

Visualização de dados

97

Gráficos no R

Criando um gráfico de barras

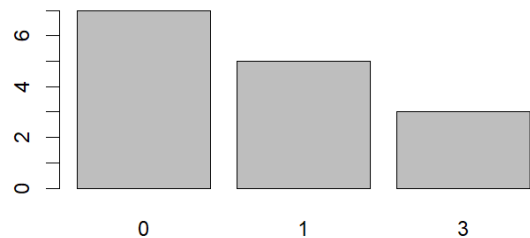
Usamos a função `barplot()` do R. Mas, antes de usar a função precisamos ter o dado em um formato de tabela com contagem de frequências

Exemplo:

`sexo = c(0,0,0,0,0,0,0,1,1,1,1,1,3,3,3)` *#vetor que armazena a informação do sexo*

`freq = table(sexo)` *#função que tabula a contagem de classes do vetor*

`barplot(freq)`



Caso seja interessante plotar a frequência relativa em vez de contagem, use a função `prop.table()` sobre o objeto `freq` criado

DICA DE OURO!
Veja alguns exemplos de gráficos com o comando `demo(graphics)`

98

Gráficos no R

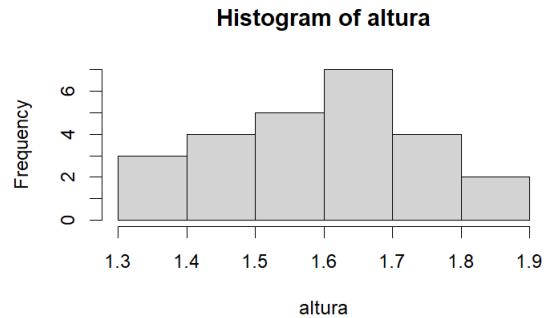
Criando um histograma

Usamos a função `hist()` do R. Note que histograma não é o mesmo que um gráfico de barras...

Exemplo:

`altura = c(1.3, 1.33, 1.4, 1.42, 1.45, 1.5, 1.5, 1.52, 1.55, 1.56, 1.58, 1.6, 1.62, 1.64, 1.65, 1.67, 1.68, 1.68, 1.7, 1.72, 1.72, 1.74, 1.8, 1.83, 1.85)` *#vetor que armazena a informação de altura*

`hist(altura)`



Com o argumento `nclass` = determinamos o número de classes que o histograma terá. Por outro lado, com `breaks=c()` é possível definir o limite de cada classe, para isso basta determinar os valores limites das classes. Ex: `breaks = c(1.3, 1.45, 1.6, 1.75, 1.9)`

99

Gráficos no R

Criando um boxplot

Usamos a função `boxplot()` do R. Caso seja interessante apresentar por grupos, usamos o complemento `~` (lê-se: "de acordo com") indicando o vetor do grupo

Exemplo:

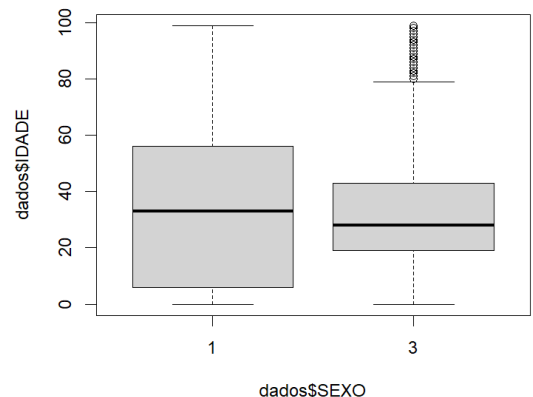
`dados$idade`

`1: [22,24,24,75...] #vetor que armazena a informação de idade`

`dados$sexo`

`2: [1,1,1,1,1,3,3...] #vetor que armazena a informação de idade`

`boxplot(dados$idade ~ dados$sexo)`



100

Gráficos no R

Criando gráficos de dispersão ou linhas

A função genérica `plot()` é usada para gráficos de dispersão

Exemplo:

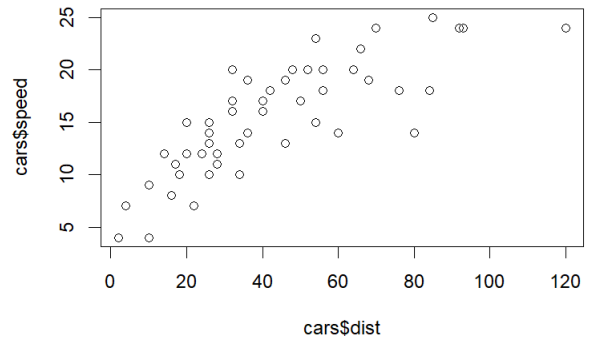
`cars$dist`

1: [2, 10, 4, 22, 16, 18 ...] *#vetor que armazena a informação de distância*

`cars$speed`

2: [4, 4, 7, 7, 8, 9, 10, 10...] *#vetor que armazena a informação de velocidade*

`plot(cars$speed ~ cars$dist)`



Para decidir se um gráfico é de dispersão ou de linhas, o argumento `'type='` deve ser usado:

- "p" para pontos (esse é o default)
- "l" para linhas
- "b" para ambos.

101

Gráficos no R

Formatando os eixos

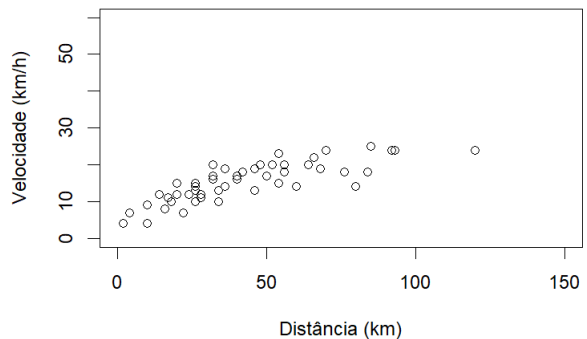
Títulos: Para o título central usamos o argumento `main` e os argumentos `xlab` e `ylab` para os títulos dos eixos

Limites: Usamos os argumentos `xlim` e `ylim` para definir os títulos dos eixos

Exemplo:

```
plot(cars$speed ~ cars$dist,
     main = "Velocidade Vs. Distância",
     xlab = "Distância (km)", ylab = "Velocidade
(km/h)")
xlim=c(0,150), ylim=c(0,60)
)
```

Velocidade Vs. Distância



Os valores do rótulo do eixo geralmente são melhor apresentados horizontalmente. Isso pode ser ajustado usando `las = 1`. Outras possibilidades:

- 0: sempre paralelo ao eixo (padrão)
- 1: sempre horizontal
- 2: sempre perpendicular ao eixo
- 3: sempre na vertical.

102

Gráficos no R

Alterando os formatos dos pontos

Tipo: O padrão de pontos pode ser alterado usando o argumento `pch =` na função `plot()`. Podemos explicitar caracteres, como `pch="."` e `pch="o"`, ou adotar algum dos padrões disponíveis, como `pch = 19`

Tamanho: O argumento `cex` ("expansão de caracteres") pode ser empregado e seu valor padrão de 1,0 alterado para aumentar ou diminuir os tamanhos dos pontos. Como `cex=2` (o dobro) ou `cex=0.5` (a metade)

Exemplo:

```
plot(cars$speed ~ cars$dist,
      xlab="Distância (km)", ylab="Velocidade (km/h)",
      xlim=c(0,150), ylim=c(0,60),
      pch = 19,
      cex = 0.5
)
```



DICA DE OURO!

O argumento `cex` também altera o tamanho dos títulos. Ex: `cex.main`, `cex.lab`.

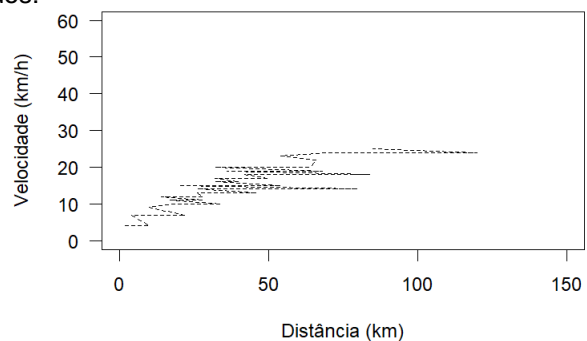
103

Gráficos no R

Alterando os formatos das linhas

Para alterar uma linha plotada de sólida (o padrão) para tracejada ou pontilhada, o argumento `lty =` pode ser usado, com os seguintes valores associados:

- 1: sólido (padrão)
- 2: tracejado
- 3: pontilhado
- 4: ponto-traço
- 5: traço longo
- 6: traço longo, traço curto.



104

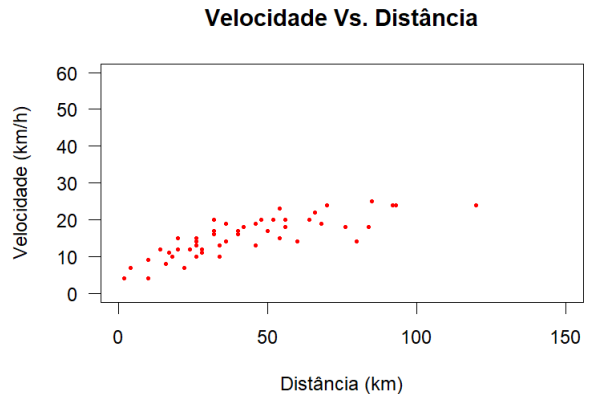
Gráficos no R

Alterando as cores

Usamos o atributo `col =`. As cores podem ser especificadas explicitamente pelo nome (ex: "red"), pelo valor hexadecimal (`#FF0000`) ou pelas diversas paletas de cores (`rainbow()`, `heat.colours()` etc.)

Exemplo:

```
plot(cars$speed ~ cars$dist,
      xlab="Distância (km)", ylab="Velocidade
      (km/h)",
      xlim=c(0,150), ylim=c(0,60),
      pch = 19,
      cex = 0.5,
      col = "red"
)
```



DICA DE OURO!
Para exibir os nomes das 657 cores padrão internas ao R, digite: `colours()`

105

Gráficos no R

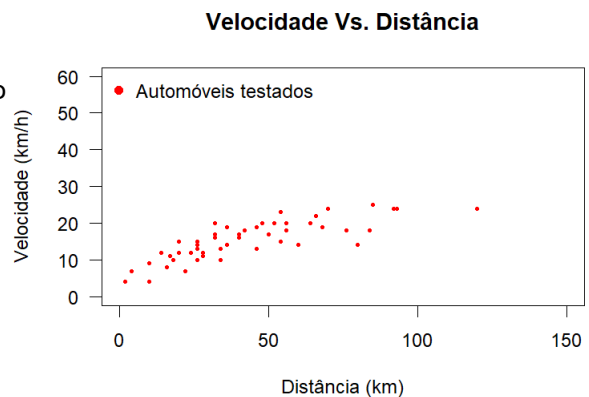
Adicionando uma legenda

A função `legend()` é um comando de plotagem de "baixo nível". É implementado após a construção do gráfico.

Exemplo:

```
legend("topleft",
      c("Automóveis"),
      pch=c(19),
      col=c("red"),
      bty="n")
)
```

O uso de `bty="n"` suprime o desenho de uma caixa ao redor da legenda



DICA DE OURO!
Consulte `?legend` para mais opções de posição de legenda

106

Gráficos no R

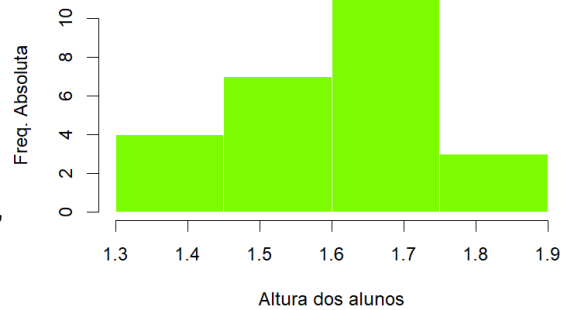
Exportando o gráfico

1. Crie um arquivo vazio jpeg
`jpeg("rplot.jpg", width = 350, height = 350)`

2. Crie o gráfico
`hist(altura,`
`main = "Altura dos alunos do 1º ano do Ensino`
`Médio",`
`xlab = "Altura dos alunos", ylab = "Freq. Absoluta",`
`col = "lawngreen",`
`border = FALSE,`
`breaks = c(1.3, 1.45, 1.6, 1.75, 1.9),`
`right = FALSE)`

3. Feche o arquivo
`dev.off()`

Altura dos alunos do 1º ano do Ensino Médio



Outras opções de formatos:

- `pdf("rplot.pdf")`: arquivo pdf
- `png("rplot.png")`: arquivo png
- `jpeg("rplot.jpg")`: arquivo jpeg
- `postscript("rplot.ps")`: arquivo postscript
- `bmp("rplot.bmp")`: arquivo bmp
- `win.metafile("rplot.wmf")`: meta-arquivo do Windows

107

Tidyverse

Packages Blog Learn Help Contribute



Pacote para visualização de dados

R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

<https://www.tidyverse.org>

108

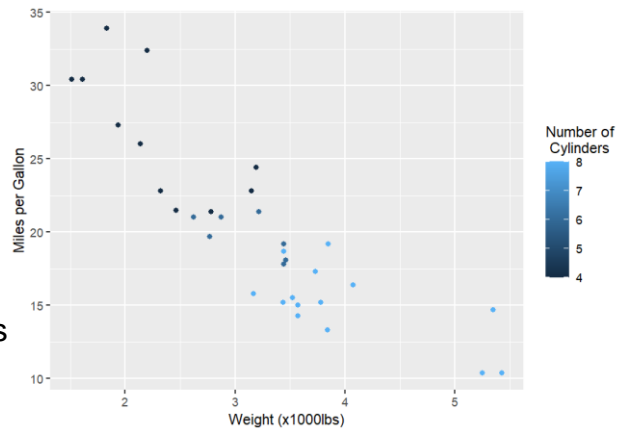
Gráficos no R

Alternativa com GGLOT2

O ggplot2 é um pacote próprio de visualização, com uma síntese de camadas:

Exemplo:

```
ggplot(mtcars,
       aes(x=wt, y=mpg, col=cyl)) +
  geom_point() +
  labs(x="Weight (x1000lbs)", y="Miles
per Gallon", colour="Number of
Cylinders")
```



109

The R Graph Gallery

<https://r-graph-gallery.com/>



Q CHART TYPES QUICK TOOLS ALL RELATED ABOUT



Violin



Density



Histogram



Boxplot



Ridgeline

Correlation



Scatter



Heatmap



Correlogram



Bubble



Connected scatter



Density 2d

110

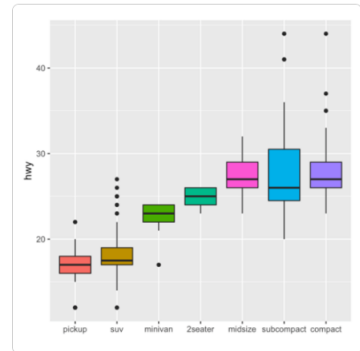
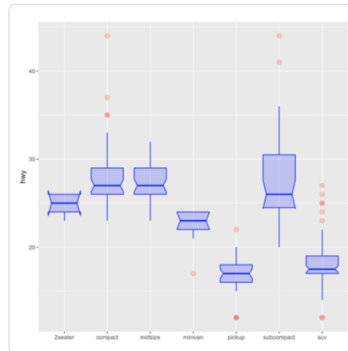
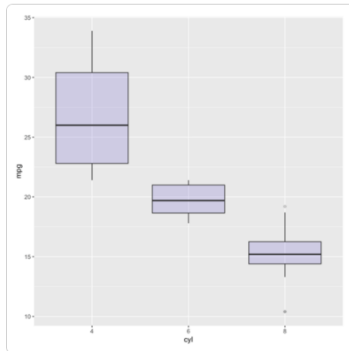
The R Graph Gallery



<https://r-graph-gallery.com/>

GGPLOT2

Boxplot are built thanks to the `geom_boxplot()` geom of `ggplot2`. See its basic usage on the [first example](#) below. Note that [reordering groups](#) is an important step to get a more insightful figure. Also, showing individual data points with [jittering](#) is a good way to avoid hiding the underlying distribution.



111

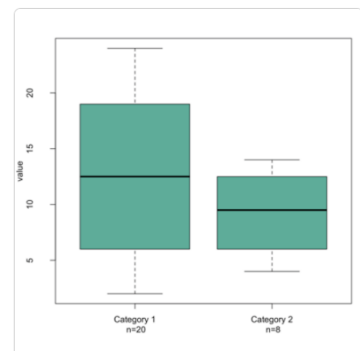
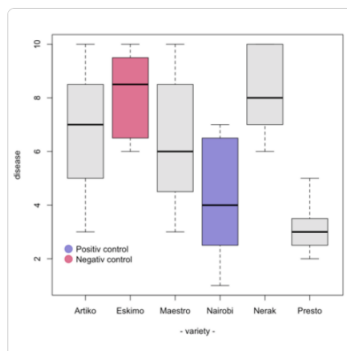
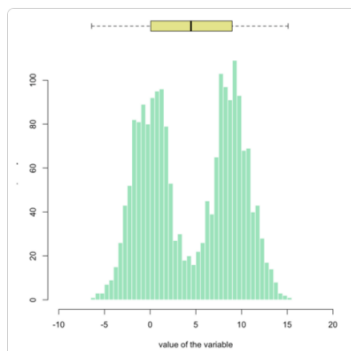
The R Graph Gallery



<https://r-graph-gallery.com/>

BASIC R

Build boxplot with base R is totally doable thanks to the `boxplot()` function. Here are a few examples of its use:



112

The R Graph Gallery

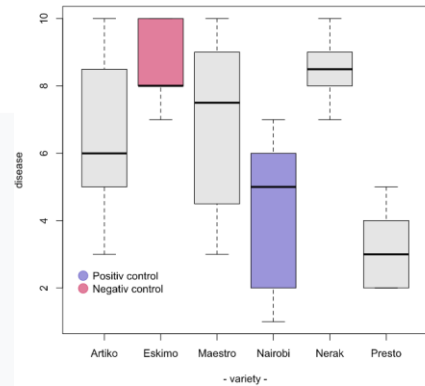

<https://r-graph-gallery.com/>

```
#Create data
names <- c(rep("Maestro", 20), rep("Presto", 20),
  rep("Nerak", 20), rep("Eskimo", 20), rep("Nairobi", 20), rep("Artiko", 20))
value <- c( sample(3:10, 20, replace=T), sample(2:5, 20, replace=T),
  sample(6:10, 20, replace=T), sample(6:10, 20, replace=T),
  sample(1:7, 20, replace=T), sample(3:10, 20, replace=T) )
data <- data.frame(names,value)

# Prepare a vector of colors with specific color for Nairobi and Eskimo
myColors <- ifelse(levels(data$names)=="Nairobi", rgb(0.1,0.1,0.7,0.5),
  ifelse(levels(data$names)=="Eskimo", rgb(0.8,0.1,0.3,0.6),
    "grey90" ) )

# Build the plot
boxplot(data$value ~ data$names,
  col=myColors,
  ylab="disease", xlab="- variety -")

# Add a legend
legend("bottomleft", legend = c("Positiv control", "Negativ control"),
  col = c(rgb(0.1,0.1,0.7,0.5), rgb(0.8,0.1,0.3,0.6)), bty = "n", pch=20, pt.cex = 3, cex = 1, horiz = FALSE, inset = c(0.03, 0.1))
```



113

plotly | Graphing Libraries

Search...

▼ Quick Reference

- Getting Started
- Is Plotly Free?
- Figure Reference
- ggplot2 integration
- Dash for R
- GitHub
- community.plotly.com

▼ Examples

- Fundamentals
- Basic Charts
- Statistical Charts
- Scientific Plots

[R](#)
[Forum](#)
[Pricing](#)
[Dash](#)
[Star](#)
2,250

Plotly R Open Source Graphing Library

Plotly's R graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, and 3D (WebGL based) charts.

Plotly.R is [free and open source](#) and you can [view the source](#), [report issues](#) or [contribute on GitHub](#).

Deploy R AI Dash apps on private Kubernetes clusters: [Pricing](#) | [Demo](#) | [Overview](#) | [AI App Services](#)

Fundamentals

The Figure Data Structure

Creating and Updating Figures

Displaying Figures

Exporting Graphs as Static Images

Configuration

More Fundamentals »

Basic Charts

More Basic Charts »

A bi-monthly newsletter from Chris Parmer & Adam Schroeder

[Join Basic Club](#)

<https://plotly.com/r/>

114

Gráficos no R

Tornando o gráfico interativo com **PLOTLY**

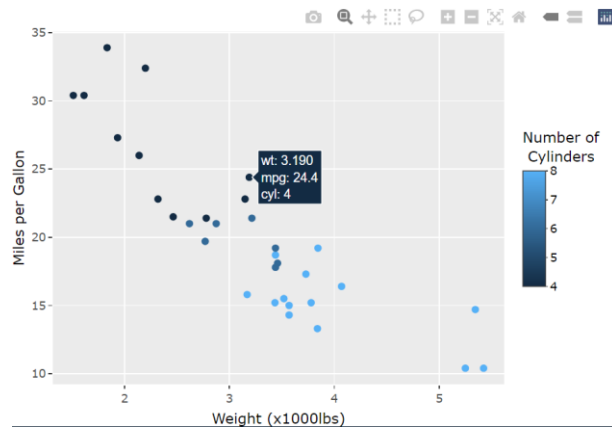
Os objetos **ggplot2** podem ser tornados interativos com o pacote **"plotly"**:

Exemplo:

```
install.packages("plotly")
```

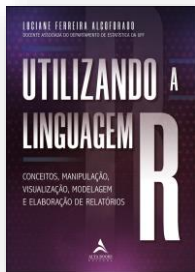
```
library(plotly)
```

```
figura = ggplot(mtcars,
  aes(x=wt, y=mpg, col=cyl)) +
  geom_point() +
  labs(x="Weight (x1000lbs)", y="Miles per
  Gallon", colour="Number of n Cylinders")
ggplotly(figura)
```

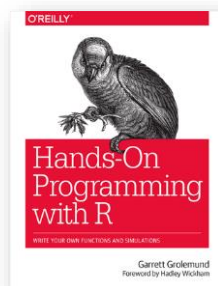


115

Algumas referências



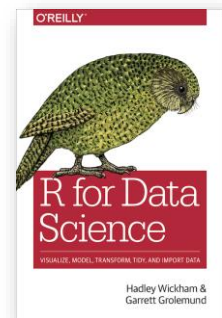
- Luciane Ferreira Alcoforado
- Pago
- <https://altabooks.com.br/produto/utilizando-a-linguagem-r>



- Garrett G. Grolemund.
- Gratuito
- <https://rstudio-education.github.io/hopr>



- Oliveira et al.
- Gratuito
- <https://www.ibpad.com.br>



- Wickham & Grolemund
- Gratuito
- <https://r4ds.had.co.nz>

116

Referências

Alcoforado LF. Utilizando A Linguagem R: Conceitos, manipulação, visualização, modelagem e elaboração de relatórios. Rio de Janeiro: Alta Book, 2021. 367p

Figueira L. Criando funções no R. Estatística é com R! Disponível em: <http://www.estadisticacomr.uff.br/?p=224>

Grolemund G. Hands-On Programming with R. O'Reilly Media; 1ª edição, 2014. Disponível em: <https://rstudio-education.github.io/hopr>

Wickham H. Advanced R. CRC Press; 1ª edição, 2014. Disponível em: <http://adv-r.had.co.nz/>

117

117



Universidade de Brasília
Faculdade de Ciências da Saúde
Departamento de Saúde Coletiva

Obrigado

Ivan Zimmermann
ivan.zimmermann@unb.br

118

118