

## Estatística Computacional I

Lupércio França Bessegato  
Dep. de Estatística/UFJF



### Roteiro Geral



1. Programando em R
2. Gráficos em R
3. Preparação e limpeza de dados
4. Manipulação de dados
5. Tópicos especiais
6. Referências

Estatística Computacional I - 2020

2

## Gráficos em R

## Gráficos e seus Argumentos – R Básico



## Pacote graphics



- Funções do sistema gráfico tradicional do R:

- ✓ Pacote `graphics`:

- Carregado automaticamente com o R

- ✓ Função `example()`:

- Executa código para exemplo de função gráfica

```
> # help de funções gráficas básicas
> par(ask = TRUE)
> example(barplot)
```

- `par(ask = TRUE)`:

- Avisa usuário antes de cada nova página
      - Caso contrário, exemplos tendem a passar rápido demais.

Estatística Computacional I - 2020

53

## Modelo Gráfico Tradicional



- Função gráfica de nível superior:

- ✓ Cria gráfico completo

- Função gráfica de nível inferior:

- ✓ Acrescenta saídas ao gráfico ativo

- Registro da saída gráfica

- ✓ Janela gráfica ou arquivo

Estatística Computacional I - 2020

54



- Modificação da saída gráfica:
- ✓ Modificar código e executá-lo novamente
    - ✓ Edição por outro software
    - Ex.:
    - Saída com função `xfig()`, edição com programa `xfig`.



Estatística Computacional I - 2020

55



## Plots Univariado e Bivariado



- Tipos básicos de plot:

- ✓ `plot()`.

- ✓ `barplot()`.

- ✓ `hist()`.

- ✓ `boxplot()`.

- ✓ `pie()`.

Estatística Computacional I - 2020

56

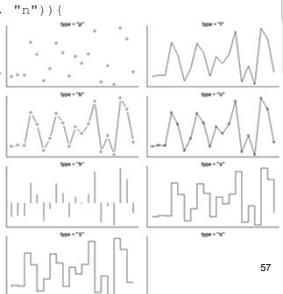


✓ `plot()` constrói *scatterplot* com:

- Símbolos na posição dos dados
- Linhas conectando posições dos dados

✓ Variações dadas pelo argumento `type`.

```
> # comando plot() - argumento type
> par(mfrow = c(4, 2), mar = rep(0.7, 4), xaxt = "n", yaxt = "n", cex = 0.8,
+ cex.main = 0.9, bty = "I")
> y <- rnorm(20)
> for(i in c("p", "l", "b", "o", "h", "s", "S", "n")) {
+ plot(y, type = i)
+ title(main = paste0("type = '", i, "'"))
+ }
```



Estatística Computacional I - 2020



• Não há distinção entre *plots* contendo:

- Um simples conjunto de dados
- Múltiplos conjuntos de dados

✓ Funções de nível inferior para inserir séries adicionais de dados no plot:

- `points()`.
- `lines()`.



58

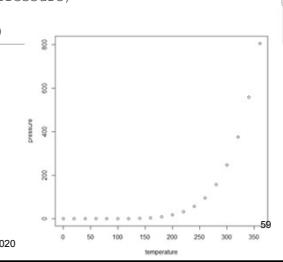
Estatística Computacional I - 2020



- Funções de nível superior:
  - ✓ Primeiro argumento são os dados.
- Flexibilidade para especificar os dados
  - ✓ Comandos para construir *scatterplot*:



```
> # especificação de dados a serem plotados
> # conjunto de dados com duas variáveis
> plot(pressure)
> # especificação vetor em x e em y
> plot(x = pressure$temperature, y = pressure$pressure)
> # especificação no formato fórmula
> plot(pressure ~ temperature, data = pressure)
```

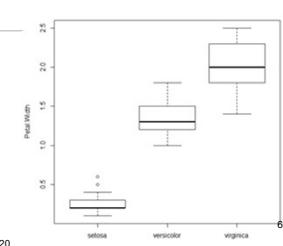


Estatística Computacional I - 2020



- As funções básicas do sistema gráfico tradicional são genéricas
  - ✓ Comportamento depende da classe do primeiro argumento da função
  - ✓ Relevante para a função `plot()`.

```
> # x é fator e y é numeric ou integer
> plot(Petal.Width ~ Species, data = iris)
```



Estatística Computacional I - 2020



- Produz gráfico de barras se x é fator e y = NULL.

```
> # x é fator e y = NULL
> plot(iris$Species)
```

61

Estatística Computacional I - 2020

- Produz gráfico de barras proporcional se x é tabela.

```
> # x é objeto table
> # Sexo de crianças não sobreviventes por classe
> tabela <- Titanic[, , Age = "Child", Survived = "No"]
> # plot da tabela
> plot(tabela)
```

tabela

Class

62

Estatística Computacional I - 2020

- Gráficos diagnóstico de regressão se x é objeto lm.

```
> # x é objeto lm
> # ajuste do modelo de regressão
> lm.SR <- lm(sr ~ pop15 + pop75 + dpi + dpi, data = LifeCycleSavings)
> # gráficos de diagnóstico - 1 por vez
> plot(lm.SR)
Esperando para confirmar mudança de página...
> # gráficos de diagnósticos em 4 painéis
> par(mfrow = c(2, 2))
> for(i in 1:4) plot(lm.SR)
> par(mfrow = c(1, 1))
```

63

Estatística Computacional I - 2020

✓ Métodos para plot() definidos por pacotes:

- Objeto agnes definido pelo pacote cluster.

```
> # comando plot do pacote cluster - objeto agnes
> library(cluster)
> # subconjunto de iris de tamanho 20
> subset <- sample(1:150, 20)
> cS <- as.character(SP <- iris$Species[subset])
> cS[SP == "setosa"] <- "S"
> cS[SP == "versicolor"] <- "V"
> cS[SP == "virginica"] <- "g"
> # procedimento de agrupamento
> # hierárquico aglomerativo
> ai <- agnes(iris[subset, 1:4])
> # gráficos do agrupamento -
> par(mfrow = c(2, 1))
> plot(ai, labels = cS)
```

Banner of agnes(x = iris[subset, 1:4])

Agglomerative Coefficient = 0.86

Dendrogram of agnes(x = iris[subset, 1:4])

iris[subset, 1:4]  
Agglomerative Coefficient = 0.86

64

Estatística Computacional I - 2020



- Função `matplot()`:
  - ✓ Não usa método `plot()`.
  - ✓ Especificamente desenvolvida para funcionar como `plot()`, com `x` ou `y` dados por matrizes
  - ✓ Apropriada para plotar várias séries de dados em um único *scatterplot*.
    - Dados são percebidos automaticamente, sendo usados diferentes cores e símbolos



Estatística Computacional I - 2020

65



- Gráfico de média semanal de peso por dieta:



```
> # Exemplo do comando matplot()
>
> # preparação dos dados
> # media do peso por semana e dieta
> medias <- aggregate(weight ~ Time + Diet, data = ChickWeight, FUN = mean)
> # lista de vetores de medias semanais por dieta
> colunas <- split(medias$weight, medias$Diet)
> # matriz com uma coluna para médias semanais de cada dieta
> matriz <- matrix(unlist(colunas), ncol = 4, byrow = F)
> colnames(matriz) <- paste0("#", 1:4)
> # Número da semana
> Tempo <- unique(ChickWeight$Time)
```

Estatística Computacional I - 2020

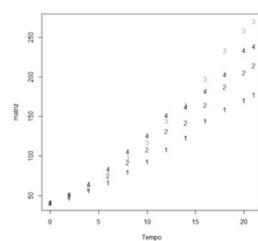
67



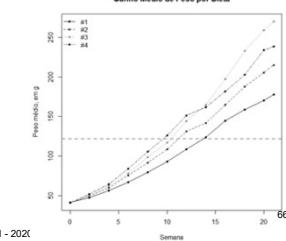
### ✓ Conjunto de dados `ChickWeight`:

– Matriz de média semanal de peso por dieta:

```
> # Construção do Gráfico
> # matplot - # semana para cada curva
> matplot(x = Tempo, y = matriz)
> # gráfico de ganho de peso médio semanal por dieta
> matplot(x = Tempo, y = matriz, type = "o", pch = 20, xlab = "Semana",
+ ylab = "Peso médio, em g", main = "Ganho Médio de Peso por Dieta")
> # legenda por dieta
> legend("topleft", legend = colnames(matriz), lty = 1:4, pch = 20, col = 1:4,
+ bty = "n")
> abline(h = mean(ChickWeight$weight)+10, lty = 2, lwd = 2, col = "gray60")
```



Estatística Computacional I - 2020



66



- Além das funções gráficas tradicionais, o R base tem outras funções de nível superior:

### ✓ `stripchart()`:

– Produz *scatterplot* para uma única variável

### ✓ `curve()`:

– Desenha curva de função matemática

### ✓ `stem()`:

– Produz ramo-e-folha

Estatística Computacional I - 2020

68

√ `stripchart()`:

– Variável do conjunto de dados `OrchardSprays`:

```
> # stripchart()
> # plot para única variável
> stripchart(decrease ~ treatment, , data = OrchardSprays, vertical = TRUE,
+ log = "y", main = "stripchart(OrchardSprays)")
```

`stripchart(OrchardSprays)`

treatment	decrease (approx.)
A	2, 3, 4, 5, 6, 7, 8, 10, 12, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90
B	4, 5, 6, 7, 8, 10, 12, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90
C	5, 6, 7, 8, 10, 12, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90
D	10, 12, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90
E	15, 20, 25, 30, 40, 50, 60, 70, 80, 90
F	20, 25, 30, 40, 50, 60, 70, 80, 90
G	30, 40, 50, 60, 70, 80, 90
H	50, 60, 70, 80, 90

The slide features the official logo of the University of São Paulo (USP) in the top left corner, which consists of a stylized 'U' and 'S' composed of geometric shapes. In the top right corner, there is a decorative graphic element consisting of a black square containing a white stylized 'B' shape made of vertical bars.



## √ stem():

– Ramo-e-folhas de islands:

```
> # stem() - ramo-e-folhas
> stem(islands)
The decimal point is 3 digit(s) to the right of the |
 0 | 0000000000000000000000000000000011111222338
 2 | 07
 4 | 5
 6 | 8
 8 | 4
10 | 5
12 |
14 |
16 | 0
> stem(log10(islands))
The decimal point is at the |
 1 | 11111222223344
 1 | 555555666667899999
 2 | 3344
 2 | 59
 3 |
 3 | 5678
 4 | 012
```

- Pacotes podem apresentar funções gráficas de nível superior:  
√ `labcurve()` {Hmisc}:
  - Desenha plot com linhas através de várias séries de dados.



- FAZER EXEMPLO

✓ `labcurve()` {Hmisc}:

- Desenha plot com linhas através de várias séries de dados.



## Argumentos das Funções Gráficas

- Em geral, a execução de uma função de nível superior não produz a saída ideal
  - ✓ Há muitas maneiras de modificar a saída de funções gráficas
    - Argumentos em funções de nível superior
    - Execução de funções complementares de nível inferior complementares



✓ Argumentos da função `boxplot()`:

- `width`: controla largura relativa de cada caixa
- `boxwex`: controla largura todas as caixas
- `horizontal`: caixas na horizontal (lógico)

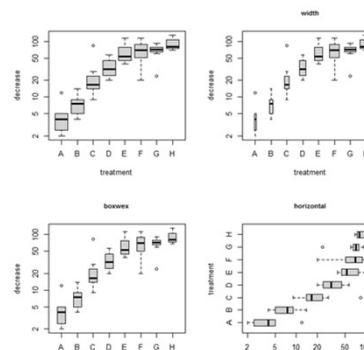


```
> # boxplot
> par(mfrow = c(2,2), cex.main = 0.8)
> # proporções default
> boxplot(decrease ~ treatment, data = OrchardSprays, log = "y",
+   col = "light grey")
> # argumento width
> proporcao <- 1:8/(8*9/2)
> boxplot(decrease ~ treatment, data = OrchardSprays, log = "y",
+   width = proporcao, col = "light grey", main = "width")
> # argumento boxwex
> boxplot(decrease ~ treatment, data = OrchardSprays, log = "y",
+   col = "light grey", boxwex = 0.5, main = "boxwex")
> # argumento horizontal
> boxplot(decrease ~ treatment, data = OrchardSprays, log = "x", boxwex = 0.5,
+   col = "light grey", horizontal = TRUE, main = "horizontal")
```



✓ Mudança em aspectos de `boxplot()`:

- Resultado comandos com argumentos diversos



✓ Adiciona caixa com todos os resultados:

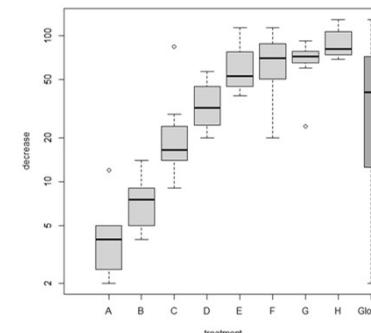
- Argumentos da função `boxplot()`:
  - `add`: adiciona boxplot em gráfico ativo (lógico).
  - `at`: posiciona box-plot
  - `xlim`: modifica dimensão eixo x (posição caixas).
  - `xaxt`: suprime eixo x do gráfico.
- Função gráfica *low-level*:
  - `axis`: modifica eixo gráfico

```
> # Inserção de box Global
> # argumento xlim e xaxt
> boxplot(decrease ~ treatment, data = OrchardSprays, log = "y",
+ col = "light grey", xlim = c(0, 9), xaxt = 'n')
> # inserção box-plot - argumento add e at
> boxplot(OrchardSprays$decrease, log = "y", add = T, at = 9, col = "darkgrey")
> # função low-level axis()
> axis(1, at = 1:9, labels = c(LETTERS[1:8], "Global"))

77
Estatística Computacional I - 2020
```

✓ Inserção de caixa adicional em *box-plot*:

- Gráfico personalizado resultante



A box plot with 'decrease' on the y-axis (ranging from 24 to 100) and 'treatment' on the x-axis. The treatments are labeled A, B, C, D, E, F, G, H, and Global. Each treatment has a box plot showing the median, quartiles, and whiskers. Treatment Global has a separate box plot at the far right.

```
78
Estatística Computacional I - 2020
```

✓ Argumentos da função `barplot()`:

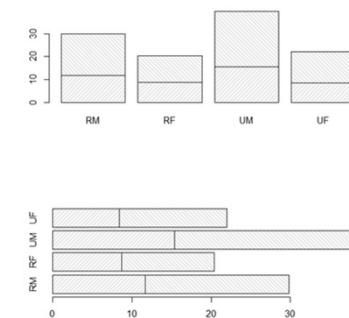
- `names`: denomina níveis da variável (barras)
- `angle`: ângulo das hachuras
- `horizontal`: caixas na horizontal (lógico)

```
> # conjunto de dados
> data(VADeaths)
> ?VADeaths
>
> # gráfico com 2 painéis
> par(mfrow = c(2, 1))
>
> # argumentos angle e names
> barplot(VADeaths[1:2, ], angle = c(45, 135), density = 20, col = "grey",
+ names = c("RM", "RF", "UM", "UF"))
> # argumento horizontal
> barplot(VADeaths[1:2, ], angle = c(45, 135), density = 20, col = "grey",
+ names = c("RM", "RF", "UM", "UF"), horiz = TRUE)

79
Estatística Computacional I - 2020
```

✓ Mudança em aspectos de `barplot()`:

- Resultado comandos com argumentos diversos



Two bar plots showing 'VADeaths' data. The top plot is a vertical bar plot with 'RM', 'RF', 'UM', and 'UF' on the x-axis. The bars are grey with diagonal hatching and are oriented vertically. The bottom plot is a horizontal bar plot with 'RM', 'RF', 'UM', and 'UF' on the y-axis. The bars are grey with diagonal hatching and are oriented horizontally.

```
80
Estatística Computacional I - 2020
```

## Referências



## Bibliografia Recomendada



- ALBERT, J.; RIZZO, M. *R by Example*. Springer, 2012.
- CHRISTIAN, N. *Basic Programming*, Lecture Notes
- DALGAARD, P. *Introductory statistics with R*. Springer, 2008.
- MURRELL, P. *R Graphics*. Chapman & Hall, 2006.

Estatística Computacional I - 2020

244