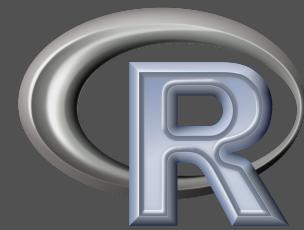




ME115 - Linguagem R

Parte 10

1º semestre de 2023



Visualização de Dados

Visualização de Dados

A representação gráfica desempenha um papel fundamental em qualquer análise estatística.

Os gráficos constituem uma das formas mais eficientes de apresentação de dados.

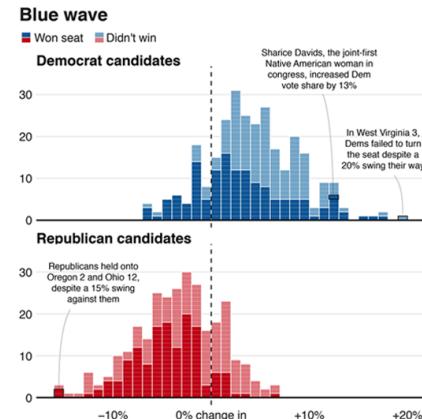
Eles permitem uma visão mais rápida e fácil das variáveis às quais se referem.

A qualidade na representação gráfica deve ser pautada na clareza, simplicidade e auto explicação.

O gráfico a ser usado depende do tipo de variável (categórica ou contínua) e da quantidade de variáveis que estamos considerando.

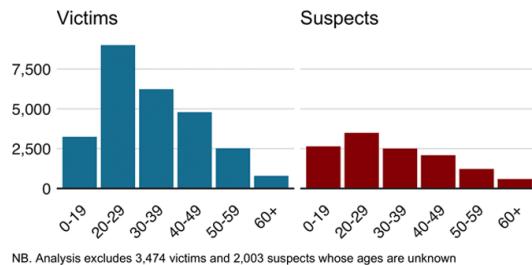


Visualização de Dados

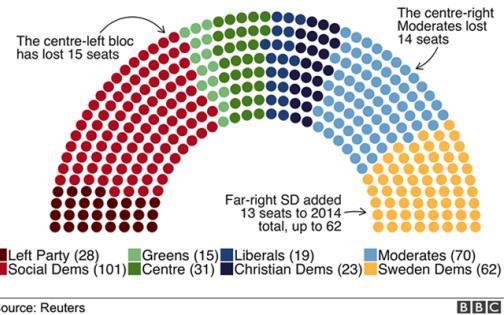


Homophobic hate crimes are mainly committed by young people on young people

Number in each age group 2014 - 2017



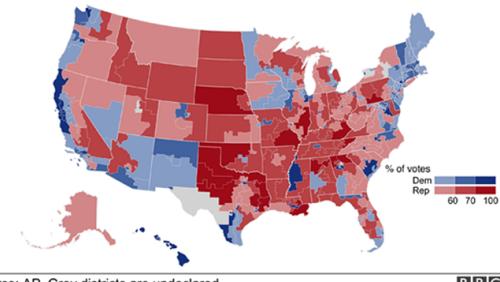
Results of the 2018 election



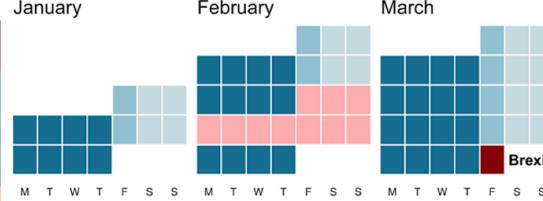
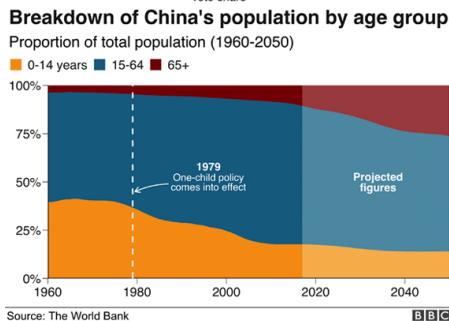
BBC

Democrats take the House

Dem 232 218 to win Rep 198



BBC



BBC Source: AP. Grey districts are undeclared

Fonte: How the BBC Visual and Data Journalism team works with graphics in R

Tipos de Variáveis

O gráfico apropriado depende do tipo e do número de variáveis de interesse.

Variável: Qualquer característica associada a uma população

Classificação das variáveis

- **Qualitativa:** são aquelas que apresentam como possíveis realizações uma qualidade ou atributo do indivíduo pesquisado.
 - Nominal: sexo, cor dos olhos
 - Ordinal: classe social, grau de instrução
- **Quantitativa:** são aquelas que apresentam como possíveis realizações números resultantes de uma contagem ou mensuração.
 - Contínua: peso, altura
 - Discreta: número de filhos, número de carros

Visualização de Uma Única Variável

A visualização da distribuição de uma variável depende se ela é contínua ou categórica.

Variáveis contínuas: podem assumir um conjunto infinito de valores. Exemplos: Números e data-hora.

Os gráficos mais utilizados nesse caso são o **histograma** e **boxplot**.

Variáveis categóricas: podem assumir um conjunto finito de valores. Geralmente salvas como vetores de caracteres ou fatores no R.

O gráfico mais utilizado nesse caso é o **gráfico de barras**.



Gráficos na Base do R

Antes de aprendermos como criar gráficos usando o `ggplot`, iremos criar os principais tipos de gráficos usando as funções da base do R.

Como exemplo, iremos trabalhar com os dados `mtcars`.

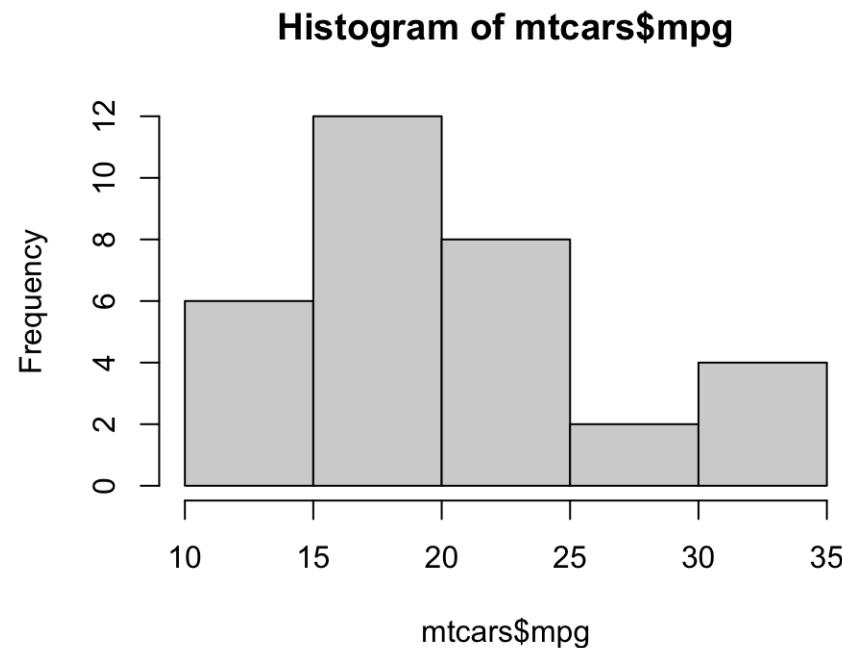
```
data(mtcars)  
head(mtcars)
```

```
##          mpg cyl disp  hp drat    wt  qsec vs am gear carb  
## Mazda RX4     21.0   6 160 110 3.90 2.620 16.46  0  1    4    4  
## Mazda RX4 Wag 21.0   6 160 110 3.90 2.875 17.02  0  1    4    4  
## Datsun 710    22.8   4 108  93 3.85 2.320 18.61  1  1    4    1  
## Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0    3    1  
## Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02  0  0    3    2  
## Valiant      18.1   6 225 105 2.76 3.460 20.22  1  0    3    1
```

Histograma

Para criar um histograma, usamos a função `hist()`:

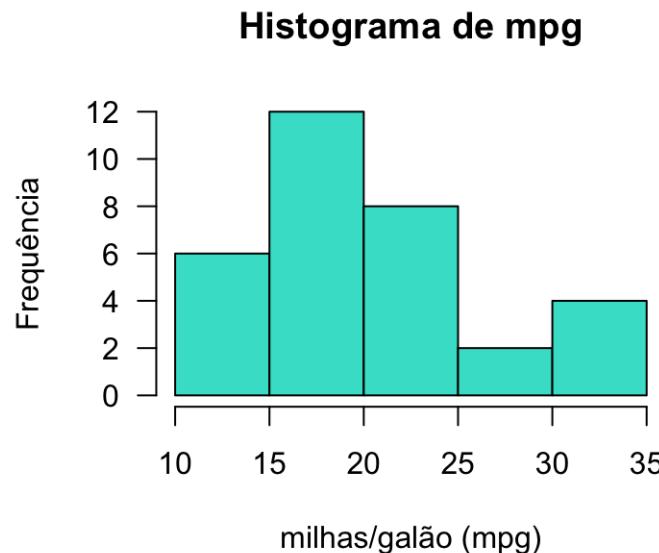
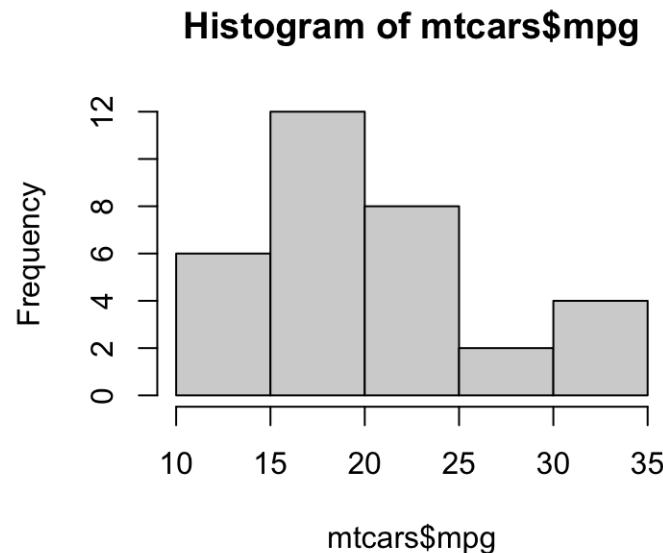
```
hist(mtcars$mpg)
```



Histograma

Alguns argumentos de `hist()` ajudam a melhorar a aparência do gráfico:

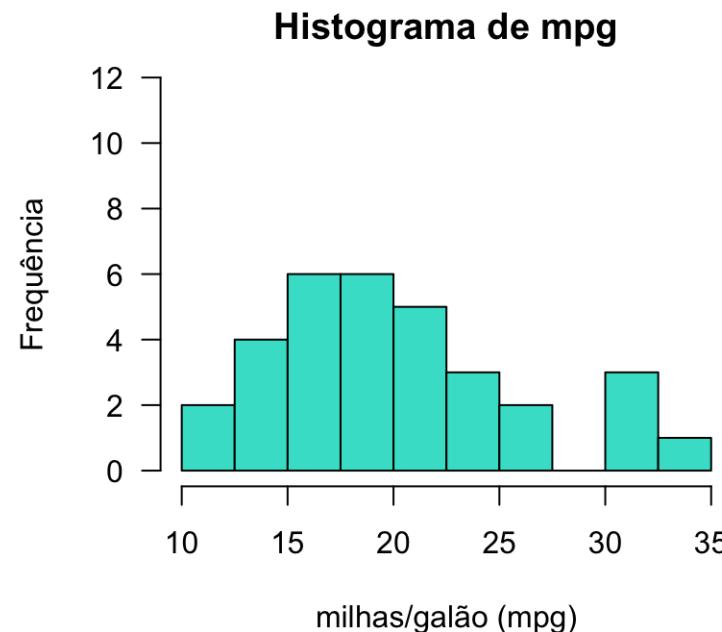
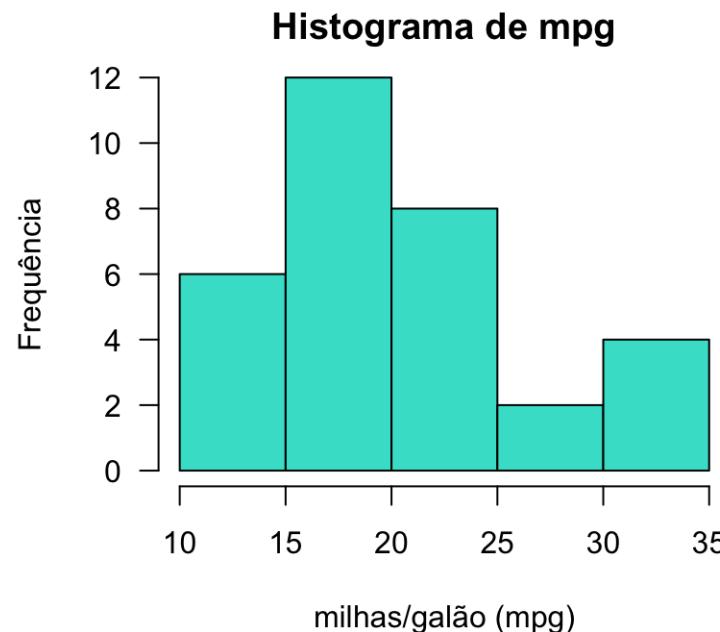
```
hist(mtcars$mpg) # histograma padrão  
hist(mtcars$mpg, col = "turquoise", las = 1,  
     main = "Histograma de mpg", xlab = "milhas/galão (mpg)", ylab = "Frequência")
```



Histograma - Número de Intervalos

Para alterar o número de intervalos, especifique o argumento `breaks`:

```
hist(mtcars$mpg, breaks = seq(10, 35, 2.5)) # especificar os limites  
hist(mtcars$mpg, breaks = 10) # fixar o número de intervalos
```



Histograma

Usando o histograma, podemos descrever uma série de características da variável sendo estudada:

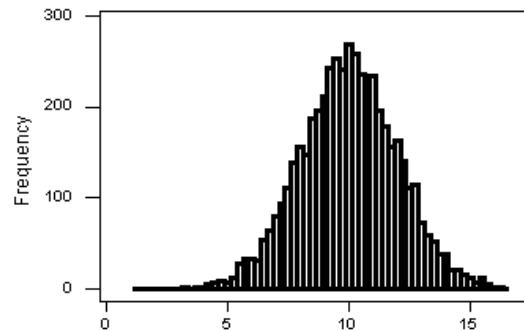
- **Padrão geral:** forma, centro, dispersão;
- **Desvio do padrão:** outliers.

Ao descrever a forma de uma distribuição, você deve considerar:

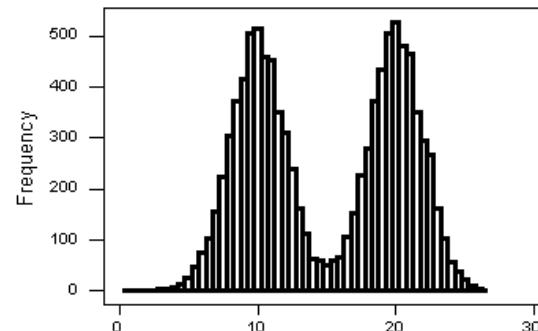
- **Simetria:** simétrica, assimétrica (à direita, à esquerda).
- **Modalidade:** número de picos que a distribuição tem.
- **Posição:** medidas do centro da distribuição.
- **Dispersão:** amplitude dos dados e dispersão em relação ao centro.
- **Outliers:** observações fora do padrão.

Histograma

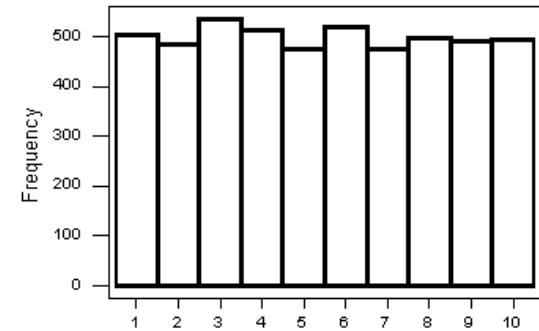
Symmetric, Single-peaked (Unimodal) Distribution



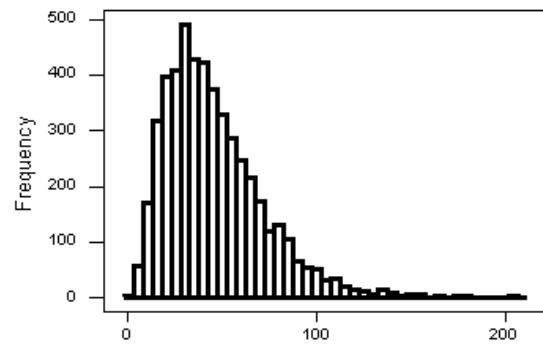
Symmetric, Double-peaked (Bimodal) Distribution



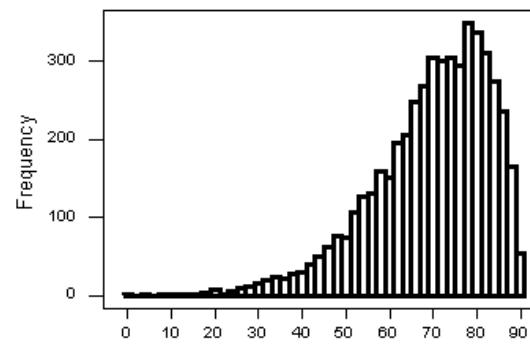
Symmetric, Uniform, Distribution



Skewed-Right Distribution

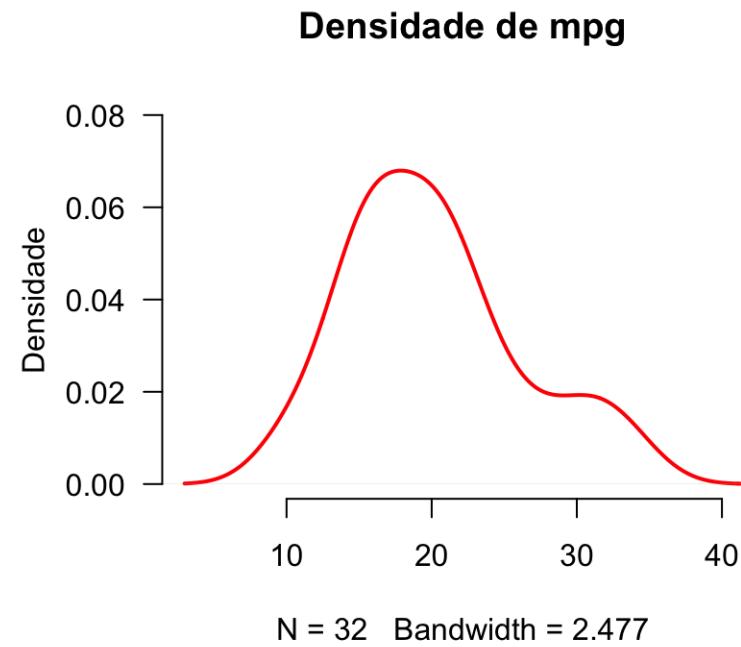
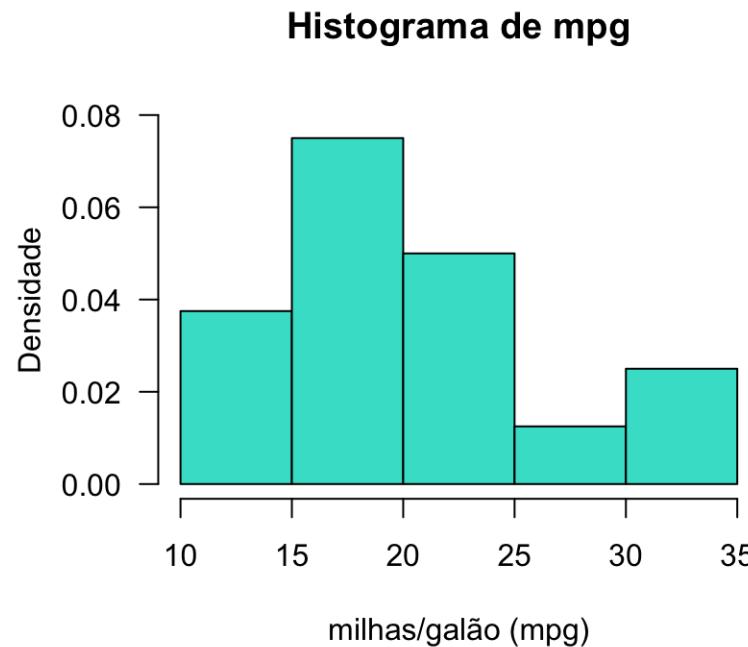


Skewed-Left Distribution



Histograma e Gráfico de Densidade

```
dens <- density(mtcars$mpg)
plot(dens, col = "red", frame = FALSE, las = 1, lwd = 2,
     main = "Densidade de mpg", ylab = "Densidade")
```



Histograma e Gráfico de Densidade

Para adicionar a densidade no histograma, use a função `lines()`:

```
hist(mtcars$mpg, probability = TRUE, col = "turquoise", las = 1)
lines(dens, col = "red", lwd = 2)
```

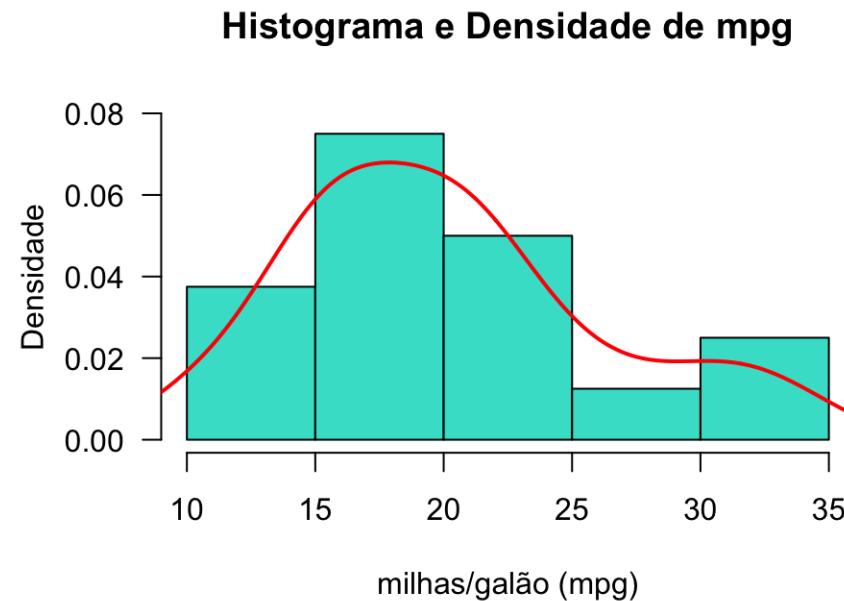
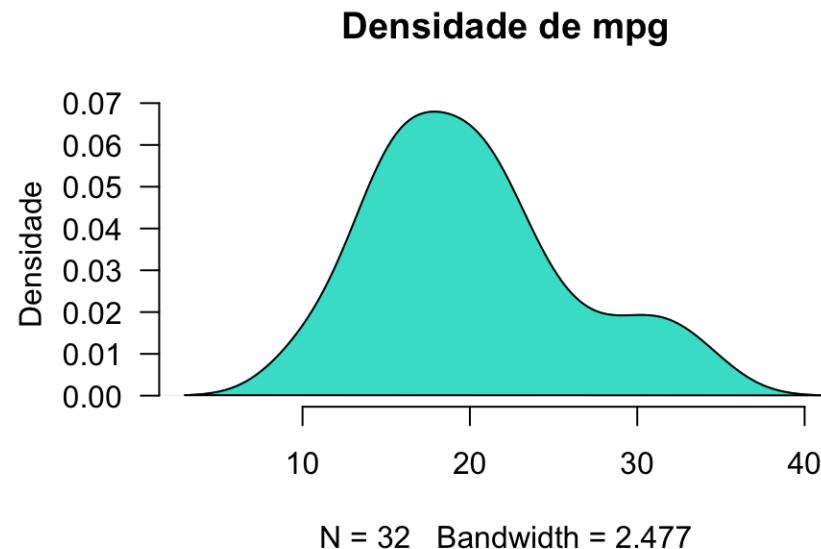


Gráfico de Densidade

Para preencher a densidade, use a função `polygon()`:

```
plot(dens, frame = FALSE, las = 1, col = "steelblue",
      main = "Densidade de mpg", ylab = "Densidade")
polygon(dens, col = "turquoise")
```



Boxplot

Boxplot é a representação gráfica do *five-number summary*.
Na base do R, a função usada é `boxplot()`:

```
boxplot(mtcars$mpg, horizontal = TRUE, col = "turquoise",
        main = "Boxplot de mpg", xlab = "mpg")
```

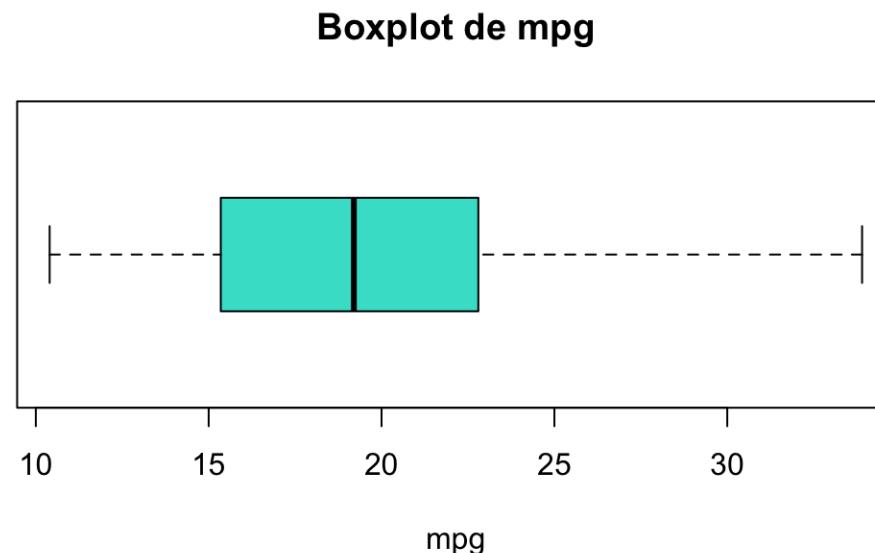
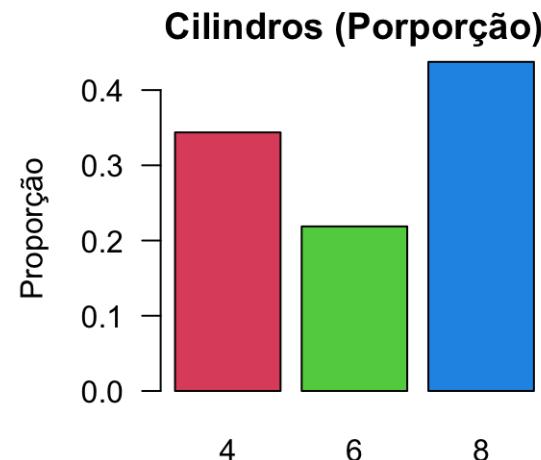
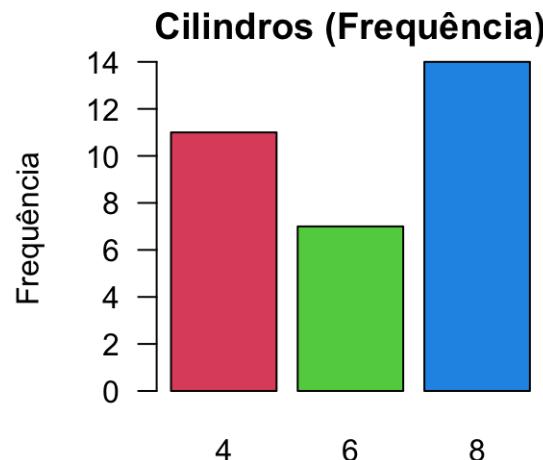


Gráfico de Barras

O gráfico de barras representa a frequência (ou proporção) que cada categoria de uma variável categórica ocorre. Na base do R, a função é `barplot()`:

```
contagem <- table(mtcars$cyl)
barplot(contagem, col = 2:4, las = 1, ylab = "Frequência")
```

```
proporcao <- prop.table(contagem)
barplot(proporcao, col = 2:4, las = 1, ylab = "Proporção")
```



Relação entre Duas Variáveis

É comum querermos analisar a relação entre duas variáveis. Podemos ter as seguintes combinações:

- Duas variáveis contínuas
- Uma variável contínua e uma categórica
- Duas variáveis categóricas

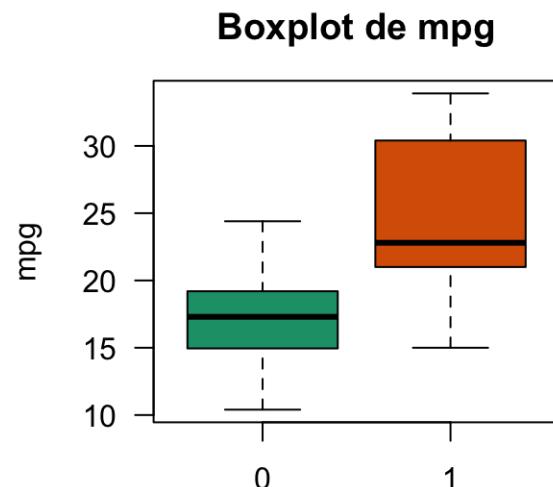
Assim como no caso univariado, para cada combinação de variáveis, existem gráficos apropriados.

"The simple graph has brought more information to the data analyst's mind than any other device." — John Tukey

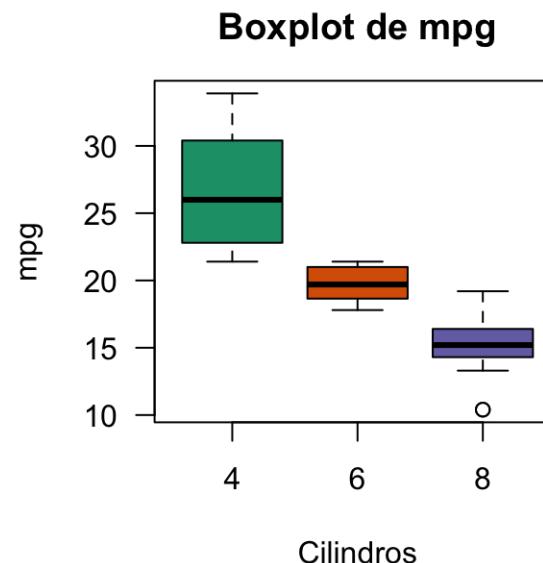
Uma variável contínua e uma categórica

Queremos avaliar a distribuição de uma variável contínua de acordo com as categorias de uma variável categórica. Nesse caso, podemos apresentar boxplots por grupos/categorias:

```
boxplot(mpg ~ am, data = mtcars, col = c("#1B9E77", "#D95F02"))
```



Transmissão (0 = automatic, 1 = manual)

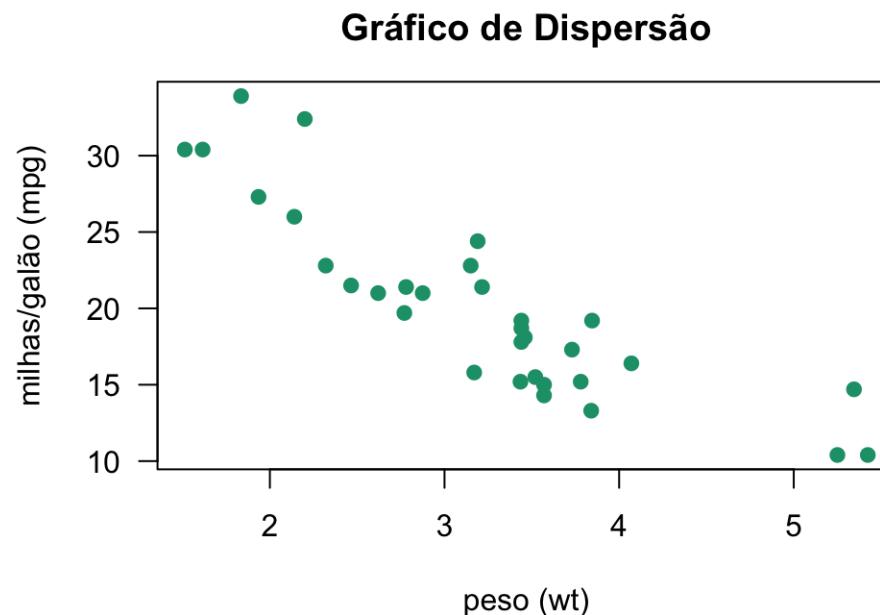


Cilindros

Duas variáveis contínuas

Quando temos duas variáveis contínuas, o gráfico mais comum é o gráfico de dispersão (*scatterplot*). Na base do R, a função é simplesmente `plot()`:

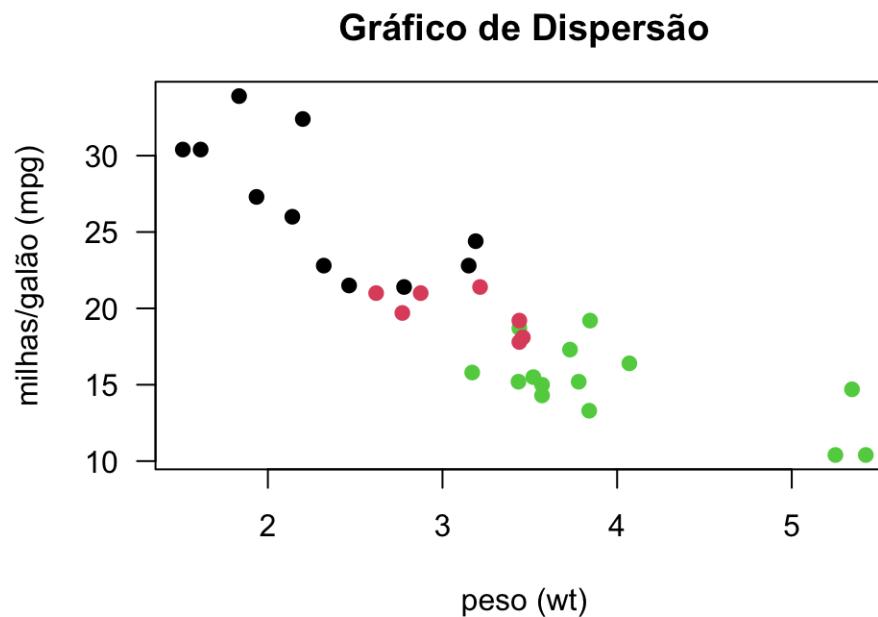
```
plot(mpg ~ wt, data = mtcars, col = "#1B9E77", pch = 19)
plot(x = mtcars$wt, y = mtcars$mpg)
```



Duas variáveis contínuas

Podemos colorir os pontos do gráfico de dispersão de acordo com uma variável categórica:

```
plot(mpg ~ wt, data = mtcars, col = as.factor(cyl), pch = 19, las = 1)
```



Parâmetros Gráficos

Você pode mudar os parâmetros gráficos como símbolo e tipo de linha através dos argumentos `pch` (figura à esquerda) e `lty` (figura à direita):

0	1	2	3	4
□	○	△	+	×
5	6	7	8	9
◇	▽	▣	*	◊
10	11	12	13	14
⊕	⊗	田	⊗	□
15	16	17	18	19
■	●	▲	◆	●
20	21	22	23	24
●	●	■	◆	▲
25				

6.'twodash'	- - - - - - -
5.'longdash'	- - - - - - - -
4.'dotdash'	- - - - - - - - -
3.'dotted'	- - - - - - - - - -
2.'dashed'	- - - - - - - - - -
1.'solid'	—
0.'blank'	

Fonte: [STHDA - Graphical parameters](#)

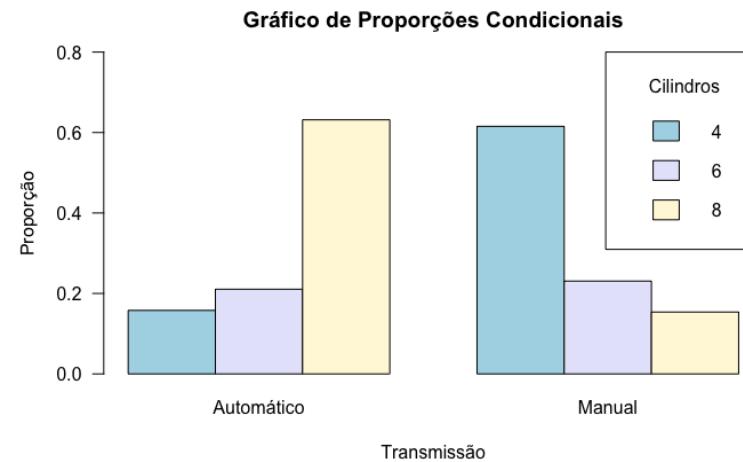
Duas variáveis categóricas

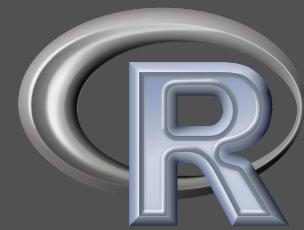
Para duas variáveis categóricas, podemos fazer um gráfico de barras representando uma tabela de contingência de proporções condicionais.

```
tbla <- prop.table(table(mtcars$cyl, mtcars$am, dnn = c("cyl", "am")), margin = 2)

barplot(tbla, beside = TRUE, col = c("lightblue", "lavender", "cornsilk"),
        names.arg = c("Automático", "Manual"), legend = TRUE)
```

```
##      am
## cyl    0     1
##   4  0.16  0.62
##   6  0.21  0.23
##   8  0.63  0.15
```





Visualização de Dados com ggplot2

Visualização de Dados com `ggplot2`

- O pacote `ggplot2` é o que temos hoje de mais moderno e o que revolucionou a visualização de dados no R.
- Escrito por Hadley Wickham, fruto de sua tese de doutorado.
- Baseado na definição da Gramática dos Gráficos (*Grammar of Graphics*).
- **Gramática dos Gráficos** (Leland Wilkinson, 2005): um gráfico é o mapeamento dos dados em atributos estéticos (posição, cor, forma, tamanho) de formas geométricas (pontos, linhas, barras, caixas).
- A partir dessa definição, Hadley escreveu [A Layered Grammar of Graphics](#).



ggplot2

Uma gramática dos gráficos em camadas: os elementos de um gráfico (dados, sistema de coordenadas, rótulos, anotações, entre outros) são as suas camadas e a construção de um gráfico se dá pela sobreposição dessas camadas.

Algumas vantagens do `ggplot2` em relação aos gráficos da base do R:

- gráficos naturalmente mais bonitos;
- fácil personalização;
- a estrutura padronizada das funções deixa o aprendizado muito mais intuitivo;
- a diferença no código entre tipos diferentes de gráficos é muito pequena.



Componentes dos gráficos no `ggplot2`

Todo gráfico em `ggplot2` tem três componentes principais:

1. **dados**;
2. **estética de mapeamento (`aes()`)** entre variáveis presentes em dados e propriedades visuais;
3. **geometria**: camadas criadas com as funções `geom_*` que indicam o tipo de gráfico.

A função `ggplot()` recebe um conjunto de dados e cria um objeto `ggplot` a partir dele. Camadas são somadas a esse objeto para criar o gráfico resultante.



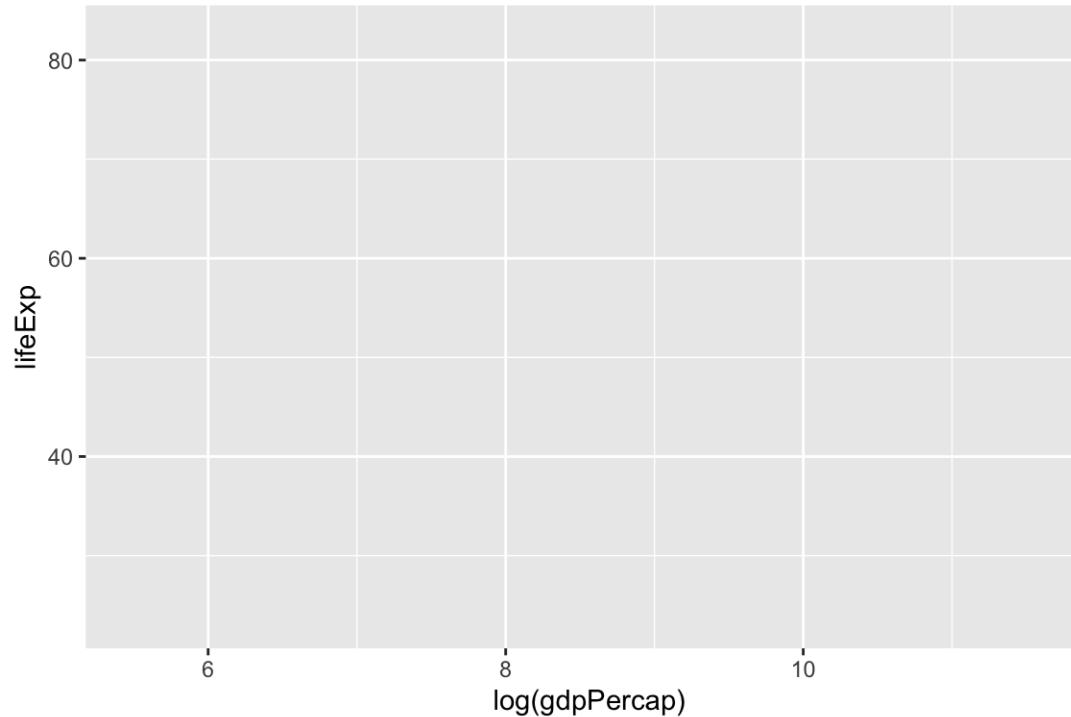
Criando um gráfico com `ggplot()`

```
library(gapminder);library( dplyr)  
glimpse(gapminder)
```

```
## Rows: 1,704  
## Columns: 6  
## $ country    <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ...  
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ...  
## $ year       <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ...  
## $ lifeExp    <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8...  
## $ pop        <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12...  
## $ gdpPercap   <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ...
```

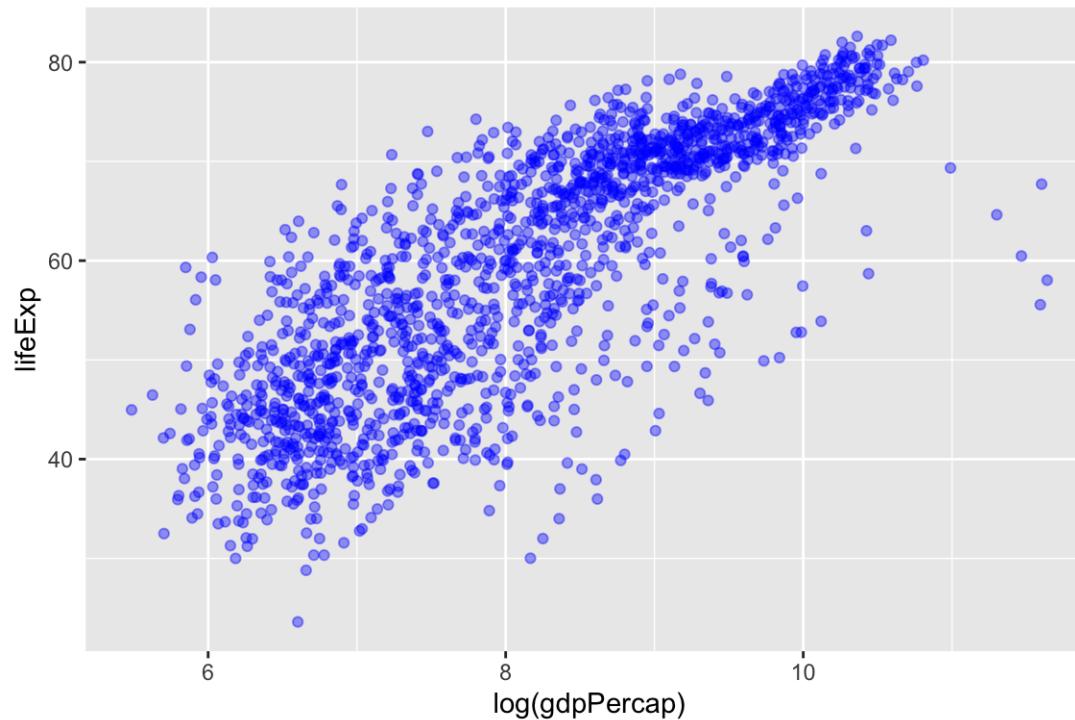
Criando um gráfico com `ggplot()`

```
ggplot(data = gapminder, aes(x = log(gdpPerCap), y = lifeExp))
```



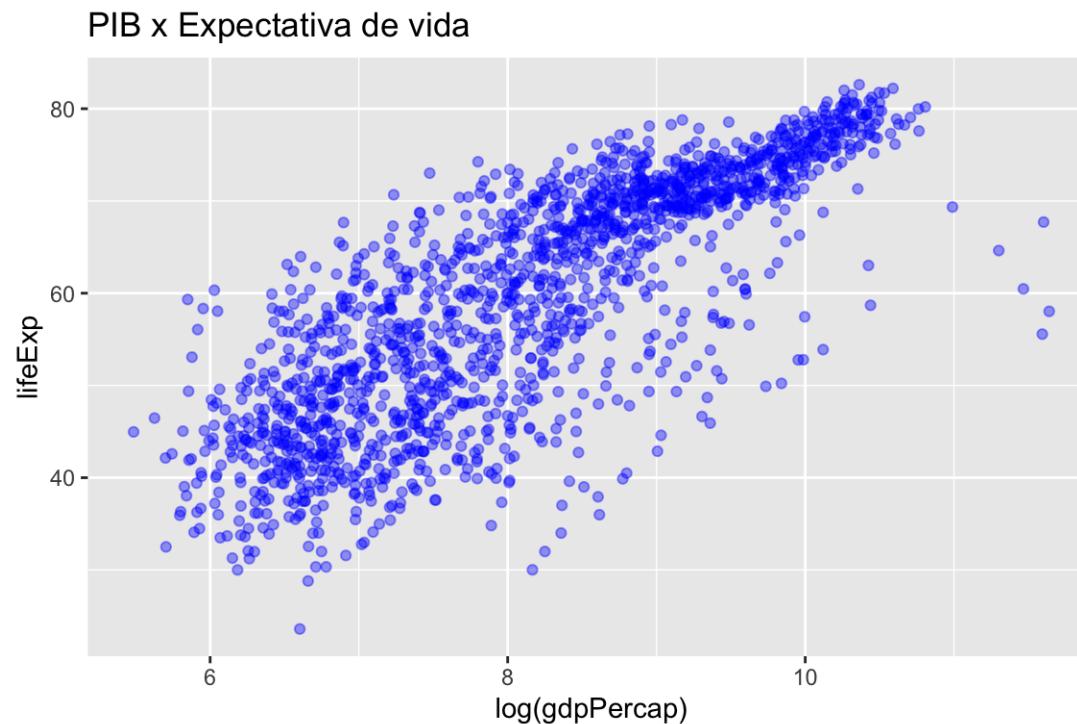
Criando um gráfico com `ggplot()`

```
ggplot(data = gapminder, aes(x = log(gdpPercap), y = lifeExp)) +  
  geom_point(color = "blue", alpha = 0.4)
```



Criando um gráfico com `ggplot()`

```
ggplot(data = gapminder, aes(x = log(gdpPercap), y = lifeExp)) +  
  geom_point(color = "blue", alpha = 0.4) +  
  ggtitle("PIB x Expectativa de vida")
```



Geometrias

É a primeira camada que será somada ao objeto ggplot e é o que define o tipo de gráfico. Os principais são:

- `geom_point()`: gráfico de pontos
- `geom_lines()`: gráfico de linhas
- `geom_boxplot()`: boxplot
- `geom_histogram()`: histograma
- `geom_bar()`: gráfico de barras
- `geom_density()`: gráfico de densidade

E muitos outros com aplicações mais específicas.

Mapeando Variáveis: aes()

A função `aes()` mapeia variáveis em atributos estéticos do gráfico como:

- `x` e `y`
- `color`, `fill` e `alpha`: cor, cor do preenchimento e transparência.
- `size`: para gráfico de pontos
- `linetype`: para gráficos com linhas
- etc

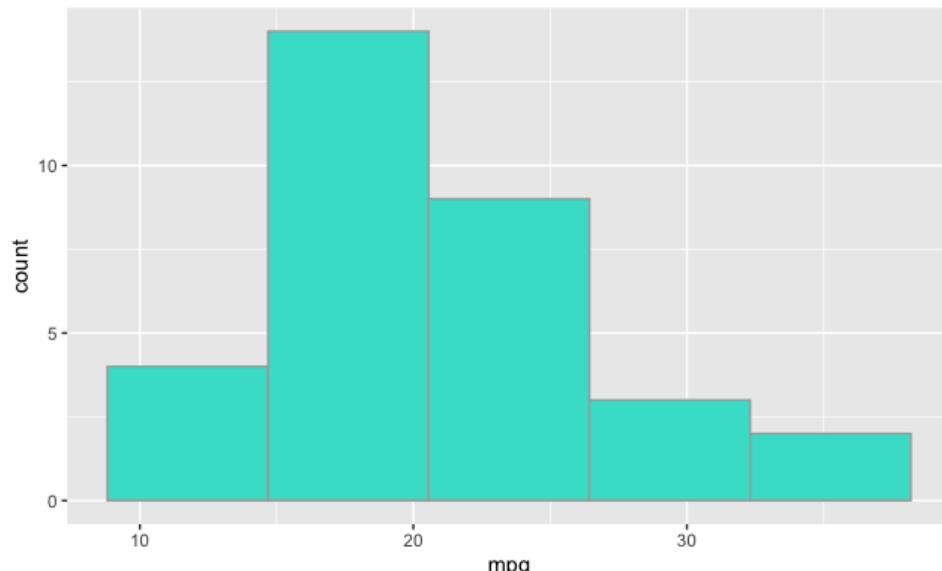
Pode ser definida de duas formas, dentro da função:

- `ggplot()`: para especificar globalmente quais a variáveis mapeadas; ou
- `geom_*`(): para usar apenas naquela “camada”.

Histograma no ggplot2

Vamos construir um histograma, agora usando o `ggplot2`.

```
ggplot(data = mtcars) +  
  geom_histogram(mapping = aes(x = mpg), bins = 5, fill = "turquoise", color = "darkgrey")
```



na forma geométrica escolhida.

O histograma ao lado foi construído em camadas, unidas pelo sinal de `+`:

- **1^a camada:** `ggplot` cria a tela inicial;
- **2^a camada:** `geom_histogram()` especifica a forma geométrica do gráfico;
- **3^a camada:** `aes()` especifica quais e como as variáveis serão mapeadas

Histogram no ggplot2

Se quisermos representar a densidade e não a contagem de pontos em cada intervalo, usamos `y = ..density..` dentro de `aes()`:

```
mtcars %>% ggplot() +  
  geom_histogram(aes(x = mpg, y = ..density..), bins = 5,  
                 fill = "turquoise", color = "darkgrey")
```

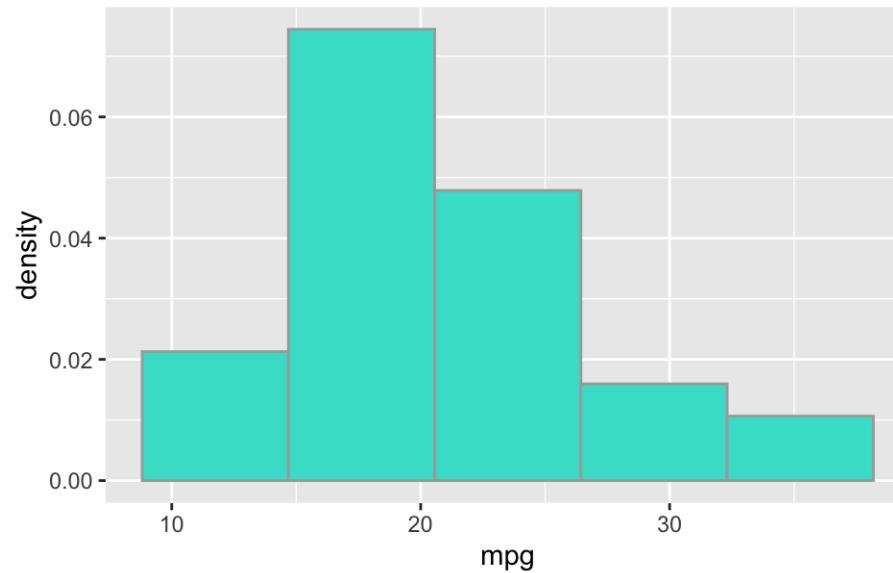
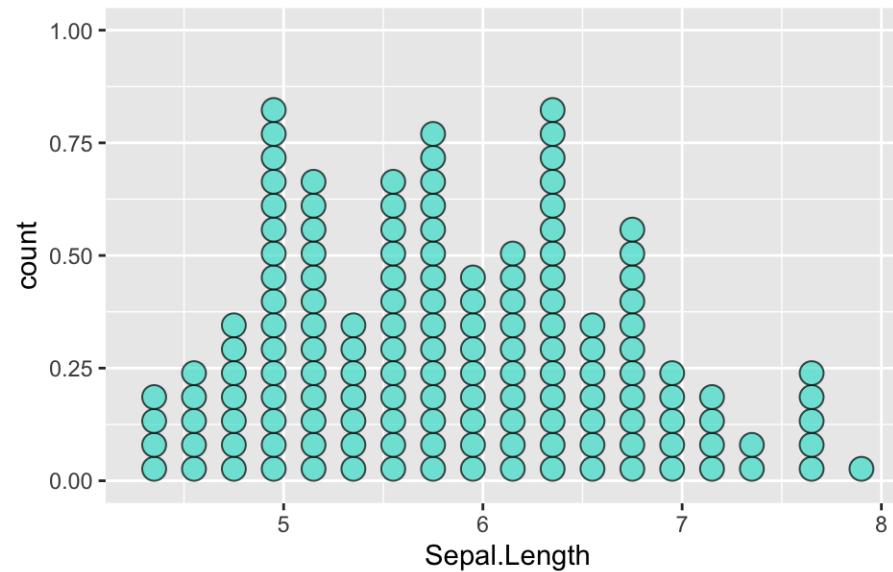


Gráfico de Pontos (*dotplot*)

Se o conjunto de dados for pequeno, podemos usar o *dotplot*. Semelhante ao gráfico “ramo-e-folhas”:

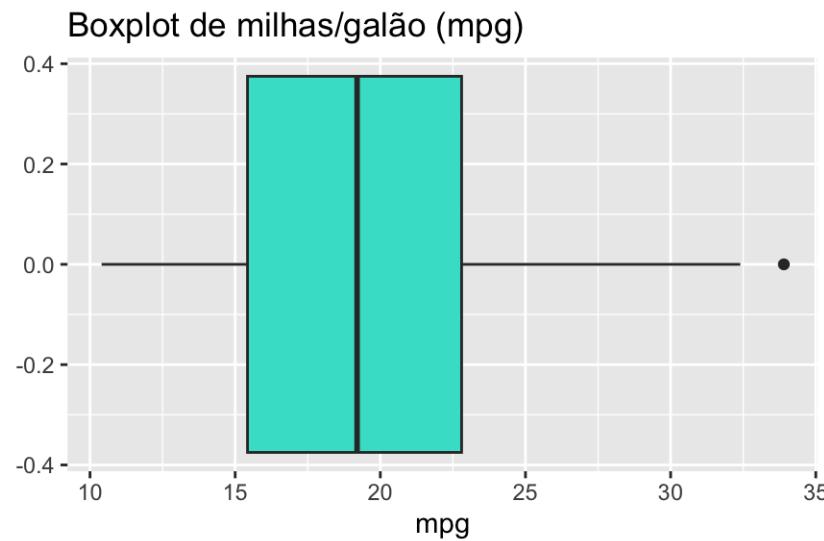
```
iris %>% ggplot() +  
  geom_dotplot(aes(x = Sepal.Length, y = ..count..), fill = "turquoise", alpha = 0.7)
```



Boxplot no `ggplot2`

A forma geométrica para criar um boxplot é o `geom_boxplot()`:

```
mtcars %>% ggplot() +  
  geom_boxplot(aes(x = mpg), fill = "turquoise") +  
  ggtitle("Boxplot de milhas/galão (mpg)")
```



Boxplot por Grupos no `ggplot2`

Para fazer o boxplot de uma variável contínua pelos níveis de uma variável categórica (fator no R), basta especificar o `x` em `aes()`:

```
mtcars %>% mutate(cyl = as.factor(cyl)) %>%
  ggplot() +
  geom_boxplot(aes(y = mpg, x = cyl, fill = cyl), show.legend = FALSE) +
  ggtitle("Boxplot de milhas/galão por cilindros")
```

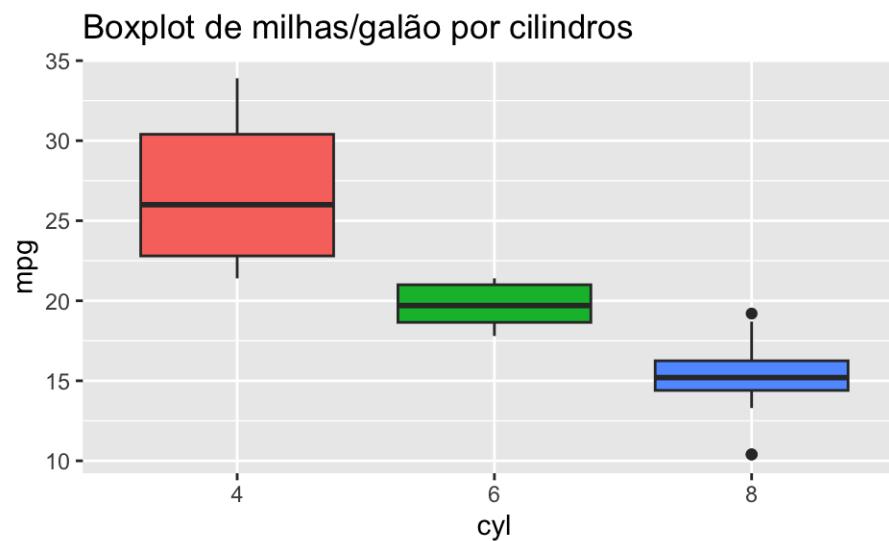
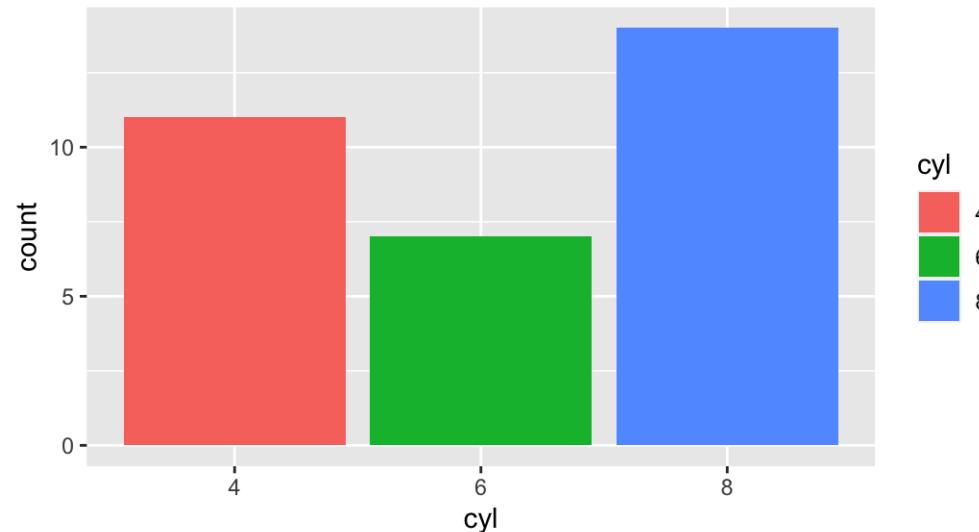


Gráfico de Barras no `ggplot2`

A forma geométrica para criar um gráfico de barras é o `geom_bar()`:

```
mtcars %>% mutate(cyl = as.factor(cyl)) %>%
  ggplot() +
  geom_bar(aes(x = cyl, fill = cyl))
```



Mais sobre `ggplot2`

Ainda há muito o que ser explorado no pacote `ggplot2`.

Na próxima aula, além de retomar algumas das formas já vistas, falaremos de outras formas geométricas:

`geom_point()`, `geom_line()`, `geom_smooth()`,
`geom_density()` e `geom_jitter()`.

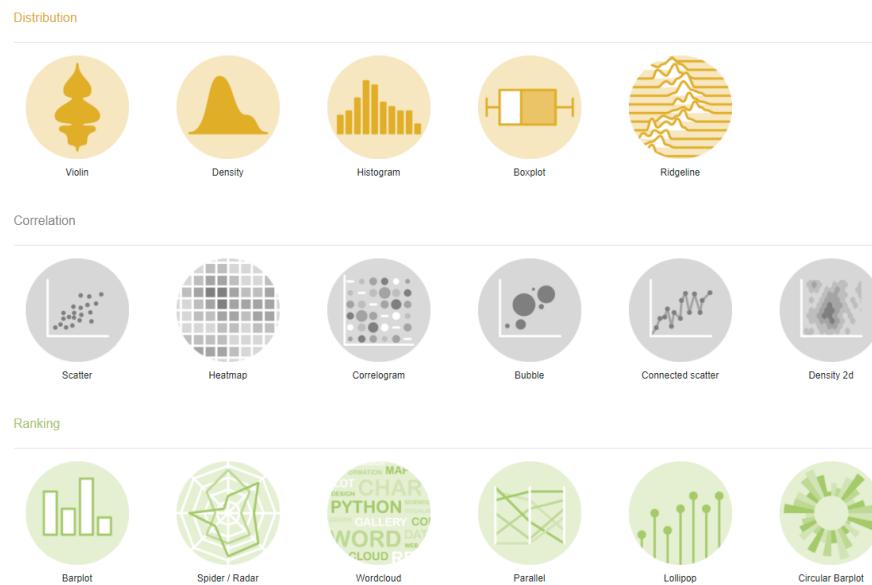
E também de outros aspectos dos gráficos:

- Temas
- Títulos e eixos
- Escalas e cores
- Múltiplos Gráficos
- Coordenadas



R Graph Gallery

Uma excelente fonte para exemplos de gráficos, organizados por tópicos, incluindo os códigos:



Fonte: [The R Graphs Gallery Blog](#)

From Data to Viz

Nesse site, você encontra indicações do gráfico mais apropriado para os seus dados!

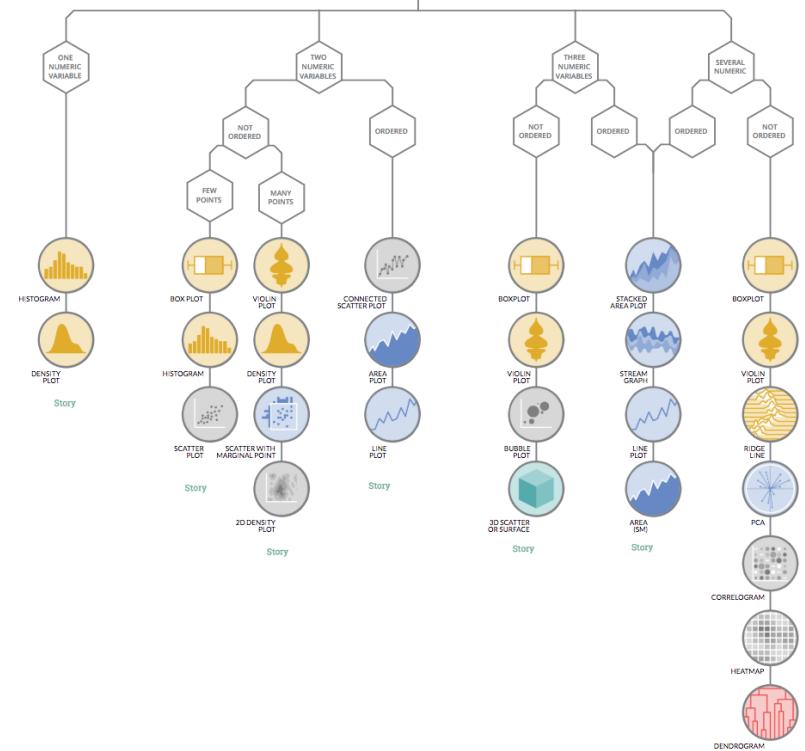


from Data to Viz

Fonte: [From Data to Viz](#)

What kind of data do you have? Pick the main type using the buttons below. Then let the decision tree guide you toward your graphic possibilities.

[Numeric](#) [Categoric](#) [Num & Cat](#) [Maps](#) [Network](#) [Time series](#)



Graphics Principles - Cheat Sheet

Graphics Principles Cheat Sheet v1.1

Communication

Effective visualizations communicate complex statistical and quantitative information facilitating insight, understanding, and decision making.

But what is an effective graph?

This cheat sheet provides general guidance and points to consider.

Planning

- Why**: Clearly identify the purpose of the graph, e.g. to deliver a message or for exploration?
- What**: Identify the quantitative evidence to support the purpose
- Who**: Identify the intended audience (specialists, non-specialists, both) and focus the design to support their needs
- Where**: Adapt the design to space or formatting constraints (e.g. clinical report, slide deck or publication)

Effectiveness Ranking

A graph is a representation of data that visually encodes numerical values into attributes such as lines, symbols and colors. The Cleveland-McGill scale can be used to select the most effective attribute(s) for your purpose.

Volume	Color hue	Depth: 3d position	Color intensity	Area	Slope or Angle	Length	Position on unaligned scale	Position on common scale
volume charts	poorly designed heat maps	multivariate density plots	heat maps	bubble charts, mosaic charts	line graphs, pie charts	stacked bar charts, waterfall chart	small multiple plots	dot plots, bar charts, parallel coordinate plots

Least accurate → Most accurate

Principles of Effective Graphic Design

Proximity – group related elements together

Alignment – elements on the same vertical or horizontal plane are perceived as having similar properties

Simplicity – cut anything superfluous, only include elements that add value, limit to 2-3 colors or fonts

White space (empty space) – use white space to minimize distraction & provide clarity

Legibility – sans serif fonts are easier to read, use color for emphasis instead of a new typeface

Color – select colors that present enough contrast to make the graph legible. Choose monochromatic color schemes to prevent clashing. Use dark colors and accent colors to emphasize important information

Visual Hierarchy – use color, font, image size, typeface, alignment & placement to create a viewing order

Focal Points – primary area of interest that immediately attracts the eye, emphasize the most important concept and make it your focal point. Use contrasting colors to draw attention

Repetition – repeating elements can be visually appealing, repeated shapes, labels, colors

Familiarity – using familiar styles, icons, navigation structure makes viewers feel confident

Consistency – be consistent with heading sizes, font choices, color scheme, and spacing. Use images with similar styles

Selecting the right base graph

Consider if a standard graph can be used by identifying suitable designs based on the:
(i) purpose (i.e. message to be conveyed or question to answer) and (ii) data (i.e. variables to display).

Example plots categorized by purpose

Deviation	Correlation	Ranking	Distribution	Evolution	Part-to-whole	Magnitude

Facilitating Comparisons

Proximity improves association

Place labels next to data instead of using legends

Group together elements to be compared directly

Ease visual inspection

Order values to help compare across many categories

Judgments are easier to make on a common vertical scale

Reduce mental arithmetic

Plot the final comparison e.g. mean difference not two means
Exception: If comparator is of interest in itself

Use reference lines and other visual anchors.

Color for emphasis or distinction

Restrained use of color is highly effective in organizing a narrative and calling attention to certain elements.

Think carefully before introducing additional color. Do you really need it?

Do not use color to differentiate between categories of the same variable

Use colors or shades to represent meaningful differences such as positive/negative values, treatments or doses

Be consistent, use the same color to mean the same thing in a series of graphs (e.g. treatment, dose)

Use a bold, saturated or contrasting color to emphasize important details

Emphasize the data by minimizing unnecessary ink, e.g. soften gridlines with a light color

Utilize existing resources for selection of appropriate palettes such as Color brewer or Munsell

Fonte: [Graphics Principles - Cheat Sheet](#)

Referências

Algumas referências utilizadas para a construção desse material:

- [R for Data Science - Chapter 3](#)
- [Curso R - Capítulo 8](#)
- [Hadley Wickham. A Layered Grammar of Graphics](#)
- [ggplot2: elegant graphics for data analysis](#)
- [Graphics Principles - Cheat Sheet](#)



Slides produzidos pela profa. Tatiana Benaglia.