

# Estatística Computacional I

Lupércio França Bessegato  
Dep. de Estatística/UFJF

## Gráficos em R

### Roteiro Geral

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

Estatística Computacional I - 2020

2

### Roteiro do Módulo

3. Gráficos em R:
  - a) Introdução
  - b) Gráficos e argumentos padrão
  - c) Personalização de gráficos tradicionais
  - d) Alguns pacotes gráficos
  - e) Referências

Estatística Computacional I - 2020

4

## Introdução

## Objetivo

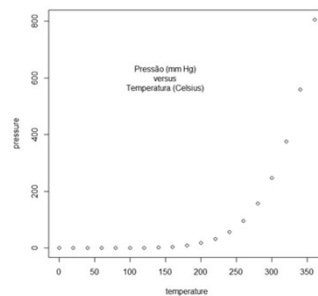
- Visão geral do sistema gráfico do R
  - ✓ Funções gráficas:
    - Operação e modificação
  - ✓ Customização de gráficos
  - ✓ Não se discutirá qual tipo de gráfico é mais apropriado para visualização gráfica correta

Estatística Computacional I - 2020

6

## • Gráficos no R base:

```
> # Fig.1.1 - simple scatter plot
> plot(pressure)
> text(150, 600, "Pressão (mm Hg)\nversus\nTemperatura (Celsius)")
```



- ✓ `plot(pressure)`:
  - Desenha o gráfico básico
    - Eixos, símbolos dos dados, retângulo circundando o gráfico
- ✓ `text()`:
  - Anota texto no ponto indicado do gráfico.

Estatística Computacional I - 2020

7

- Produção de saídas gráficas com R base:
  - ✓ Funções que produzem uma figura completa
  - ✓ Funções que adicionam alguma saída em figura existente
- Segue “modelo do pintor”:
  - ✓ Saída gráfica ocorre em etapas
  - ✓ Saída posterior oculta saídas sobrepostas (anteriores)

Estatística Computacional I - 2020

8



- Pacotes com funções gráficas do R Base são carregados ao iniciar o R
- Há muitos pacotes com funções gráficas especializadas



Estatística Computacional I - 2020

9

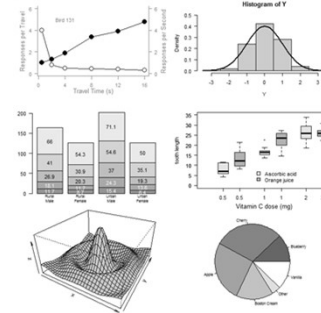


## Exemplos de Gráficos no R



### • Gráficos padrão:

```
> # Exemplos de Gráficos
> # Fonte: MURRELL, P. R Graphics. Chapman & Hall, 2006
> caminho <- "codes/murrell"
> # Fig. 1.2 - alguns gráficos padrão
> source(file.path(caminho, "fig.1.2_examples-stdplots.R"))
```



✓ Diagrama de dispersão, histograma, barplot, boxplot, gráfico em 3D, gráfico de setores

Estatística Computacional I - 2020

10



- Comando `plot()` :  
✓ Pode ser considerado ponto de partida para produzir imagens mais complexas



Estatística Computacional I - 2020

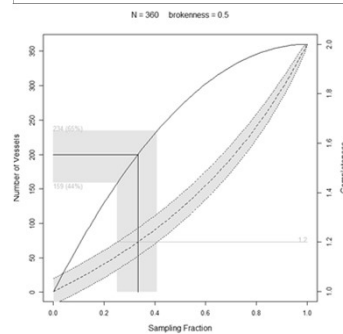
11



### • Flexibilidade para incorporar elementos:





```
> caminho <- "codes/murrell"
> # Fig. 1.3 - Flexibilidade na construção de gráficos
> source(file.path(caminho, "fig.1.3_examples-simonmatt.R"))
```



✓ Gráfico base:  
– diagrama de dispersão  
✓ Elementos adicionais:  
– linhas extras, polígonos, texto, múltiplos sistemas de coordenadas

Estatística Computacional I - 2020

12

✓ Contexto do gráfico:

- Estimação número original de vasos baseando-se em fragmentos reunidos em sítio arqueológico

✓ “completeness”



- Fração de amostragem, calculada a partir dos fragmentos no local

✓ Número original de vasos:

- Outra relação teórica


13

Estatística Computacional I - 2020

• *Trellis plots:*

```
> caminho <- "codes/murrell"
> # Fig. 1.4 - Trellis graph - package: lattice
> source(file.path(caminho, "fig.1.4_examples-dotplot.R"))
```



✓ Condicionado em painéis:


- Plots separados de acordo com níveis de outra variável

✓ Rendimentos de variedades de cevada:

- 6 locais diferentes:
  - Painel com faixa nome do site
- Período de 2 anos
  - Símbolos distintos por ano

14

Estatística Computacional I - 2020




• Gráficos especiais:

```
> caminho <- "codes/murrell"
> # Fig. 1.5 - mapa - packages: maps; maps
> if(!require("maps")) install.packages("maps")
> library(maps)
> source(file.path(caminho, "fig.1.5_examples-map.R"))
```



✓ Mapa da Nova Zelândia

✓ Conjunto de funções para produzir saídas gráficas primitivas



- Linhas, texto, figuras, etc.

✓ Pacotes:

- maps e mapproj.

15


Estatística Computacional I - 2020

• Alguns pacotes para construção de mapas:

✓ Pacote maps:

```
> # pacote maps
> library(maps)
> map('world', fill = TRUE, col = 1:10)
> map('world', fill = TRUE, col = 1:10, wrap = c(-180, 180))
```




16

Estatística Computacional I - 2020

✓ Pacote mapdata:

```
> library(mapdata)
> par(mar = c(1, 1, 1, 1))
> map("world", "Brazil", fill=T, col="grey90") # map("worldHires", "Brazil")
> map.axes()
> map.scale(ratio = F, cex = 0.7) #tentem ratio = T
> abline(h = 0, lty = 2)
```



✓ Construir passo a passo e observar os elementos que são acrescentados ao mapa

Estadística Computacional I - 2020

✓ Acrescentando países fronteiriços:

```
> # países fronteiriços
> library(mapproj)
> m <- map("world", "Brazil", fill=T, col="grey95")
> map(, , add=T)
> map.axes()
> map.scale(ratio = F, cex = 0.7)
> abline(h = 0, lty = 2)
> map.grid(m, nx = 5, ny = 5, col = "grey50", font = 1, cex = 0.7, pretty = T)
> #library(mapproj)
> map.cities(country = "Brazil", minpop = 2E6, pch = 19, cex = 1.2)
```




✓ Verificar possibilidades dos pacotes  
✓ E os estados?

Estadística Computacional I - 2020

✓ Acrescentar ao mapa aberto:

- Municípios com mais de 2 milhões de habitantes
  - Argumento minpop:

```
> # municípios com mais de 2.000 k habitantes
> map.cities(country = "Brazil", minpop = 2E6, pch = 19, cex = 1.2)
```



✓ Verificar outras populações

Estadística Computacional I - 2020

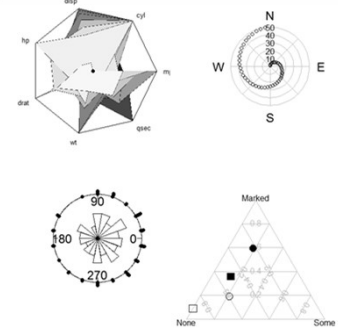
• Para iniciar a construir mapas no R:

- ✓ Introdução aos mapas em R
  - <http://rpubs.com/andreasancheztapia/176768>
- ✓ Pacote brazilmaps:

Estadística Computacional I - 2020

**Visualização de dados em outros sistemas**

```
> caminho <- "codes/murrell"
> # Fig. 1.6 - Coordenadas polares
> if(!require("CircStats")) install.packages("CircStats")
> if(!require("ggplotify")) install.packages("ggplotify")
> source(file.path(caminho, "fig.1.6_examples-polar.R"))
```

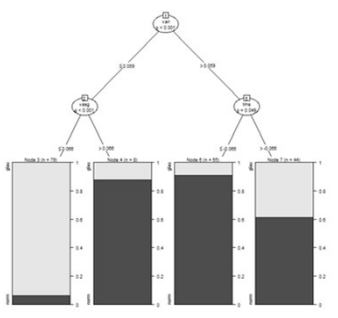


- ✓ Gráficos baseados em coordenadas polares
- ✓ Gráficos:
  - stars(graphics)
  - polar.plot(user)
  - rose.diag(CircStats)
  - ternaryplot(vcd)
- ✓ Pacotes:
  - CircStats, ggplotify e vcd.

Estadística Computacional I - 2020

**Gráficos especiais:**

```
> caminho <- "codes/murrell"
> # Fig. 1.7 - Árvores de decisão
> library("party")
> library("ipred")
> source(file.path(caminho, "fig.1.7_examples-recpart.R"))
```



- ✓ Árvore de decisão:
  - Visualização distribuição de variável dependente em cada nó
- ✓ Pacotes:
  - party e ipred.

Estadística Computacional I - 2020

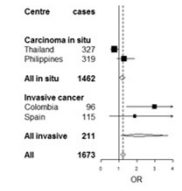
**Importante**

- Além do que são normalmente considerados gráficos estatísticos, é possível produzir imagens gráficas

Estadística Computacional I - 2020

**Produção de imagens gráficas:**

```
> caminho <- "codes/murrell"
> # Fig. 1.8 - Apresentação de arranjos tabulares
> source(file.path(caminho, "fig.1.8_examples-table.R"))
```



- ✓ Incorporação de arranjos tabulares de texto como elementos gráficos:
  - Maneira padrão de apresentar resultados de metanálise.
- ✓ Pacote:
  - grid.

Estadística Computacional I - 2020



## • Metanálise:

- ✓ Técnica estatística para combinar resultados provenientes de diferentes estudos.
- ✓ Exemplo na área da saúde:
  - Combinação do risco relativo entre dois tratamentos estimado em diferentes estudos.



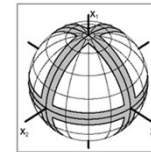
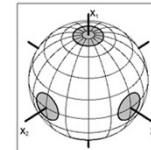
Estatística Computacional I - 2020

25



## • Gráficos especiais:

```
> caminho <- "codes/murrell"
> # Fig. 1.9 - Visualização de conceitos
> source(file.path(caminho, "fig.1.9_examples-arden.R"))
```



- ✓ Representação geométrica de extensões do teste F
- ✓ Gráfico usando o R base.

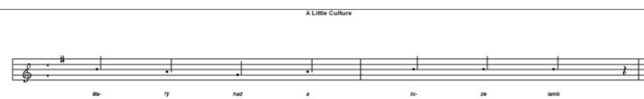
Estatística Computacional I - 2020

26



## • Gráficos especiais:

```
> caminho <- "codes/murrell"
> # Fig. 1.10 - pauta musical
> source(file.path(caminho, "fig.1.10_examples-stevemiller.R"))
```



- ✓ Pauta musical
- ✓ Figura usando o R base.



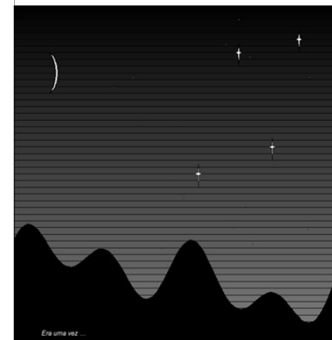
Estatística Computacional I - 2020

27



## • Produção de clip art:

```
> caminho <- "codes/murrell"
> # Fig.1.11 - clip art
> source(file.path(caminho, "fig.1.11_examples-once.R"))
```



- ✓ Pacote:
  - grid

Estatística Computacional I - 2020

28

## Organização dos Gráficos em R

Estrutura do Sistema Gráfico do R

- Níveis do sistema gráfico:
  - ✓ Pacotes gráficos:
    - graphics, maps
    - lattice, ggplot2
  - ✓ Sistemas gráficos:
    - Tradicional e grid
  - ✓ Dispositivos gráficos
    - Pacote grDevices
    - Manipulação cores, fontes, etc.
    - Saídas em diferentes formatos
  - ✓ Pacotes de dispositivos gráficos
    - Formatos adicionais de saídas gráficas

29

## Tipos de Funções Gráficas

- Funções de nível superior (*high-level*):
  - ✓ Produz gráficos completos
- Funções de nível inferior (*low-level*):
  - ✓ Adiciona saída em gráficos existentes
  - ✓ Trabalho interativo com a saída gráfica

30

## Pacotes gráficos:



- ✓ Maioria produz gráficos completos
- ✓ Em geral, oferecem gráficos especializados:
  - Análise ou campo de estudo específico
- ✓ Ex.:
  - maps.
  - hexbin.
  - scatterplot3d.

31

## Gráficos: Tradicionais vs. grid

- Sistemas gráficos em R:
  - ✓ Tradicional
  - ✓ grid.
- Comando único para construir gráfico:
  - ✓ Escolha do sistema depende muito do tipo de gráfico desejado.
  - ✓ Escolha do sistema pode ser irrelevante se não for necessário acrescentar nenhuma saída adicional ao gráfico.



32

- Sistema *grid*:
  - ✓ Amplas possibilidades para produzir gráficos a partir de uma página em branco
    - Necessário aprender alguns conceitos adicionais
  - ✓ Divide o gráfico em visualizações (camadas)
    - Cria objeto gráfico (*grob* – *graphical object*)
  - ✓ Objetos são manipulados para criar o gráfico
- Pacotes do sistema gráfico *grid*:
  - ✓ *lattice*
  - ✓ *ggplot2*
  - ✓ etc.

33

Estatística Computacional I - 2020






## Formatos de Saídas Gráficas

- É possível produzir arquivo com o *plot*.
- Vários formatos possíveis de saídas gráficas:
  - ✓ R dirige figura para dispositivo de saída:
    - Dispositivo necessita ser criado
    - Dispositivo necessita ser fechado após receber saída
      - Função `dev.off()`.

34

Estatística Computacional I - 2020



- Exemplos:
  - ✓ Arquivo *.pdf*:
 

```
> # Saídas
> # arquivo pdf
> pdf(file = "outputs\\meuPlot01.pdf")
> plot(pressure)
> dev.off()
windows
2
```
  - ✓ Arquivo *.jpg*:
 

```
> # arquivo jpg
> jpeg(file = "outputs\\meuPlot01.jpg")
> plot(pressure)
> dev.off()
windows
2
```
  - ✓ Verifique os arquivos gerados no subdiretório *outputs*.

35


Estatística Computacional I - 2020


- *Default* para janela gráfica do R:
  - ✓ Dispositivo *default* abre automaticamente em qualquer saída gráfica produzida.
    - `windows()`: Microsoft Windows
    - `x11()`: Unix
  - ✓ Função `options()`:
    - Usuário pode controlar o formato do dispositivo *default*.

36

Estatística Computacional I - 2020




## Dispositivos Gráficos




- Controle de dispositivos gráficos:
  - √ Tamanho da janela ou do documento
- É possível ter mais de um dispositivo aberto ao mesmo tempo, mas somente um deles está “ativo”
  - √ Todas saídas gráficas são enviadas para esse dispositivo

37

Estatística Computacional I - 2020




## • Algumas funções de controle:




- √ `dev.list()`:
  - Lista dispositivos gráficos abertos.
- √ `dev.cur()`:
  - Dispositivo atual ativo.
- √ `dev.set()`:
  - Ativa dispositivo.
- √ `dev.next()`:
  - Ativa o próximo dispositivo listado.
- √ `dev.prev()`:
  - Ativa o dispositivo listado anterior.
- √ `graphics.off()`:
  - Fecha todos os dispositivos gráficos.

38

Estatística Computacional I - 2020



## • Funções e formatos gráficos:




- √ Janela gráfica ou dispositivos GUI
  - GUI: *Graphical User Interface*:
    - Interface que permite a interação com dispositivos por meio de elementos gráficos  
(interface não se dá por linha de comando)


Função	Formato gráfico
<code>x11()</code> ou <code>X11()</code>	Janela X
<code>windows()</code>	Janela Microsoft Windows
<code>quartz()</code>	Janela Mac OS

39

Estatística Computacional I - 2020




## √ Dispositivos em arquivos (grDevices):




Função	Formato gráfico
<code>postscript()</code>	Arquivo Adobe PostScript
<code>pdf()</code>	Arquivo Adobe PDF
<code>pictext()</code>	Arquivo LaTeX PicTex
<code>xfig()</code>	Arquivo XFIG
<code>bitmap()</code>	Conversão GhostScript para arquivo
<code>png()</code>	Arquivo bitmap PNG
<code>jpeg()</code>	Arquivo bitmap JPEG
<i>Windows apenas</i>	
<code>win.metafile()</code>	Arquivo Metafile Windows
<code>bmp()</code>	Arquivo BMP Windows

40

Estatística Computacional I - 2020



## Saída com Múltiplas Páginas



- Funções `ps()` e `pdf()`:
  - ✓ Permitem múltiplas páginas
  - ✓ Argumento `onefile = TRUE`.


```
> # Sairas com múltiplas páginas
> TopDir <- getwd()
> pdf(file = file.path(TopDir, "outputs", "example.pdf"), onefile = TRUE)
> # grafico 1
> plot(pressure)
> # grafico 2
> plot(iris[-5])
> dev.off()
windows
2
```

✓ Verifique `example.pdf` no diretório especificado:

- Arquivo possui 2 gráficos

Estatística Computacional I - 2020

42



## ✓ Possível especificar armazenamento de cada saída gráfica em arquivos separados:

– Argumento `onefile = FALSE`.


```
> # Sairas armazenadas em arquivos separados
> pdf(file = file.path(TopDir, "outputs", "hist%02d.pdf"), onefile = F,
+ width = 6, height = 4, paper = 'special')
> for(i in 1:4){
+   # histograma de amostra selecionada
+   hist(amostra <- rnorm(100), freq = F, ylab = "Densidade",
+ xlab = "Dados", main = paste("Amostra # ", i))
+   # amostra selecionada salva em arquivo txt (diretório dados)
+   write.table(x = amostra,
+ file = file.path(TopDir, "dados", sprintf("amostra%02d.txt", i)),
+ sep = "\t", dec = ",")
+ }
> dev.off()
windows
2
```

✓ Padrão para nome do arquivo `.pdf`:


– file = `"hist.%02d"` (indica número da página)

Estatística Computacional I - 2020

43




## • Notação facilita salvar arquivos em *loop*:




- ✓ `%02d`: algarismos com dois dígitos
- ✓ Numeração do arquivo com os dados:
  - `sprintf("amostra.%02d.txt", i)`.

Estatística Computacional I - 2020

44



## Listas de Exibição



- R mantém lista de exibição para cada dispositivo gráfico aberto.
  - ✓ Usado para redesenhar a saída
    - Redimensionamento do dispositivo
    - Copiar saída de um dispositivo para outro

Estatística Computacional I - 2020

45



## • Comandos:

✓ `dev.copy()`:

- Copia saída do dispositivo ativo
- Cópia para outro dispositivo pode ser distorcida se razão altura/largura não for a mesma

✓ `dev.copy2eps()`:

- Preserva a razão de aspecto e cria arquivo em EPS
- Formato bom para incorporação em outros documentos (LaTeX)

✓ `device2bitmap()`:

- Preserva razão de aspecto em formato de saída via bitmap.



46

Estatística Computacional I - 2020



✓ `dev.print()`:

- Tenta imprimir saída no dispositivo ativo.
- Default: copia PS com comando de print dado por `options("printcmd")`.

✓ Lista de exibição pode aumentar consumo de memória:

- Gráfico complexo
- Muitos dispositivos abertos ao mesmo tempo

✓ É possível desativar a lista de exibição

- `dev.control(displaylist="inhibit")`.
- Saída não será redesenhada quando dispositivo for redimensionado
- Saída não poderá ser copiada entre dispositivos.



47

Estatística Computacional I - 2020



## • Comandos adicionais:

✓ `recordPlot()`:

- Salva gráfico atual em uma variável R

✓ `replayPlot()`:

- Reproduz gráfico armazenado em variável R



48

Estatística Computacional I - 2020



## • Lista de exibição com 3 gráficos:

```
> # Listas de exibição de saídas gráficas
> num.plots <- 3
> # lista para armazenar 3 gráficos
> meus.plots <- vector(num.plots, mode = "list")
>
> # 1o. gráfico
> plot(x = mtcars$wt, y = mtcars$mpg,
+      pch = 16, frame = FALSE,
+      xlab = "wt", ylab = "mpg", col = "#2E9FDF")
> abline(lm(mpg ~ wt, data = mtcars), lty = 2, col = "red")
> title("Linha de Regressão")
> meus.plots[[1]] <- recordPlot()
>
> # 2o. gráfico
> plot(iris[-5])
> meus.plots[[2]] <- recordPlot()
>
> # 3o. gráfico
> boxplot(Petal.Width ~ Species, data = iris)
> meus.plots[[3]] <- recordPlot()
>
> # fecha dispositivos gráficos
> graphics.off()
```



49

Estatística Computacional I - 2020

✓ Recuperação de lista de exibição:

```
> # recupera gráficos
>
> meus.plots
```



50

Estatística Computacional I - 2020

✓ Gravação de *plots* da lista em arquivo .pdf:

- Argumento `onefile = FALSE`.

```
> pdf(file = file.path(TopDir, "outputs", "meusPlots.pdf"), onefile = TRUE)
> for (meus.plots in meus.plots) {
+   replayPlot(meus.plots)
+ }
> dev.off()
windows
2
```

✓ Verificar arquivo `meusPlots.pdf`: no diretório desejado.

51

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.

244

Estatística Computacional I - 2020