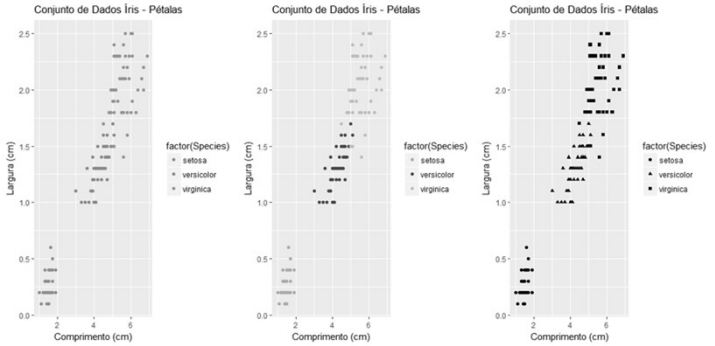


√ Modificação da forma e do tamanho dos pontos

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

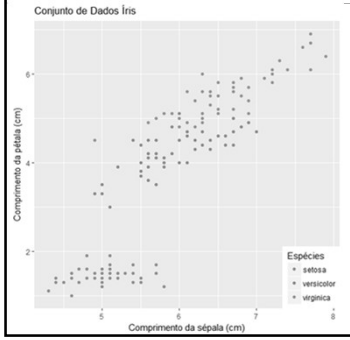
√ Painel com gráficos

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



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

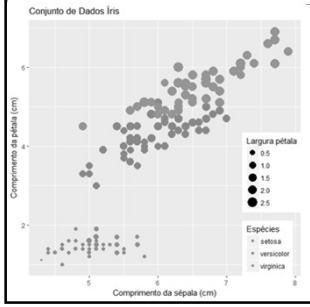
```
> # Scatterplot comprimentos + espécies
> ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
+ geom_point()+
+ xlab("Comprimento da sépala (cm)") +
+   ylab("Comprimento da pétala (cm)") +
+ ggtitle("Conjunto de Dados Íris") +
+ scale_color_discrete(name ="Espécies") +
+ theme(legend.position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

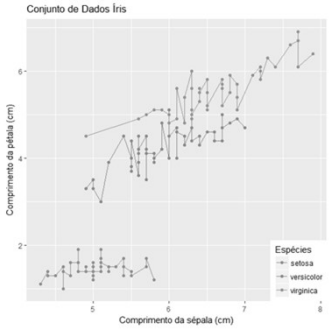
```
> # Scatterplot comprimentos vs. espécies vs.largura pétala (bubble)
> ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species,
+  size = Petal.Width), alpha = I(0.7)) +# alpha: reduz overplotting
+ geom_point() +
+ xlab("Comprimento da sépala (cm)") +
+   ylab("Comprimento da pétala (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:

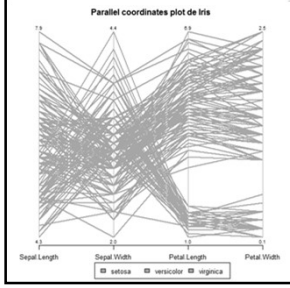
```
>   # Scatter plot em linhas
> ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
+ geom_line()+ geom_point() +
+ xlab("Comprimento da sépala (cm)") +
+   ylab("Comprimento da pétala (cm)") +
+ ggtitle("Conjunto de Dados Íris") +
+ scale_color_discrete(name ="Espécies") +
+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

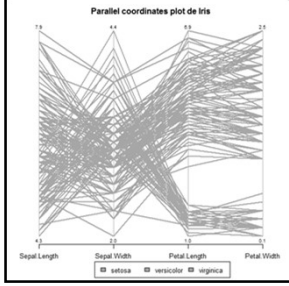
```
> library(MASS)
> library(colorspace) # get nice colors
> species.cor <- rev(rainbow_hcl(3))[as.numeric(iris$Species)]
> par(las = 1, mar = c(4.5, 3, 3, 2) + 0.1, cex = .8)
> parcoord(iris[-5], col = species.cor, var.label = TRUE, lwd = 2)
> title("Parallel coordinates plot de Iris")
> par(xpd =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



√ Flores da espécie setosa têm as pétalas mais estreitas

• *Parallel coordinate plot*:

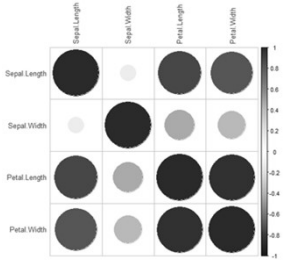
```
> library(MASS)
> library(colorspace) # get nice colors
> species.cor <- rev(rainbow_hcl(3))[as.numeric(iris$Species)]
> par(las = 1, mar = c(4.5, 3, 3, 2) + 0.1, cex = .8)
> parcoord(iris[-5], col = species.cor, var.label = TRUE, lwd = 2)
> title("Parallel coordinates plot de Iris")
> par(xpd =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



√ Flores da espécie setosa têm as pétalas mais estreitas

• Correlograma:

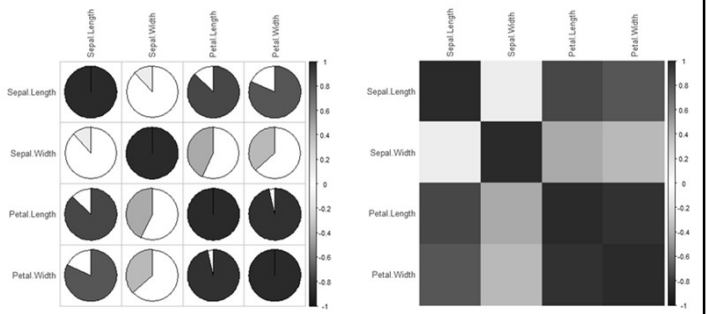
```
        # Matriz de correlações
(iris.cor <- cor(iris[-5]))
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000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



√ círculos

• Correlograma:

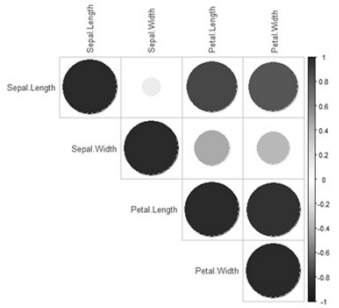
```
> # correlograma - pizza
> corrplot(iris.cor, method = "pie")
> # coorelograma - cor
> corrplot(iris.cor, method = "color")
```



• Correlograma:

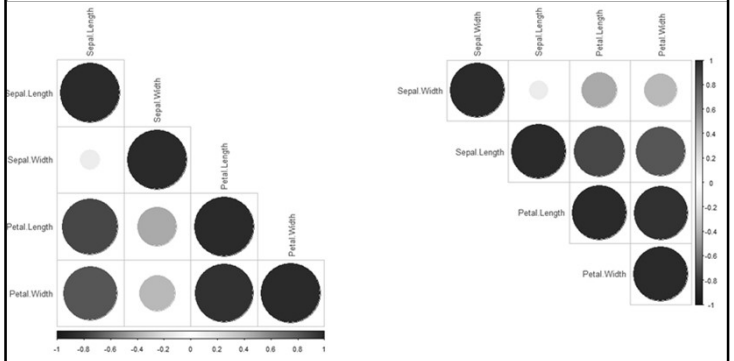
```
> # correlograma - valores
> corrplot(iris.cor, method = "number")
> # correlograma - superior
> corrplot(iris.cor, type = "upper")
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1	-0.12	0.87	0.82
Sepal.Width	-0.12	1	-0.43	-0.37
Petal.Length	0.87	-0.43	1	0.96
Petal.Width	0.82	-0.37	0.96	1



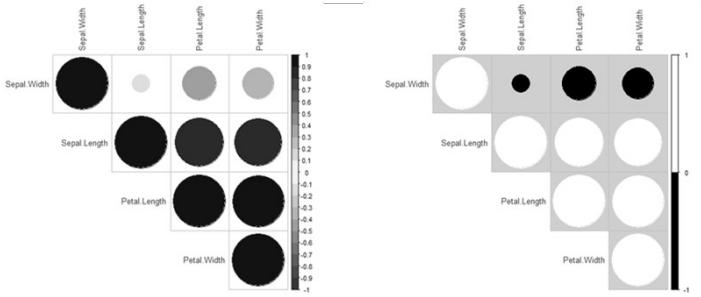
• Correlograma:

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



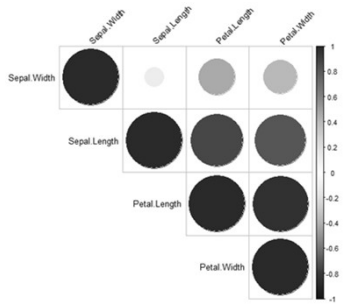
• Correlograma:

```
> # 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

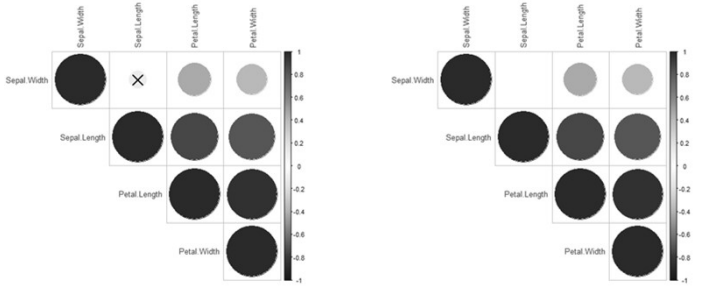
```
> # Função para cálculo do p-valor das correlações
> cor.mteste <- function(mat, ...) {
+     mat <- as.matrix(mat)
+     n <- ncol(mat)
+     p.mat<- matrix(NA, n, n)
+     diag(p.mat) <- 0
+     for (i in 1:(n - 1)) {
+         for (j in (i + 1):n) {
+             tmp <- cor.test(mat[, i], mat[,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

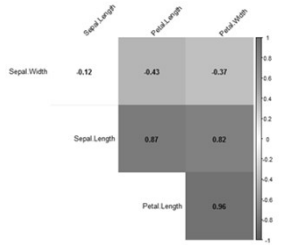

• Correlograma:

```
> # 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")
```



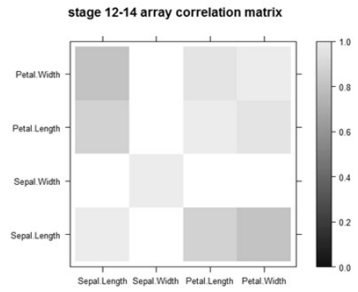
• Correlograma:

```
> # 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
+          )
```



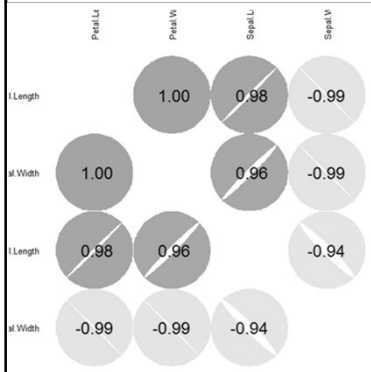
• Matriz de Correlações – pacote `lattice`:

```
> 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)
+ )
```



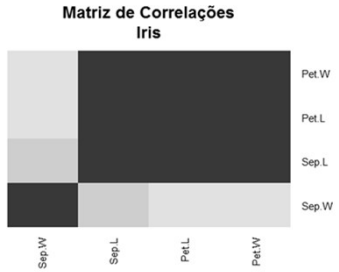
• Matriz de Correlações – pacote `lattice`:

```
> 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:

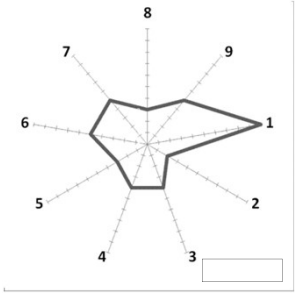
```
> library(plotly)
> p1 <- plot_ly(data = iris, x =~Sepal.Length, y =~Sepal.Width, split =~Species,
+   showlegend = F)
> p2 <- plot_ly(data = iris, x =~Sepal.Length, y =~Sepal.Width, split =~Species,
+   showlegend = T)
> subplot(p1,p2)
```

√ 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

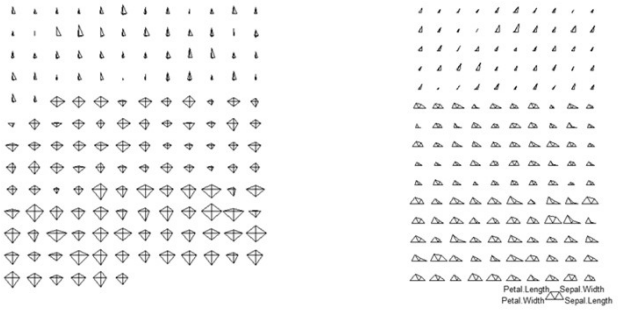
- Estrelas para visualização de dados
- Formação da estrela:
  - √ Raio para cada variável
  - √ Comprimento é proporcional à variável
- Útil para visualização de itens com número arbitrário de variáveis



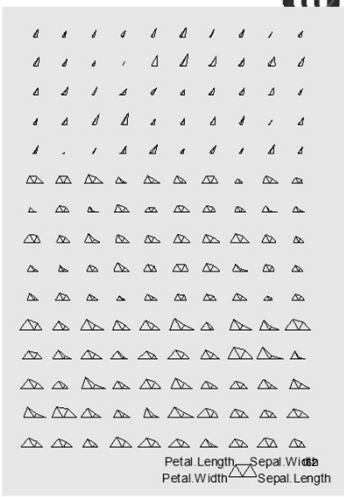
- Pode ser usado para responder as seguintes perguntas:
  - √ Quais variáveis são dominantes para uma determinada observação?
  - √ Quais observações são similares?
    (Existem agrupamentos de observações?)
  - √ Existem valores discrepantes?

- *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?



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.