Exploração e visualização de dados

Gilberto Pereira Sassi

Departamento de Estatística
Instituto de Matemática e Estatística

Departamento de Estatística
IME - UFBA

Sobre o curso

Departamento de Estatística
IME - UFBA

Preparando o ambiente

- Em casa, você pode usar:
  - colab.research.google.com/#create=true&language=r;
  - posit.cloud.
- No seu dia-a-dia, recomenda-se instalar o `R` com versão pelo menos `4.1`: cran.r-project.org.
- **IDE** recomendadas: *RStudio* e *VSCode*.
  - Caso você queira usar o *VSCode*, instale a extensão da linguagem `R`.
- Neste curso, usaremos o *framework* **tidyverse**:
  - Instale o framework a partir do repositório CRAN: `install.packages("tidyverse")`
- Outras linguagens interessantes: `python` e `julia`.
  - `python`: linguagem interpretada de próposito geral, contemporânea do `R`, simples e fácil de aprender.
  - `julia`: linguagem interpretada para análise de dados, lançada em 2012, promete simplicidade e velocidade.

A linguagem R:

uma introdução

Departamento de Estatística
IME - UFBA

O precursor do R: S.

- R é uma linguagem derivada do S.
- S foi desenvolvido em `fortran` por **John Chambers** em *1976* no **Bell Labs**.
- S foi desenvolvido para ser um ambiente de análise estatística.
- Filosofia do S: permitir que usuários possam analisar dados usando estatística com pouco conhecimento de programação.

História do R

- Em *1991*, **Ross Ihaka** e **Robert Gentleman** criaram o R na **Nova Zelândia**.
- Em *1996*, **Ross** e **Robert** liberam o R sob a licença "GNU General License", o que tornou o R um software livre.
- Em *1997*, **The Core Group** é criado para melhorar e controlar o código fonte do R.

Porque usar R

- Constante melhoramento e atualização.
- Portabilidade (roda em praticamente todos os sistemas operacionais).
- Grande comunidade de desenvolvedores que adicionam novas capacidades ao R através de pacotes.
- Gráficos de maneira relativamente simples.
- Interatividade.
- Um grande comunidade de usuários (especialmente útil para resolução de problemas).

Onde estudar fora de aula?

Livros

Recomendo principalmente o livro R for Data Science.

- **Nível Iniciante:** R Tutorial na W3Schools.
- **Nível Iniciante:** Hands-On Programming with R.
- **Nível Iniciante:** R for Data Science.
- **Nível Intermediário:** Advanced R.

Livros em português

- **Nível *cheguei agora aqui*:** zen do R.
- **Nível Avançado:** Advanced R.
- **Nível Iniciante:** material.curso-r.com.
- **Nível Iniciante:** ecoR.
- **Nível Iniciante:** analises-ecologicas.com.

Plataformas de ensino on-line

- **Datacamp:** datacamp.com
- **Dataquest:** dataquest.io

O que você pode fazer quando estiver em apuros?

- consultar a documentação do R:

```
help(mean)
?mean
```

- Peça ajuda a um programador mais experiente.
- Conmsulte Rstudio community.
- Consulte pt.stackoverflow.com.
- Use ferramentas de busca como o google e duckduckgo.com.

```
sqrt("Gilberto")
```

- Na ferramenta de busca, pesquise por `Error in sqrt("Gilberto"): non-numeric argument to mathematical function`

Soma

```
1 + 1
```

```
[1] 2
```

Substração

```
2 - 1
```

```
[1] 1
```

Divisão

```
3 / 2
```

```
[1] 1.5
```

Potenciação

```
2^3
```

```
[1] 8
```

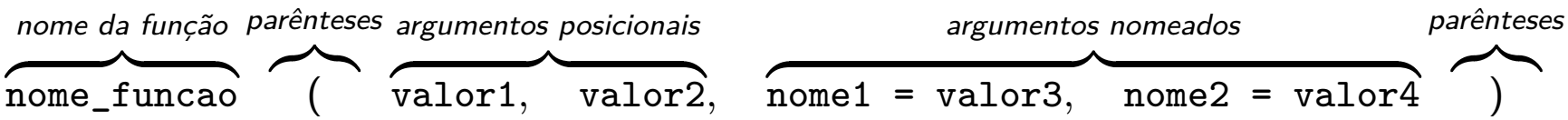
Exercício

Qual o resultado das seguintes operações?

1. `5.32 + 7.99`
2. `5.55 - 10`
3. `3.33 * 5.12`
4. `1 / 4.55`
5. `5^1.23`

Função: é uma ação e tem os seguinte componentes na ordem:

- *nome da função*
- *parênteses*
- *argumentos posicionais*
- *argumentos nomeados*



example:

```
read_xlsx('data/raw/casas.xlsx', sheet=1)
```

Funções na linguagem R
Exercício

- Obtenha ajuda para `mean` usando a função `help`.
- Calcule o logaritmo de 10 na base 3 usando a função `log`.
- Leia o conjunto de dados `amostra_enem_salvador.xlsx` usando a função `read_xlsx` do pacote `readxl`.

Estrutura de dados no R

- **Tipo de dados:** caracter (`character`), número real (`double`), número inteiro (`integer`), número complexo (`complex`) e lógico (`logical`).
- **Estrutura de dados:** atomic `vector` (a estrutura de dados mais básicA no R), `matrix`, `array`, `list` e `data.frame` (`tibble` no `tidyverse`).
- **Estrutura de dados Homogênea:** `vector`, `matrix` e `array`.
- **Estrutura de dados Heterôgenea:** `list` e `data.frame` (`tibble` no `tidyverse`).

Número inteiro

```
class(1L)
```

```
[1] "integer"
```

Número real

```
class(1.2)
```

```
[1] "numeric"
```

Número complexo

```
class(1 + 1i)
```

```
[1] "complex"
```

Número lógico ou valor booleano

```
class(TRUE)
```

```
[1] "logical"
```

Caracter ou *string*

```
class("Gilberto")
```

```
[1] "character"
```

Vetor

- Agrupamento de valores de mesmo tipo em um único objeto.
- Criação de vetor:
  - `c(...)`;
  - `vector('<tipo de dados>', <comprimento do vetor>)`;
  - `seq(from = a, to = b, by = c)`;
  - `seq_along(<vetor>)` - vetor de números inteiros com o mesmo trabalho de `<vetor>`;
  - `seq_len(<número inteiro>)` - vetor de números inteiros com o tamanho `<número inteiro>`;
  - `<número inicial>:<número final>` - sequência de números inteiros entre `<número inicial>` e `<número final>`
- Podemos checar o `tipo de dados` de um vetor com a função `class`.

Vetor de caracteres

```
nomes  <- c("Gilberto", "Sassi")
class(nomes)
```

```
[1] "character"
```

```
nomes
```

```
[1] "Gilberto" "Sassi"
```

```
texto_vazio <- vector("character", 3)
class(texto_vazio)
```

```
[1] "character"
```

```
texto_vazio
```

```
[1] "" "" ""
```

Vetor de números reais

```
vetor_real  <- c(0.2, 1.35)
class(vetor_real)
```

```
[1] "numeric"
```

```
vetor_real
```

```
[1] 0.20 1.35
```

```
vetor_real <- vector("double", 3)
vetor_real
```

```
[1] 0 0 0
```

```
vetor_real <- seq(from = 1, to = 3.5, by = 0.5)
vetor_real
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5
```

Vetor de números inteiros

```
vetor_inteiro  <- c(1L, 2L)
class(vetor_inteiro)
```

```
[1] "integer"
```

```
vetor_inteiro
```

```
[1] 1 2
```

```
vetor_inteiro <- vector("integer", 3)
vetor_inteiro
```

```
[1] 0 0 0
```

```
vetor_inteiro <- 1:4
vetor_inteiro
```

```
[1] 1 2 3 4
```

```r
vetor_real  <- seq_along(nomes)
class(vetor_real)
```

```
[1] "integer"
```

```r
vetor_real
```

```
[1] 1 2
```

```r
vetor_real <- seq_len(5)
class(vetor_real)
```

```
[1] "integer"
```

```r
vetor_real
```

```
[1] 1 2 3 4 5
```

Vetor lógico

```
vetor_logico  <- c(TRUE, FALSE)
class(vetor_logico)
```

```
[1] "logical"
```

```
vetor_logico
```

```
[1]  TRUE FALSE
```

```
vetor_logico <- vector("logical", 3)
vetor_logico
```

```
[1] FALSE FALSE FALSE
```

Estrutura de dados homogênea
Exercício

Crie os seguintes vetores:

1. $(0,1 \quad 0,2 \quad 0,3 \quad 0,4 \quad 0,5)$
2. $(TRUE \quad TRUE \quad FALSE)$
3. $(\text{"}Marx\text{"} \quad \text{"}Engels\text{"} \quad \text{"}Lênin\text{"})$
4. $(1 \quad 2 \quad 3)$

Departamento de Estatística
IME - UFBA

Operações com vetores números (`double`, `integer` e `complex`).

- Operações básicas (operação, substração, multiplicação e divisão) realizada em cada elemento do vetor.
- *Slicing*: extrair parte de um vetor (não precisa ser vetor numérico).

Slicing

```
vetor <- c("a", "b", "c", "d", "e", "f", "g", "h", "i")
# selecionado todos os elementos entre o primeiro e o quinta
vetor[1:5]
```

```
[1] "a" "b" "c" "d" "e"
```

Adição (vetores números)

```
vetor_1 <- 1:5
vetor_2 <- 6:10
vetor_1 + vetor_2
```

```
[1]  7  9 11 13 15
```

Substração (vetores numéricos)

```
vetor_1 <- 1:5
vetor_2 <- 6:10
vetor_2 - vetor_1
```

```
[1] 5 5 5 5 5
```

Multiplicação (vetores numéricos)

```
vetor_1 <- 1:5
vetor_2 <- 6:10
vetor_2 * vetor_1
```

```
[1]  6 14 24 36 50
```

Divisão (vetores numéricos)

```
vetor_1 <- 1:5
vetor_2 <- 6:10
vetor_2 / vetor_1
```

```
[1] 6.000000 3.500000 2.666667 2.250000 2.000000
```

Estrutura de dados homogênea
Exercício

Realize as seguintes operações envolvendo vetores:

1. $\begin{pmatrix}1 & 2 & 3\end{pmatrix} + \begin{pmatrix}0,1 & 0,05 & 0,33\end{pmatrix}$
2. $\begin{pmatrix}1 & 2 & 3\end{pmatrix} - \begin{pmatrix}0,1 & 0,05 & 0,33\end{pmatrix}$
3. $\begin{pmatrix}1 & 2 & 3\end{pmatrix} * \begin{pmatrix}0,1 & 0,05 & 0,33\end{pmatrix}$
4. $\begin{pmatrix}1 & 2 & 3\end{pmatrix} / \begin{pmatrix}0,1 & 0,05 & 0,33\end{pmatrix}$

Matriz

- Agrupamento de valores de mesmo tipo em um único objeto de dimensão 2.
- Criação de matriz:
  - `matrix(..., nrow = <integer>, ncol = <integer>, byrow = TRUE)` - preenche a matriz a partir das linhas se `byrow = TRUE`;
  - `diag(<vector>)` - diagonal principal igual a `<vetor>` e outros elementos zero;
  - `rbind()` - especificação das linhas da matriz;
  - `cbind()` - especificação das colunas da matriz.

Matriz de caracteres

```
matriz_texto <- rbind(c("a", "b"), c("c", "d"))
matriz_texto
```

```
     [,1] [,2]
[1,] "a"  "b"
[2,] "c"  "d"
```

Matriz de números reais

```
matriz_real <- matrix(seq(from = 0, to = 1.5, by = 0.5),
                      nrow = 2, byrow = TRUE)
matriz_real
```

```
     [,1] [,2]
[1,]    0  0.5
[2,]    1  1.5
```

Matriz de inteiros

```
matriz_inteiro <- cbind(c(1L, 2L), c(3L, 4L))
matriz_inteiro
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

Matriz de valores lógicos

```
matriz_logico <- matrix(c(TRUE, F, F, T), nrow = 2)
matriz_logico
```

```
      [,1]  [,2]
[1,]  TRUE FALSE
[2,] FALSE  TRUE
```

Array

- Agrupamento de valores de mesmo tipo em um único objeto em duas ou mais dimensões.
- Criação de array: `array(..., dim = <vector of integers>)`.

```
dados_matriz_1 <- 10:13
dados_matriz_2  <- 14:17
resultado <- array(c(dados_matriz_1, dados_matriz_2),
                   dim = c(2, 2, 2))
resultado
```

```
, , 1

     [,1] [,2]
[1,]   10   12
[2,]   11   13

, , 2

     [,1] [,2]
[1,]   14   16
[2,]   15   17
```

Operações com matrizes números (`double`, `integer` e `complex`).

- Operações básicas (operação, substração, multiplicação e divisão) realizada em cada elemento das matrizes.
- Outras operações:
  - Multiplicação de matrizes;
  - Inversão de matrizes;
  - Matriz transposta;
  - Determinante;
  - Solução de sistema de equações lineares.

Matrizes

```
matriz_a <- rbind(c(1, 2), c(0, 3))
matriz_b <- matrix(runif(4), ncol = 2)
```

Soma

```
matriz_soma <- matriz_a + matriz_b
matriz_soma
```

```
          [,1]     [,2]
[1,] 1.0205102 2.305041
[2,] 0.8506394 3.928573
```

Subtração

```
matriz_menos <- matriz_a - matriz_b
matriz_menos
```

```
           [,1]     [,2]
[1,]  0.9794898 1.694959
[2,] -0.8506394 2.071427
```

Produto de Hadamard

- Multiplicação de matrizes, elemento por elemento.
- Para detalhes consulte produto de Hadamard.

```
matriz_hadamard <- matriz_a * matriz_b
matriz_hadamard
```

```
           [,1]      [,2]
[1,] 0.0205102 0.6100826
[2,] 0.0000000 2.7857189
```

Multiplicação de matrizes

```
matriz_multiplicacao <- matriz_a %*% matriz_b
matriz_multiplicacao
```

```
         [,1]     [,2]
[1,] 1.721789 2.162187
[2,] 2.551918 2.785719
```

Matriz inversa

```
matriz_inversa <- solve(matriz_a)
matriz_inversa
```

```
     [,1]       [,2]
[1,]    1 -0.6666667
[2,]    0  0.3333333
```

```
matriz_a %*% matriz_inversa
```

```
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

Matriz transposta

```
matriz_transposta <- t(matriz_a)
matriz_transposta
```

```
     [,1] [,2]
[1,]    1    0
[2,]    2    3
```

Determinante

```
det(matriz_a)
```

```
[1] 3
```

Solução de sistema de equações lineares

```
b <- c(1, 2)
solve(matriz_a, b)
```

```
[1] -0.3333333  0.6666667
```

Matriz inversa generalizada

G é a matriz inversa generalizada de A se $A \cdot G \cdot A = A$. Para detalhes vide matriz inversa generalizada.

```
library(MASS) # ginv é uma função do pacote MASS
ginv(matriz_a)
```

```
              [,1]       [,2]
[1,]  1.000000e+00 -0.6666667
[2,] -2.775558e-17  0.3333333
```

Outras operações com matrizes.

Operador ou função	Descrição
`A %o% B`	produto diádico $A \cdot B^T$
`crossprod(A, B)`	$A \cdot B^T$
`crossprod(A)`	$A \cdot A^T$
`diag(x)`	retorna uma matrix diagonal com diagonal igual a `x`
`diag(A)`	retorna um vetor com a diagona de A
`diag(k)`	retorna uma matriz diagona de ordem k

Exercício

Realize as seguinte operações envolvendo as matrizes:

1. $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} + \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$
2. $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} - \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$
3. Multiplicação de matriz: $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$
4. Divisão elemento a elemento: $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} / \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$
5. Resolva o seguinte sistema de equações: $\begin{cases} x + 2y = 21 \\ x - 2y = 1 \end{cases}$.
6. Encontre a matriz inversa de $\begin{pmatrix} 1 & 2 \\ 1 & -2 \end{pmatrix}$.

Lista

- Agrupamento de valores de tipos diversos e estrutura de dados
- Criação de listas: `list(...)` e `vector("list", <comprimento da lista>)`

```
lista_info <- list(pedido_id = 8001406,
          nome = "Fulano",
          sobrenome = "de Tal",
          cpf = "12345678900",
          itens = list(list(descricao = "Ferrari",
                            frete = 0,
                            valor = 500000),
                       list(descricao = "Dolly", frete = 1.5,
                            valor = 3.90)))
lista_info
```

```
$pedido_id
[1] 8001406

$nome
[1] "Fulano"

$sobrenome
[1] "de Tal"

$cpf
[1] "12345678900"

$itens
$itens[[1]]
$itens[[1]]$descricao
[1] "Ferrari"

$itens[[1]]$frete
[1] 0

$itens[[1]]$valor
[1] 5e+05


$itens[[2]]
$itens[[2]]$descricao
[1] "Dolly"

$itens[[2]]$frete
[1] 1.5

$itens[[2]]$valor
[1] 3.9
```

Estrutura de dados heterogênea
Exercício

Crie uma lista, chamada `informacoes_pessoais` com os seguintes campos:

- `nome`: seu nome
- `idade`: sua idade
- `informacao_profissional`: uma lista com os seguintes campos:
  - `matricula`: escolaridade
  - `origem`: variável qualitativa com a sua cidade de origem.
- `matriz`: inclua uma matriz de números reais de dimensão 2×2

- *slicing* - `[]` - extrai parte da lista (valor retornado é uma lista).
- Acessando *k*-ésimo valor da lista: `lista[[k]]`.
- Acessando um valor da lista pela chave (nome do campo): `lista$cpf`.
- Concatenação de listas: `c()`.

Slicing

```
lista_info[c(2, 4)]
```

```
$nome
[1] "Fulano"

$cpf
[1] "12345678900"
```

Acessando elemento pela posição

```
lista_info[[2]]
```

```
[1] "Fulano"
```

Acessando elemento pela chave

```
lista_info$nome
```

```
[1] "Fulano"
```

Concatenação de listas

```
lista_1 <- list(1, 2)
lista_2 <- list("Gilberto", "Sassi")
lista_concatenada <- c(lista_1, lista_2)
lista_concatenada
```

```
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] "Gilberto"

[[4]]
[1] "Sassi"
```

Estrutura de dados heterogênea
Exercício

Recupe e imprima as seguintes informações da lista `informacoes_pessoais`:

- os três primeiros campos de `informacoes_pessoais`
- os nomes dos campos de `informacoes_pessoais`
- campo `nome` de `informacoes_pessoais`
- o terceiro campo de `informacoes_pessoais`

Tidy data

- Dados em formato de tabela.
- Cada coluna é uma variável e cada linha é uma observação.

`tibble` (data frame)

- Estrutura de dados tabular.
- Assumimos que os dados estão **tidy**.
- Criação de `tibble`: `tibble(...)` e `tribble(....)`.
- `glimpse` mostra as informações do `tibble`.

```r
library(tidyverse) # carregando o framework tidyverse
data_frame <- tibble(
  nome = c("Marx", "Engels", "Rosa", "Lênin", "Olga Benário"),
  idade = c(22, 23, 21, 24, 30)
)
glimpse(data_frame)
```

```
Rows: 5
Columns: 2
$ nome  <chr> "Marx", "Engels", "Rosa", "Lênin", "Olga Benário"
$ idade <dbl> 22, 23, 21, 24, 30
```

Valores especiais em R

Valores especiais	Descrição	Função para identificar
`NA`	Valor faltante.	`is.na()`
`NaN`	Resultado do cálculo indefinido.	`is.nan()`
`Inf`	Valor que excede o valor máximo que sua máquina aguenta.	`is.inf()`
`NULL`	Valor indefinido de expressões e funções (diferente de `NaN` e `NA`)	`is.null()`

Operações básicas em um `tibble`

Função	Descrição
`head()`	Mostra as primeiras linhas de um `tibble`
`tail()`	Mostra as últimas linhas de um `tibble`
`glimpse()`	Impressão de informações básicas dos dados
`add_case()`	Adiciona uma nova observação
`add_row()`	Adiciona uma nova observação

```r
head(data_frame, n=2)
```

```
# A tibble: 2 x 2
  nome   idade
  <chr>  <dbl>
1 Marx      22
2 Engels    23
```

```r
tail(data_frame, n=2)
```

```
# A tibble: 2 x 2
  nome          idade
  <chr>         <dbl>
1 Lênin            24
2 Olga Benário     30
```

Estrutura de dados heterogênea
Exercício

Realize as seguintes operações no *dataset* `iris` (disponível no `R`):

- imprima um resumo sobre o *dataset* `iris`.
- pegue as 5 primeiras linhas de `iris`.
- pegue as 5 últimas linhas de `iris`.
- crie *na mão* o seguinte conjunto de dados:

nomes	origem
Fidel Castro	Cuba
Ernesto 'Che' Guevara	Cuba
Célia Sánchez	Cuba

Organização é fundamental

Departamento de Estatística
IME - UFBA

O nome de um objeto precisa ter um *significado*.

O nome deve indicar e deixar claro o que este objeto é ou faz.

- Use a convenção do R:
  - Use apenas letras minúsculas, números e *underscore* (comece sempre com letras minúsculas).
  - Nomes de objetos precisam ser substantivos e precisam descrever o que este objeto é ou faz (seja conciso, direto e significativo).
  - Evite ao máximo os nomes que já são usados (*buit-in*) do R.Por exemplo: `c`.
  - Coloque espaço depois da vírgula.
  - Não coloque espaço antes nem depois de parênteses. Exceção: Coloque um espaço `()` antes e depois de `if`, `for` ou `while`, e coloque um espaço depois de `()`.
  - Coloque espaço entre operadores básicos: `+`, `-`, `*`, `==` e outros. Exceção: `^`.

Estrutura de diretórios

Mantenha uma estrutura (organização) consistente de diretórios em seus projetos.

- Sugestão de estrutura:
  - `dados`: diretório para armazenar seus conjuntos de dados.
    - `brutos`: dados brutos.
    - `processados`: dados processados.
  - `scripts`: código fonte do seu projeto.
  - `figuras`: figuras criadas no seu projeto.
  - `output`: outros arquivos que não são figuras.
  - `legado`: arquivos da versão anterior do projeto.
  - `notas`: notas de reuniões e afins.
  - `relatorio` (ou `artigos`): documento final de seu projeto.
  - `documentos`: livros, artigos e qualquer coisa que são referências em seu projeto.

Para mais detalhes, consulte esse guia do curso-r: diretórios e `.Rproj`.

Importação e exportação de dados

Departamento de Estatística
IME - UFBA

Leitura de arquivos no formato `xlsx` ou `xls`

- **Pacote:** `readxl`
- Parêmetros das funções `read_xls` (arquivos `.xls`) e `read_xlsx` (arquivos `.xlsx`):
  - `path`: caminho até o arquivo.
  - `sheet`: especifica a planilha do arquivo que será lida.
  - `range`: especifica uma área de uma planilha para leitura. Por exemplo: `B3:E15`.
  - `col_names`: Argumento lógico com valor padrão igual a `TRUE`. Indica se a primeira linha tem o nome das variáveis.

Para mais detalhes, consulte a documentação: documentação de `read_xl`.

Leitura de arquivos no formato `xlsx` ou `xls`

```
library(tidyverse)
library(readxl)
dados_iris <- read_xlsx("dados/brutos/iris.xlsx")
dados_iris <- clean_names(dados_iris)

glimpse(dados_iris)
```

```
Rows: 150
Columns: 5
$ comprimento_sepala <dbl> 5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4,
$ largura_sepala     <dbl> 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9,
$ comprimento_petala <dbl> 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5, 1.4,
$ largura_petala     <dbl> 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2, 0.2,
$ especies           <chr> "setosa", "setosa", "setosa", "setosa", "set
```

Lendo dados no R
Exercício

Leia o *dataset* `dados_leitura.xlsx` usando o pacote `readxl`.

As formatações dos arquivos `csv`

- `csv`: *comma separated values* (valores separados por coluna). O *separador* varia em diferentes sistemas de medidas.

- No sistema métrico:
  - As casas decimais são separadas por ,
  - O agrupamento de milhar é marcada por .
  - As colunas dos arquivos de texto são separadas por ;

- No sistema imperial inglês (UK e USA):
  - As casas decimais são separadas por .
  - O agrupamento de milhar é marcada por ,
  - As colunas dos arquivos de texto são separadas por ,

Preste atenção em como o seus dados estão armazenados!

Leitura de arquivos no formato `csv`

- **Pacote:** `readr` do `tidyverse` (instale com o comando `install.packages('readr')`).
- Parêmetros das funções `read_csv` (sistema imperial inglês) e `read_csv2` (sistema métrico):
  - `path`: caminho até o arquivo.

Para mais detalhes, consulte a documentação oficial do *tidyverse*: documentação de `read_r`.

Leitura de arquivos no formato csv

```
dados_mtcarros <- read_csv2("dados/brutos/mtcarros.csv")
dados_mtcarros <- clean_names(dados_mtcarros)
glimpse(dados_mtcarros)
```

```
Rows: 32
Columns: 11
$ milhas_por_galao <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1, 14.3, 24.4, 22.8,~
$ cilindros        <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8, 8, 8, 8, 8, 8, 4,~
$ cilindrada       <dbl> 160.0, 160.0, 108.0, 258.0, 360.0, 225.0, 360.0, 146.~
$ cavalos_forca    <dbl> 110, 110, 93, 110, 175, 105, 245, 62, 95, 123, 123, 1~
$ eixo             <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76, 3.21, 3.69, 3.92,~
$ peso             <dbl> 2.620, 2.875, 2.320, 3.215, 3.440, 3.460, 3.570, 3.19~
$ velocidade       <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 20.22, 15.84, 20.0~
$ forma            <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1,~
$ transmissao      <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,~
$ marchas          <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 4,~
$ carburadores     <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3, 3, 3, 4, 4, 4, 1,~
```

Lendo dados no R
Exercício

Leia o *dataset* `dados_leitura.csv` usando o pacote `readr`.

Leitura de arquivos no formato `ods`

- **Pacote:** `readODS` (instale com o comando `install.packages('readODS')`).
- Parêmetros das funções `read_ods`:
- `path`: caminho até o arquivo.
  - `sheet`: especifica a planilha do arquivo que será lida.
  - `range`: especifica uma área de uma planilha para leitura. Por exemplo: `B3:E15`.
  - `col_names`: Argumento lógico com valor padrão igual a `TRUE`. Indica se a primeira linha tem o nome das variáveis.

Para mais detalhes, consulte a documentação do *readODS*: documentação de `readODS`.

Leitura de arquivos no formato ods

```
library(readODS)
dados_dentes <- read_ods("dados/brutos/crescimento_dentes.ods")
dados_dentes <- clean_names(dados_dentes)

glimpse(dados_dentes)
```

```
Rows: 60
Columns: 3
$ comprimento <dbl> 4.2, 11.5, 7.3, 5.8, 6.4, 10.0, 11.2, 11.2,
$ suplemento  <chr> "Vitamina C", "Vitamina C", "Vitamina C", "V
$ dose        <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
```

Lendo dados no R
Exercício

Leia o *dataset* `dados_leitura.ods` usando o pacote `readODS`.

Salvar no formato `.csv` (sistema métrico)

`write_csv2` é parte do pacote `readr`.

```
write_csv2(dados_dentes, file = "dados/processados/nome.csv")
```

Salvar no formato `.xlsx`

`write_xlsx` é parte do pacote `writexl`.

```
write_xlsx(dados_dentes, path = "dados/processados/nome.xlsx")
```

Salvar no formato `ods`

`write_ods` é parte do pacote `readODS`.

```
write_ods(dados_toothgrowth, path = "dados/processados/nome.ods")
```

Salvando dados no R
Exercício

1. Salve o objeto `milhas` do pacote `dados` como `milhas.ods` na pasta `output` do seu projeto.
2. Salve o objeto `diamante` do pacote `dados` como `diamante.csv` na pasta `output` do seu projeto.
3. Salve o objeto `velho_fiel` do pacote `dados` como `velho_fiel.xlsx` na pasta `output` do seu projeto.

O operador pipe
|>

Departamento de Estatística
IME - UFBA

O valor resultante da expressão do lado esquerdo vira primeiro argumento da função do lado direito.

Principal vantagem: simplifica a leitura e a documentação de funções compostas.

Executar

```
f(x, y)
```

é exatamente a mesma coisa que executar

```
x |> f(y)
```

```r
log(sqrt(sum(x^2)))
```

é exatamente a mesma coisa que executar

```r
x^2 |> sum() |> sqrt() |> log()
```

Fazendo um bolo

Exemplo adaptado de 6.1 O operador pipe.

Para cozinhar o bolo precisamos usar as seguintes funções:

- `acrescente(lugar, algo)`
- `misture(algo)`
- `asse(algo)`

Fazendo um bolo

- Passo 1:

```
acrescente(
  "tigela vazia",
  "farinha"
)
```

- Passo2:

```
acrescente(
  acrescente(
    "tigela vazia",
    "farinha"
  ),
  "ovos"
)
```

- Passo3:

```
acrescente(
  acrescente(
    acrescente(
      "tigela vazia",
      "farinha"
    ),
    "ovos"
  ),
  "leite"
)
```

- Passo4:

```
acrescente(
  acrescente(
    acrescente(
      acrescente(
        "tigela vazia",
        "farinha"
      ),
      "ovos"
    ),
    "leite"
  ),
  "fermento"
)
```

- Passo 5:

```
misture(
  acrescente(
    acrescente(
      acrescente(
        acrescente(
          acrescente(
            "tigela vazia",
            "farinha"
          ),
          "ovos"
        ),
        "leite"
      ),
      "fermento"
  )
)
```

- Passo 6:

```
asse(
  misture(
    acrescente(
      acrescente(
        acrescente(
          acrescente(
            "tigela vazia",
            "farinha"
          ),
          "ovos"
        ),
        "leite"
      ),
      "fermento"
    )
  )
)
```

Usando o operador |>.

```
acrescente("tigela vazia", "farinha") |>
  acrescente("ovos") |>
  acrescente("leite") |>
  acrescente("fermento") |>
  misture() |>
  asse()
```

Estatística descritiva

Departamento de Estatística
IME - UFBA

- **População**: todos os elementos ou indivíduos alvo do estudo.
- **Amostra**: parte da população.
- **Parâmetro:** característica numérica da população. Usamos letras gregas para denotar parâmetros populacionais.
- **Estatística:** função ou *cálculo* da amostra
- **Estimativa:** característica numérica da amostra, obtida da estatística computada na amostra. Em geral, usamos uma estimativa para estimar o parâmetro populacional.
- **Variável:** *característica mensurável comum a todos os elementos da população.*
  - Usamos letras maiúsculas do alfabeto latino para representar uma variável.
  - Usamos letras minúsculas do alfabeto latino para representar o valor observado da variável em um elemento da amostra.

Exemplo

- **População**: todos os eleitores nas eleições gerais de 2022.
- **Amostra**: 3.500 pessoas abordadas pelo `datafolha`.
- **Variável:** candidato a presidente de cada pessoa.
- **Parâmetro:** porcentagem de pessoas que escolhem Lula como presidente entre todos os eleitores.
- **Estatística:** porcentagem de pessoas que escolhem o lula
- **Estimativa:** porcentagem de pessoas que escolhem Lula como presidente entre todos os eleitores da amostra de 3.500 pessoas entrevistas pelo datafolha.

Classificação de variáveis

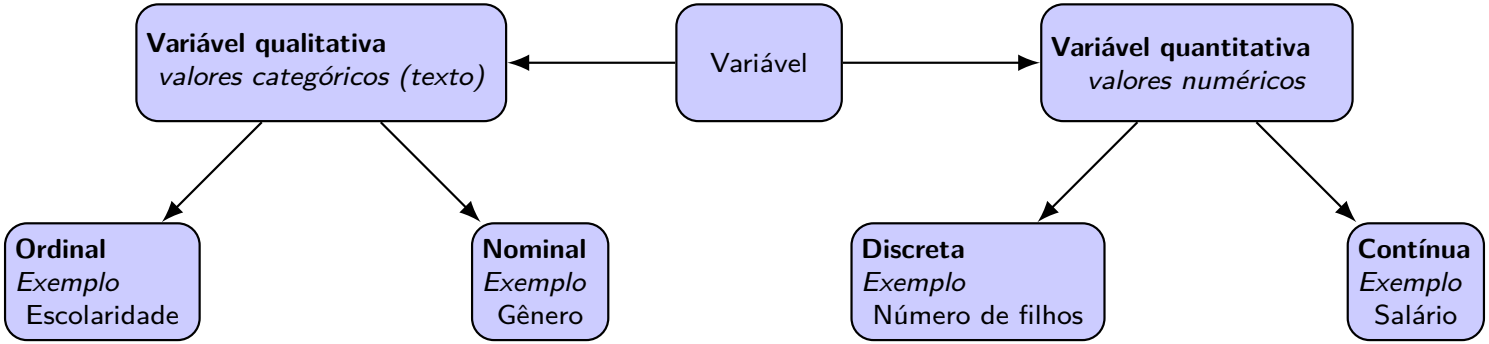


Figura 1: Classificação de variáveis.

Tabela

Tabela de frequência
Variável qualitativa

A primeira coisa que fazemos é contar!

X	frequência	frequência relativa	porcentagem
B_1	n_1	f_1	$100 \cdot f_1\%$
B_2	n_2	f_2	$100 \cdot f_2\%$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_k	n_k	f_k	$100 \cdot f_k\%$
Total	n	1	100%

Em que n é o tamanho da amostra.

Tabela de distribuição de frequências
Variável qualitativa

- **Pacote:** `janitor`.
- `tabyl`: cria a tabela de distribuição de frequências e tem os seguintes parâmetros:
  - `dat`: *data frame* ou vetor com os valores da variável que desejamos tabular.
  - `var1`: nome da primeira variável.
  - `var2`: nome da segunda variável (opcional).
- `adorn_totals`: adiciona uma linha com os totais de cada coluna
- `adorn_pct_formatting`: acrescenta o sinal de porcentagem e tem o seguinte parâmetro:
  - `digits`: o número de casas decimais depois da vírgula
- `rename` (do pacote `dplyr`) muda os nomes das colunas para português no seguinte formato:
  - `"novo nome" = "velho nome"`

Para mais detalhes, consulte a documentação oficial do *janitor*: documentação de `tabyl`.

Tabela de distribuição de frequências
Variável qualitativa

```
dados_iris <- read_xlsx("dados/brutos/iris.xlsx")
tab <- tabyl(dados_iris, especies)  |>
  adorn_totals()  |>
  adorn_pct_formatting(digits = 2) |>
  rename(
    "Espécies" = especies, "Frequência" = n,
    "Porcentagem" = percent
  )
tab
```

```
   Espécies Frequência Porcentagem
     setosa         50      33.33%
 versicolor         50      33.33%
  virginica         50      33.33%
      Total        150     100.00%
```

Tabela de distribuição de frequências
Variável qualitativa
Exercício

Para o conjunto de dados `amostra_enem_salvador.xlsx`, construa a tabela de distribuição de frequências para as seguintes variáveis:

- `tp_sexo`: gênero que a pessoa se identifica (segundo classificação usada pelo IBGE)
- `tp_cor_raca`: raça (segundo classificação usada pelo IBGE)

Tabela de distribuição de frequências
Variável quantitativa discreta

Muito semelhante a tabela de distribuição de frequência para variáveis qualitativas.

X	frequência	frequência relativa	porcentagem
x_1	n_1	f_1	$100 \cdot f_1\%$
x_2	n_2	f_2	$100 \cdot f_2\%$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_k	n_k	f_k	$100 \cdot f_k\%$
Total	n	1	100%

Em que n é o tamanho da amostra e $\{x_1, \ldots, x_k\}$ são os números que são valores únicos de X na amostra.

Tabela de distribuição de frequências
Variável quantitativa discreta

```
dados_mtcarros <- read_csv2("dados/brutos/mtcarros.csv")
tab <- tabyl(dados_mtcarros, carburadores)  |>
  adorn_totals()  |>
  adorn_pct_formatting(digits = 2) |>
  rename(
    "Carburadores" = carburadores, "Frequência" = n,
    "Porcentagem" = percent
  )
tab
```

Carburadores	Frequência	Porcentagem
1	7	21.88%
2	10	31.25%
3	3	9.38%
4	10	31.25%
6	1	3.12%
8	1	3.12%
Total	32	100.00%

Tabela de distribuição de frequências
Variável quantitativa discreta
Exercício

Para o conjunto de dados `amostra_enem_salvador.xlsx`, construa a tabela de distribuição de frequências para a variável `q005`: número de pessoas que moram na casa da(o) candidata(o).

Tabela de frequência
Variável quantitativa contínua

X: variável quantitativa contínua

Tabela 7: Tabela de frequências para a variável quantitativa contínua.

X	Frequência	Frequência relativa	Porcentagem
$[l_0, l_1)$	n_1	$f_1 = \frac{n_1}{n_1+\cdots+n_k}$	$p_1 = f_1 \cdot 100$
$[l_1, l_2)$	n_2	$f_2 = \frac{n_2}{n_1+\cdots+n_k}$	$p_2 = f_2 \cdot 100$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[l_{k-1}, l_k]$	n_k	$f_k = \frac{n_k}{n_1+\cdots+n_k}$	$p_k = f_k \cdot 100$

- menor valor de $X = l_0 \leq l_1 \leq \cdots \leq l_{k-1} \leq l_k =$ maior valor de X
- n_i é número de valores de `X` entre l_{i-1} e l_i
- $l_0, l_1, \ldots, l_k$ quebram o suporte da variável `X` (*breakpoints*).
- $l_0, l_1, \cdots, l_k$ são escolhidos de acordo com a teoria por trás da análise de dados

Recomendações:

- use $l_0, l_1, \cdots, l_k$ igualmente espaçados
- e use a regra de Sturges para determinar o valor de k:
  - $k = 1 + \log 2(n)$ onde n é tamanho da amostra
  - Se $1 + \log 2(n)$ não é um número inteiro, usamos $k = \lceil 1 + \log 2(n) \rceil$.

Tabela de frequência
Variável quantitativa contínua

Primeiro agrupamos os valores em faixas usando a regra de Sturges.

Usamos a função `cut`, com os seguintes argumentos:

- `breaks` - número de intervalos ou os limites dos intervalos;
- `include.lowest` - se `TRUE` inclue o valor à esquerda no intervalo;
- `right` - se `TRUE` inclue o valor à direita no intervalo.

Usamos a função `mutate` para adicionar uma nova coluna em um `tibble`, com os seguintes argumentos:

- `.data` - `tibble` para adicionar uma nova coluna;
- `<nome da variavel> = <vetor>` - adicione uma ou mais colunas separadas por vírgula.

```r
k <- ceiling(1 + log(nrow(dados_iris)))
dados_iris2 <- mutate(
  dados_iris,
  comprimento_sepala_int = cut(
    comprimento_sepala,
    breaks = k,
    include.lowest = TRUE,
    right = FALSE
  )
)
glimpse(dados_iris2)
```

```
Rows: 150
Columns: 6
$ comprimento_sepala     <dbl> 5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0,
$ largura_sepala         <dbl> 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4,
$ comprimento_petala     <dbl> 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5,
$ largura_petala         <dbl> 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2,
$ especies               <chr> "setosa", "setosa", "setosa", "setosa",
$ comprimento_sepala_int <fct> "[4.81,5.33)", "[4.81,5.33)", "[4.3,4.81
```

Tabela de frequência
Variável quantitativa contínua

Agora podemos contar a frequência de cada intervalo.

```
tabyl(dados_iris2, comprimento_sepala_int) |>
  adorn_totals() |>
  adorn_pct_formatting(digits = 2) |>
  rename(
    "Comprimento de sépala" = comprimento_sepala_int,
    "Frequência absoluta" = n,
    "Porcentagem" = percent
  )
```

Comprimento de sépala	Frequência absoluta	Porcentagem
[4.3,4.81)	16	10.67%
[4.81,5.33)	30	20.00%
[5.33,5.84)	34	22.67%
[5.84,6.36)	28	18.67%
[6.36,6.87)	25	16.67%
[6.87,7.39)	10	6.67%
[7.39,7.9]	7	4.67%
Total	150	100.00%

Tabela de frequência
Variável quantitativa contínua
Exercício

Para o conjunto de dados `amostra_enem_salvador.xlsx`, construa as seguintes tabelas de distribuição de frequências:

- `nu_nota_mt` (nota da prova em matemática): $l_0, l_1, \ldots, l_k$ são igualmente espaços com $l_k - l_{k-1} = 100$
- `nu_nota_cn` (nota da prova de ciências humanas): use a regra de Sturges

Departamento de Estatística
IME - UFBA

Gráficos

Departamento de Estatística
IME - UFBA

Gráficos usando ggplot2

- **Pacote:** `ggplot2`.
- Permite gráficos personalizados com uma sintaxe simples e rápida, e iterativa *por camadas*.
- Começamos com um camada com os dados `ggplot(dados)`, e vamos adicionando as camadas de anotações, e sumários estatísticos.
- Usa a *gramática de gráficos* proposta por Leland Wilkinson: Grammar of Graphics.
- Ideia desta gramática: delinear os atributos estéticos das figuras geométricas (incluindo transformações nos dados e mudança no sistema de coordenadas).

Para mais detalhes, você pode consultar `ggplot2`: elegant graphics for data analysis e documentação do `ggplot2`.

Estrutura básica de ggplot2

```
ggplot(data = <data possible tibble>) +
  <Geom functions>(mapping = aes(<MAPPINGS>)) +
  <outras camadas>
```

Você pode usar diversos temas e extensões que a comunidade cria e criou para melhorar a aparência e facilitar a construção de ggplot2.

Lista com extensões do ggplot2: extensões do ggplot2.

Indicação de extensões:

- Temas adicionais para o pacote ggplot2: ggthemes.
- Gráfico de matriz de correlação: ggcorrplot.
- Gráfico quantil-quantil: qqplotr.

Gráfico de barras no `ggplot2`

- **função:** `geom_bar()`. Para porcentagem: `geom_bar(x = <variável no eixo x>, y = after_stat(prop * 100))`.
- Argumentos adicionais:
  - `fill`: mudar a cor do preenchimento das figuras geométricas.
  - `color`: mudar a cor da figura geométrica.
- Rótulos dos eixos
  - **Mudar os rótulos:** `labs(x = <rótulo do eixo x>, y = <rótulo do eixo y>)`.
  - **Trocar o eixo-x pelo eixo-y:** `coord_flip()`.

Gráfico de barras
Variável qualitativa

Gráfico de barras para a variável qualitativa `especies` do conjunto de dados `iris.xlsx`.

```
ggplot(dados_iris) +
  geom_bar(mapping = aes(especies), fill = "blue") +
  labs(x = "Espécies", y = "Frequência") +
  theme_minimal()
```

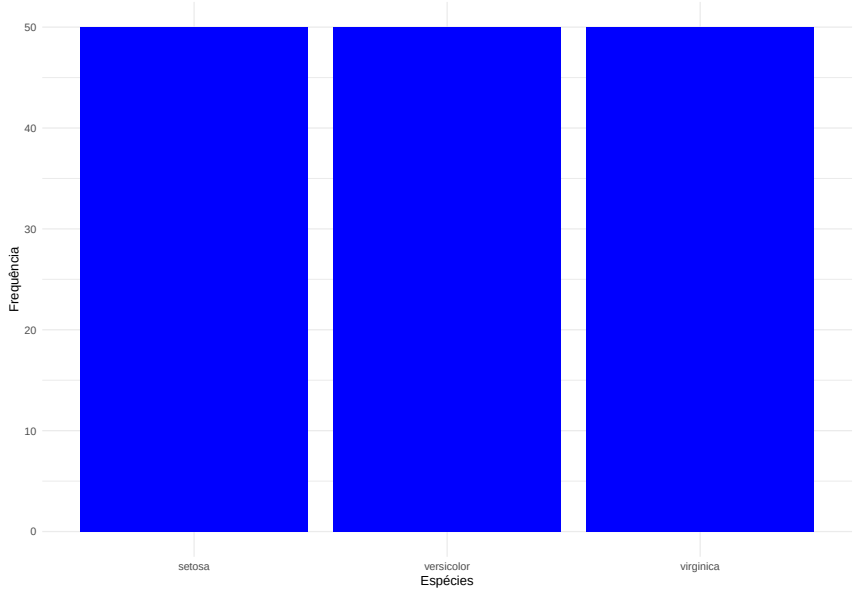


Gráfico de barras
Variável qualitativa
Exercício

Para o conjunto de dados `amostra_enem_salvador.xlsx`, construa o gráfico de barras para as seguintes variáveis:

- `tp_sexo`: gênero que a pessoa se identifica (segundo classificação do IBGE);
- `tp_cor_raca`: raça autodeclarada (segundo classificação do IBGE).

Tabela de distribuição de frequências
Variável quantitativa discreta

De maneira similar, podemos contar quantas vezes cada valor de uma variável quantitativa discreta foi amostrado.

X	frequência	frequência relativa	porcentagem
x_1	n_1	f_1	$100 \cdot f_1\%$
x_2	n_2	f_2	$100 \cdot f_2\%$
x_3	n_3	f_3	$100 \cdot f_3\%$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_k	n_k	f_k	$100 \cdot f_k\%$
Total	n	1	100%

Em que n é o tamanho da amostra.

Tabela de distribuição de frequências
Variável quantitativa discreta

Vamos construir a tabela de distribuição de frequências para a variável quantitativa discreta `carburadores` do conjunto de dados `mtcarros`.

```
tab <- tabyl(dados_mtcarros, carburadores) |>
  adorn_totals() |>
  adorn_pct_formatting(digits = 2) |>
  rename(
    "Número de carburadores" = carburadores,
    "Frequência (absoluta)" = n,
    "Porcentagem" = percent
  )
tab
```

Número de carburadores	Frequência (absoluta)	Porcentagem
1	7	21.88%
2	10	31.25%
3	3	9.38%
4	10	31.25%
6	1	3.12%
8	1	3.12%
Total	32	100.00%

Gráfico de barras
Variável quantitativa discreta

Gráfico de barras para a variável quantitativa discreta `carburadores` do conjunto de dados `mtcarros.csv`.

- `after_stat(prop)` retorna a *frequência relativa* ou *proporção* de um valor (ou categoria) de uma variável.
- `after_stat(count)` retorna a *frequência absoluta* de um valor (ou categoria) de uma variável.

```
ggplot(dados_mtcarros) +
  geom_bar(
    mapping = aes(carburadores, after_stat(100 * prop)),
    fill = "#002f81"
  ) +
  labs(x = "Número de carburadores", y = "Porcentagem") +
  theme_minimal()
```

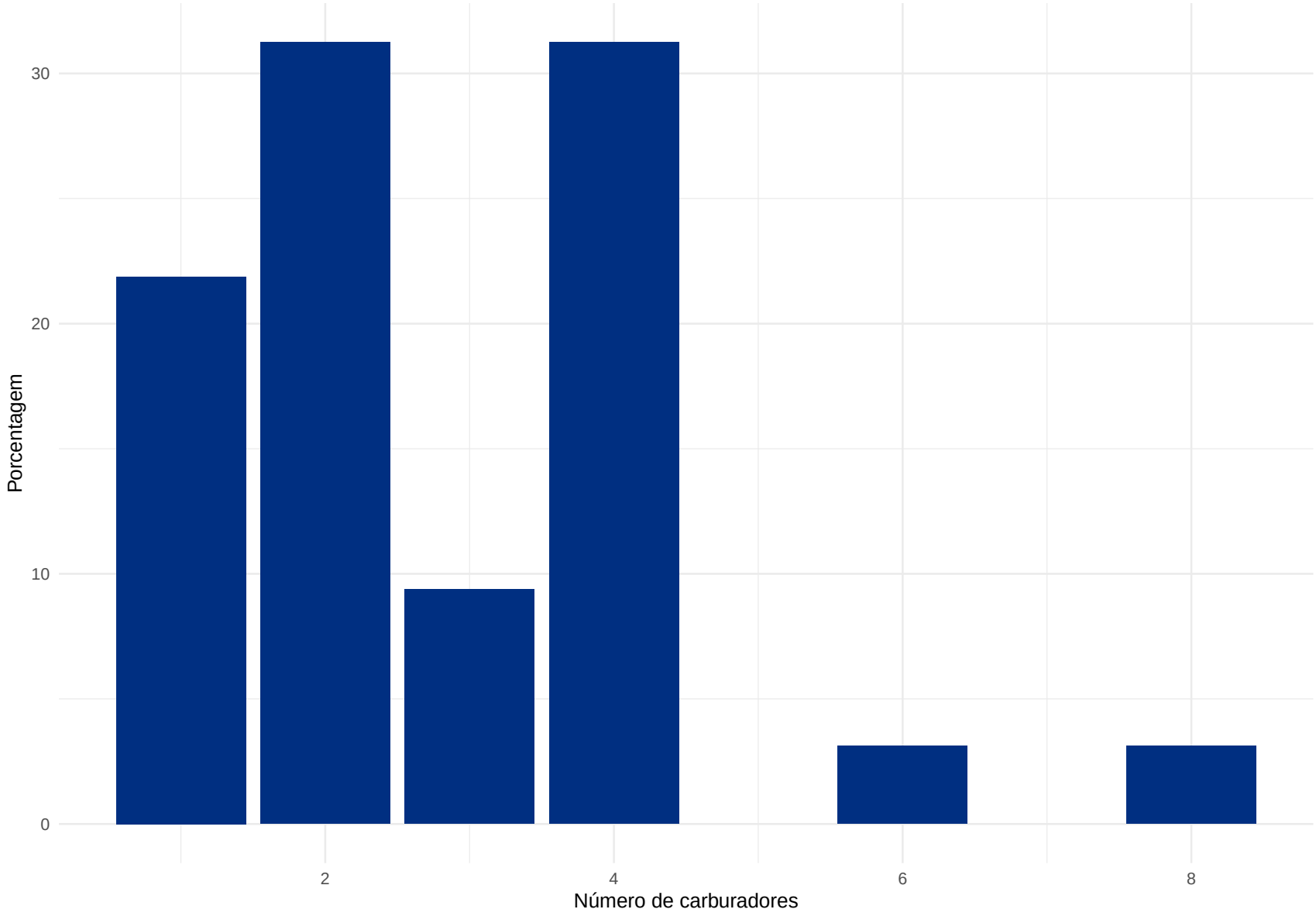


Gráfico de barras
Variável quantitativa discreta
Exercício

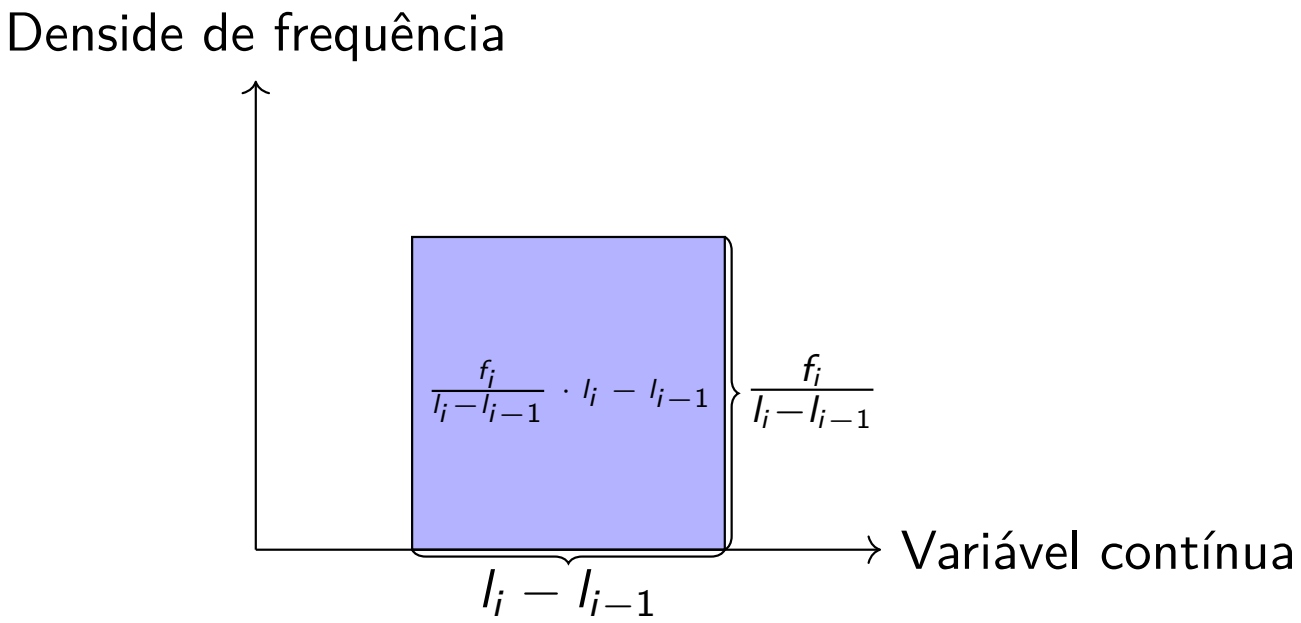
- Para a variável `q005` do conjunto de dados `amostra_enem_salvador.xlsx`, construa o gráfico de barras onde o eixo y é a frequência absoluta.
- Para a variável `q005` do conjunto de dados `amostra_enem_salvador.xlsx`, construa o gráfico de barras onde o eixo y é a frequência relativa.
- Para a variável `q005` do conjunto de dados `amostra_enem_salvador.xlsx`, construa o gráfico de barras onde o eixo y é a porcentagem.

Histograma

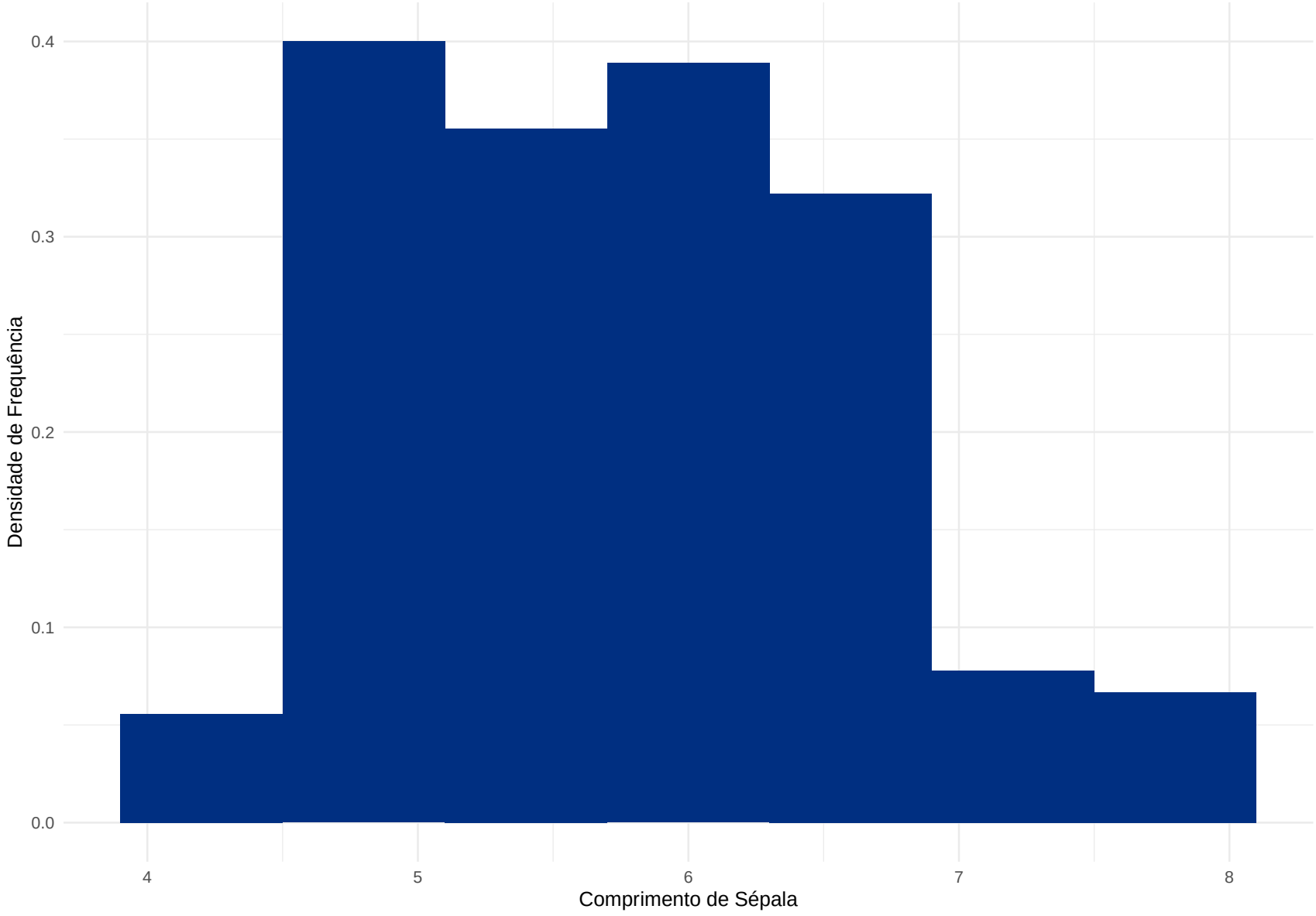
Para variávieis quantitativas contínuas, geralmente não construímos gráficos de barras, e sim uma figura geométrica chamada de *histograma*.

- O histograma é um gráfico de barras contíguas em que a área de cada barra é igual à frequência relativa.
- Cada faixa de valor $[l_{i-1}, l_i), i = 1, \ldots, n$, será representada por um barra com área $f_i, i = 1, \ldots, n$.
- Como cada barra terá área igual a f_i e base $l_i - l_{i-1}$, e a altura de cada barra será $\frac{f_i}{l_i - l_{i-1}}$.
- $\frac{f_i}{l_i - l_{i-1}}$ é denominada de densidade de frequência.
- Podemos usar os seguintes parâmetros (**obrigatório o uso de apenas um deles**):
  - `bins`: número de intervalos no histograma (usando, por exemplo, a regra de Sturges)
  - `binwidth`: tamanho (ou largura) dos intervalos
  - `breaks`: os limites de cada intervalo

Figura 2: Representação de uma única barra de um histograma.



```
ggplot(dados_iris) +
  geom_histogram(
    aes(x = comprimento_sepala, y = after_stat(density)),
    bins = k,
    fill = "#002f81"
  ) +
  theme_minimal() +
  labs(
    x = "Comprimento de Sépala",
    y = "Densidade de Frequência"
  )
```



Histograma
Exercício

- Para a variável `nu_nota_mt` do conjunto de dados `amostra_enem_salvador.xlsx`, construa o histograma onde os intervalos tem o mesmo tamanho igual a 100.
- Para a variável `nu_nota_cn` do conjunto de dados `amostra_enem_salvador.xlsx`, construa o histograma usando a regra de Sturge.

Medidas resumo
Variável quantitativa

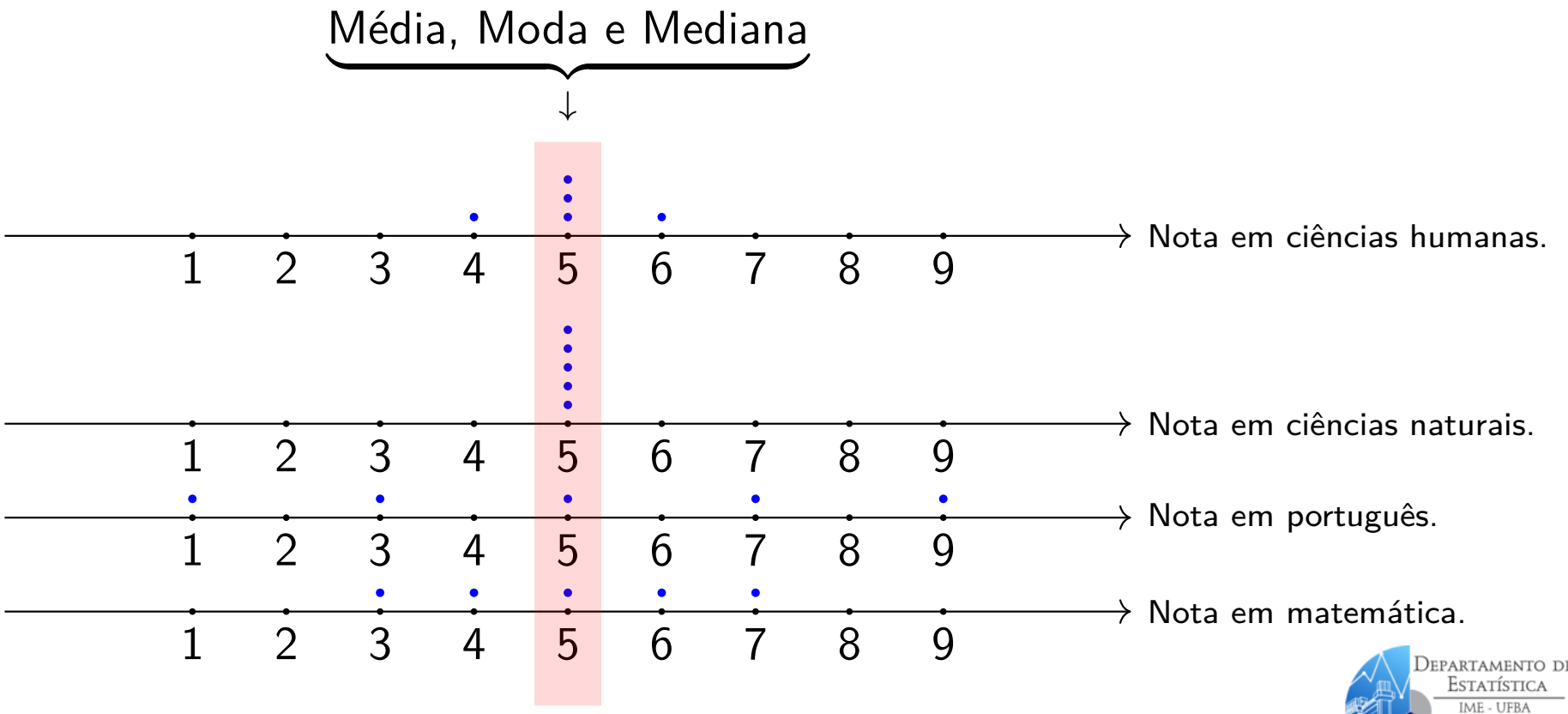
A ideia é encontrar um ou alguns valores que sintetizem todos os valores.

Medidas de posição (tendência central)

A ideia é encontrar um valor que representa *bem* todos os valores.

- **Média:** $\overline{x} = \dfrac{x_1 + \cdots + x_n}{n}$.
- **Mediana:** valor que divide a sequência ordenada de valores em duas partes iguais.
  - Ordene os valores do menor ao maior;
  - Valor que divide os valores entre os 50% menores e os 50% maoires:
    - 50% dos valores x_i satisfazem: $x_i \leq$ Mediana;
    - 50% dos valores x_i satisfazem: $x_i \geq$ Mediana.

Figura 3: Representação gráfica para nota em matemática, português, ciências naturais e ciências humanas.



A variáveis *nota em matemática*, *nota em português*, *nota em ciências naturais*, e *nota em ciências humanas* têm a mesma média, moda e mediana, mas as variáveis não são guais.

Precisamos analisar como os valores são distribuídos.

Medidas de dispersão

A ideia é medir a homogeneidade dos valores.

- **Variância:** $s^2 = \dfrac{(x_1 - \overline{x})^2 + \cdots + (x_n - \overline{x})^2}{n-1}$.
- **Desvio padrão:** $s = \sqrt{s^2}$ (mesma unidade dos dados).
- **Coeficiente de variação** $cv = \dfrac{s}{\overline{x}} \cdot 100\%$ (adimensional, ou seja, "sem unidade").

Medidas resumo: exemplo

Podemos usar a função `summarise` do pacote `dplyr` (incluso no pacote `tidyverse`).

```
dados_iris |>
  summarise(
    media = mean(comprimento_sepala),
    mediana = median(comprimento_sepala),
    dp = sd(comprimento_sepala),
    cv = dp / media
  )
```

```
# A tibble: 1 x 4
  media mediana    dp    cv
  <dbl>   <dbl> <dbl> <dbl>
1  5.84     5.8 0.828 0.142
```

Medidas resumo: exemplo

Podemos usar a função group_by para calcular medidas resumo por categorias de uma variável qualitativa.

```
tabela <- dados_iris |>
  group_by(especies) |>
  summarise(
    media = mean(comprimento_sepala),
    mediana = median(comprimento_sepala),
    dp = sd(comprimento_sepala),
    cv = dp / media
  )
tabela
```

```
# A tibble: 3 x 5
  especies   media mediana    dp      cv
  <chr>      <dbl>   <dbl> <dbl>  <dbl>
1 setosa      5.01     5   0.352 0.0704
2 versicolor  5.94     5.9 0.516 0.0870
3 virginica   6.59     6.5 0.636 0.0965
```

Medidas de resumo
Exercício

- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_mt` do conjunto de dados `amostra_enem_salvador.xlsx` por gênero (`tp_sexo`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_cn` do conjunto de dados `amostra_enem_salvador.xlsx` por gênero (`tp_sexo`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_mt` do conjunto de dados `amostra_enem_salvador.xlsx` por raça (`tp_cor_raca`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_cn` do conjunto de dados `amostra_enem_salvador.xlsx` por raça (`tp_cor_raca`).

Ideia

$q(p)$ é um valor que satisfaz;

- $100 \cdot p\%$ das observações x_i satisfazem $x_i \leq q(p)$
- $100 \cdot (1-p)\%$ das observações satisfazem $x_i \geq q(1-p)$

Alguns quantis especiais

- *Primeiro quartil:* $q_1 = q(0,25)$
- *Primeiro quartil:* $q_2 = q(0,5)$
- *Primeiro quartil:* $q_3 = q(0,75)$

```r
dados_iris |>
  group_by(especies) |>
  summarise(
    q1 = quantile(comprimento_sepala, 0.25),
    q2 = quantile(comprimento_sepala, 0.5),
    q3 = quantile(comprimento_sepala, 0.75),
    frequencia = n()
  )
```

```
# A tibble: 3 x 5
  especies      q1    q2    q3 frequencia
  <chr>      <dbl> <dbl> <dbl>      <int>
1 setosa       4.8     5   5.2         50
2 versicolor   5.6     5.9 6.3         50
3 virginica    6.22    6.5 6.9         50
```

`n()` calcula a frequência de cada valor de uma variável qualitativa.

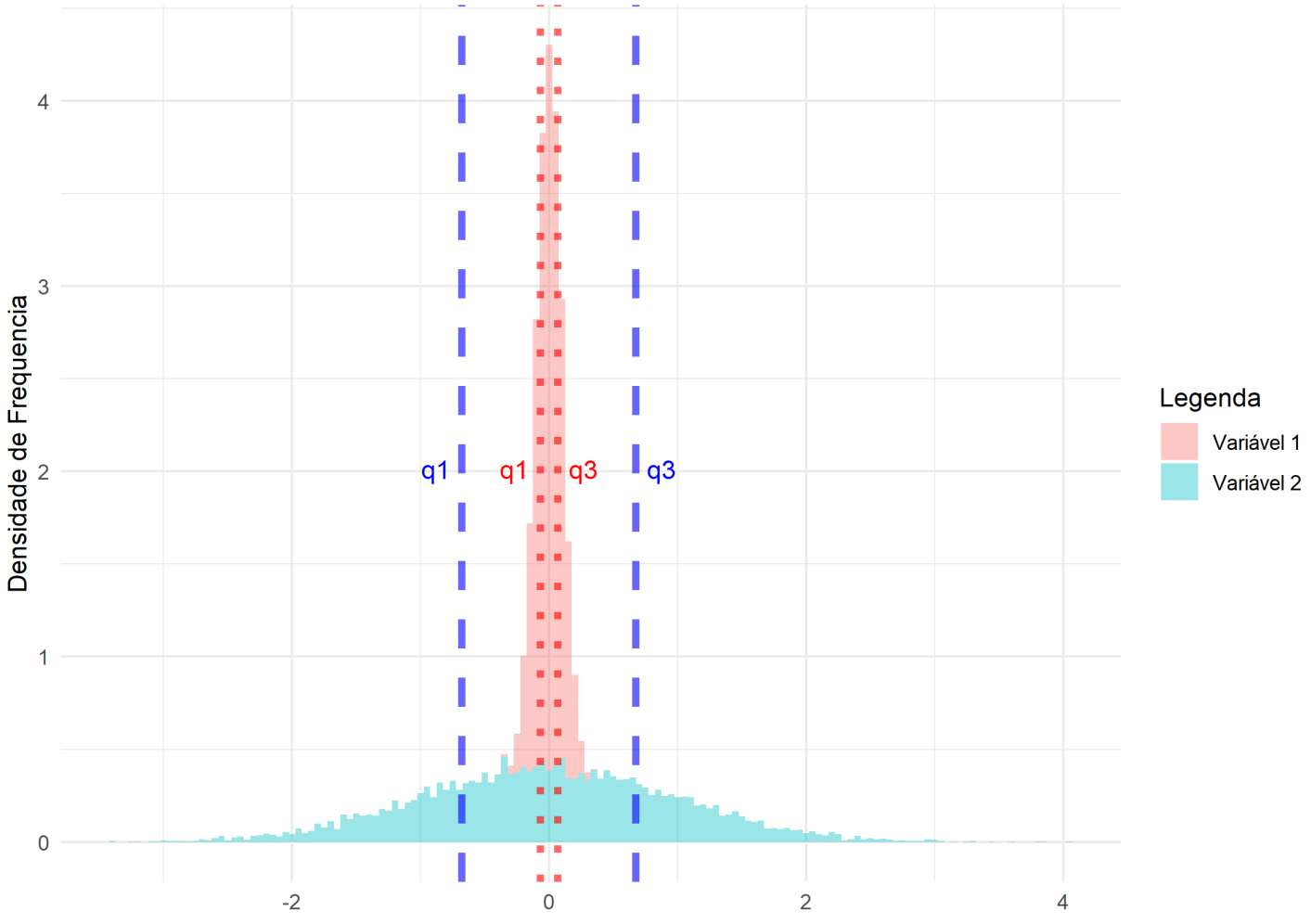
Quantis
Exercício

- Calcule o primeiro quartil, segundo quartil e o terceiro quartil para a variável `nu_nota_mt` do conjunto de dados `amostra_enem_salvador.xlsx` por gênero (`tp_sexo`). Inclua uma coluna com a frequência da variável `tp_sexo`.
- Calcule o primeiro quartil, segundo quartil e o terceiro quartil para a variável `nu_nota_cn` do conjunto de dados `amostra_enem_salvador.xlsx` por gênero (`tp_sexo`). Inclua uma coluna com a frequência da variável `tp_sexo`.
- Calcule o primeiro quartil, segundo quartil e o terceiro quartil para a variável `nu_nota_mt` do conjunto de dados `amostra_enem_salvador.xlsx` por raça (`tp_cor_raca`). Inclua uma coluna com a frequência da variável `tp_cor_raca`.
- Calcule o primeiro quartil, segundo quartil e o terceiro quartil para a variável `nu_nota_cn` do conjunto de dados `amostra_enem_salvador.xlsx` por raça (`tp_cor_raca`). Inclua uma coluna com a frequência da variável `tp_cor_raca`.

Diagrama de caixa (ou *boxplot*)

Medida de dispersão: distância entre q_3 e q_1

Diferença de quartis: $dq = q_3 - q_1$



Departamento de Estatística
IME - UFBA

Assimetria à direita ou positiva:

- frequências diminuem à direita no histograma
- q_2 perto q_1: $q_2 - q_1 < q_3 - q_2$

Assimetria à esquerda ou negativa: frequências diminuem à esquerda no histograma

- frequências diminuem à direita no histograma
- q_2 perto q_3: $q_2 - q_1 > q_3 - q_2$

Diagrama de caixa (ou *boxplot*)

Assimetria

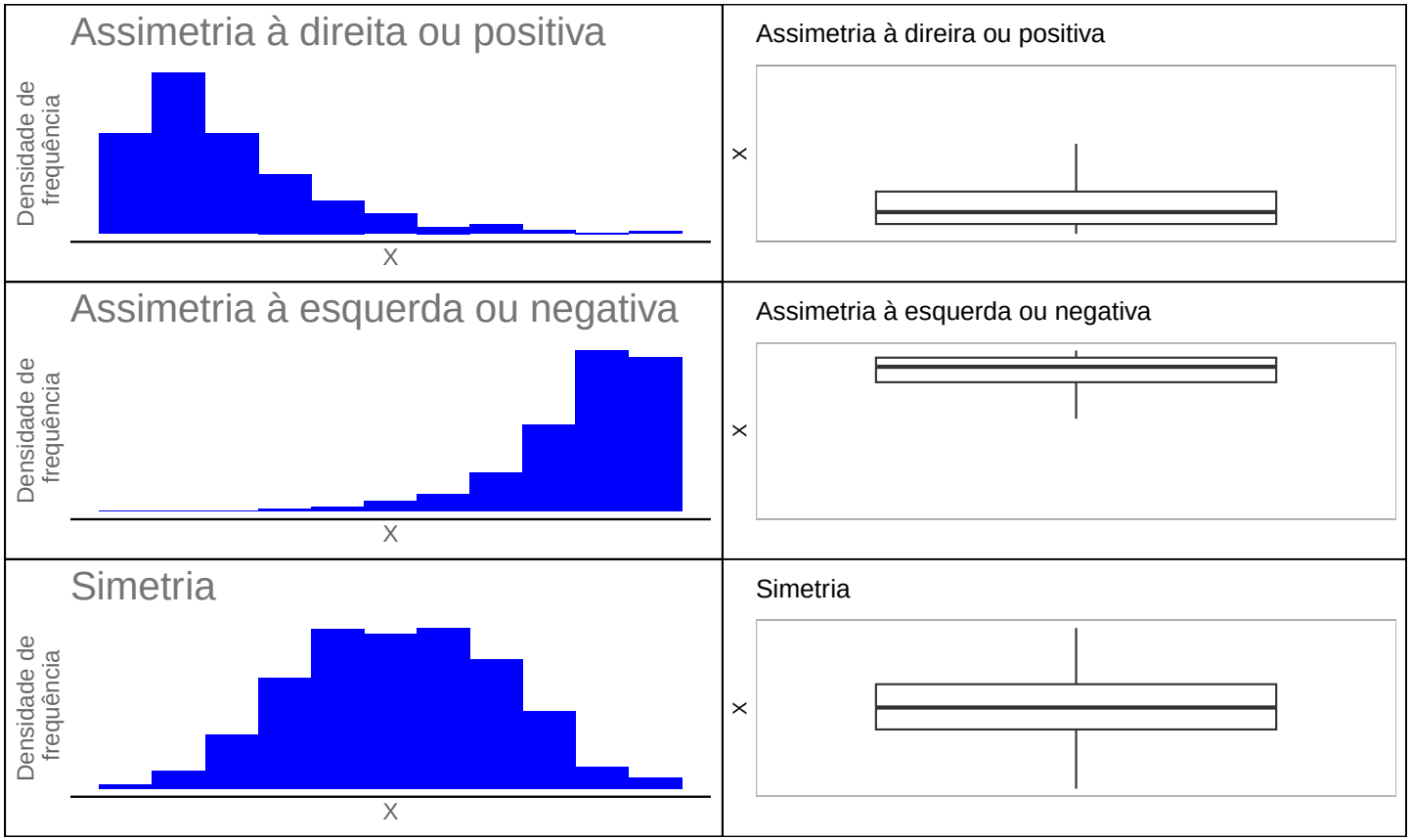


Diagrama de caixa (ou *boxplot*)

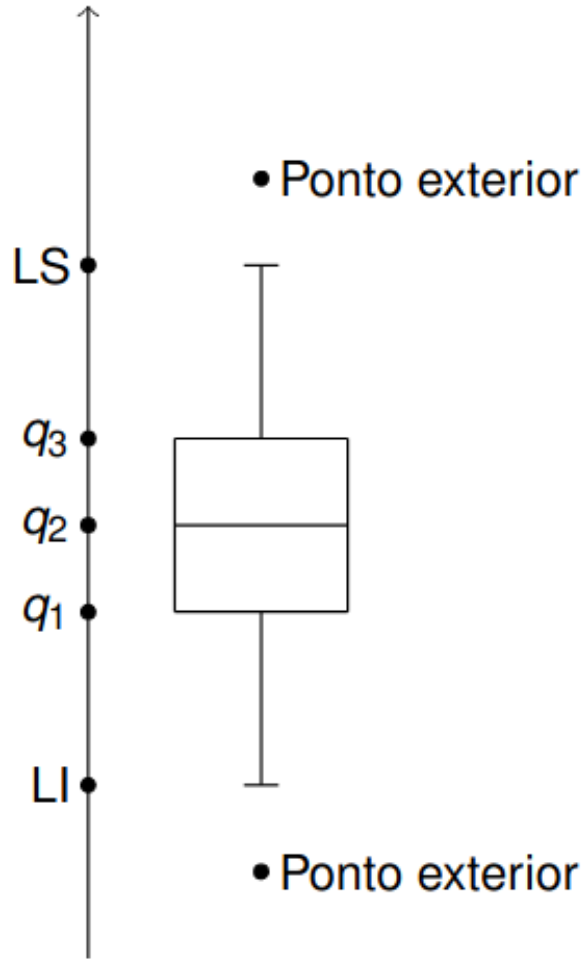
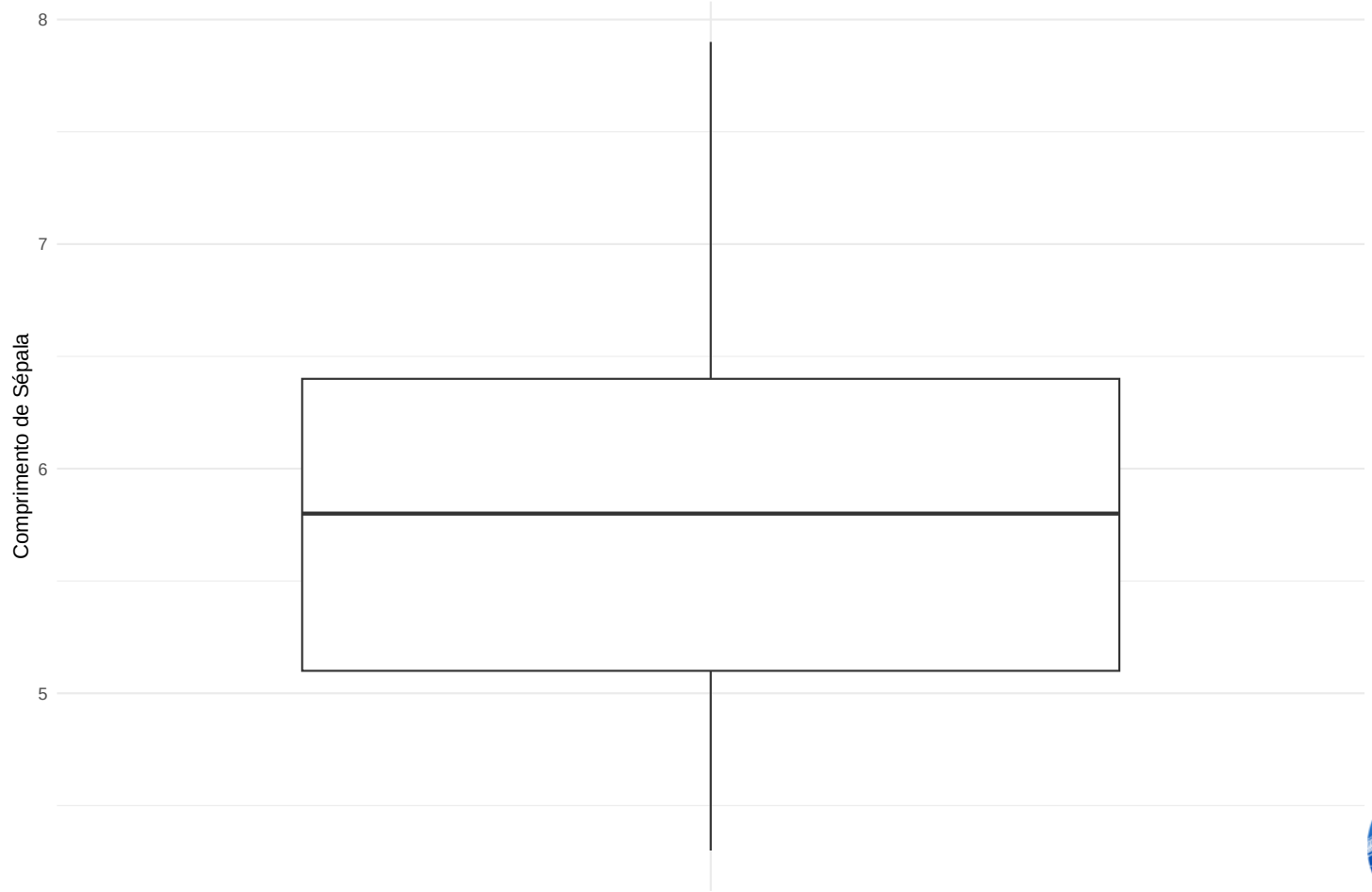


Diagrama de caixa (ou *boxplot*)

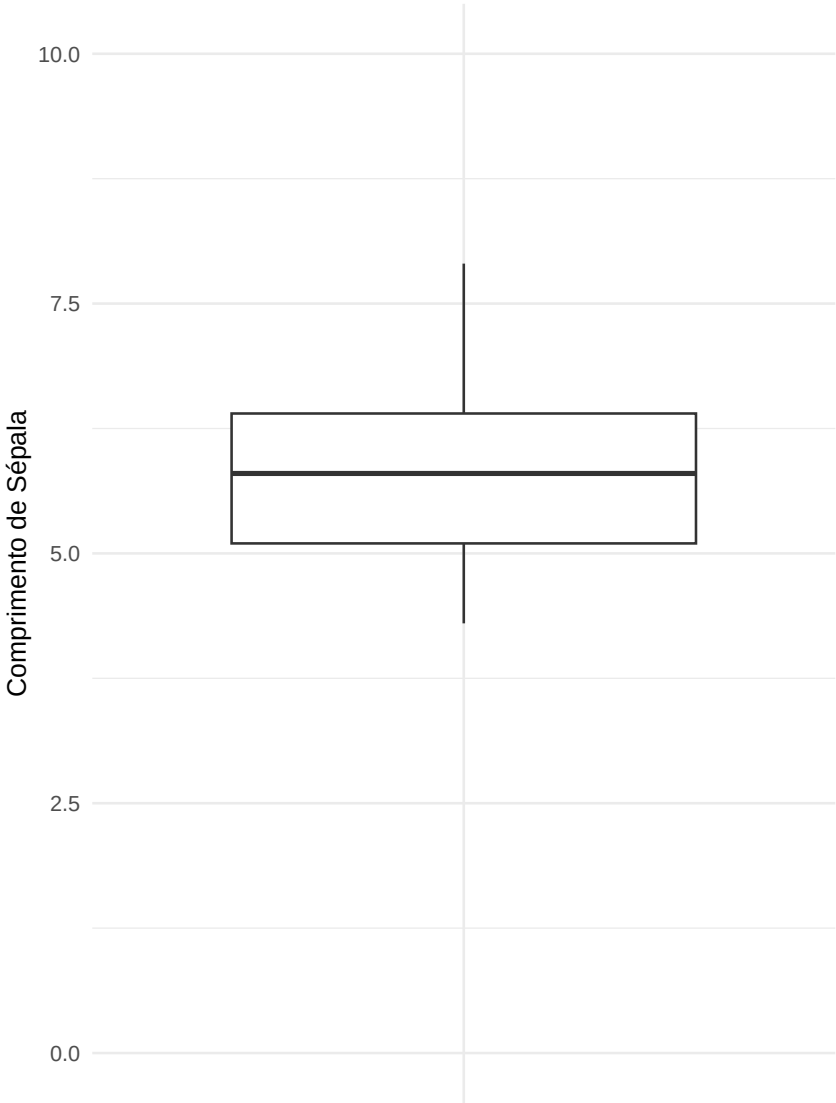
```
ggplot(dados_iris) +
  geom_boxplot(aes(x = "", y = comprimento_sepala)) +
  labs(x = "", y = "Comprimento de Sépala") +
  theme_minimal()
```



Gráficos lado a lado com patchwork

- patchwork permite que colocar gráficos lado a lado com
  - +: figuras ao lado
  - \: figuras embaixo
- Para mais detahes, visite a documentação do patchwork

```
sepala <- ggplot(dados_iris) +
  geom_boxplot(aes(x = "", y = comprimento_sepala)) +
  labs(x = "", y = "Comprimento de Sépala") +
  ylim(c(0, 10)) +
  theme_minimal()
petala <- ggplot(dados_iris) +
  geom_boxplot(aes(x = "", y = comprimento_petala)) +
  labs(x = "", y = "Comprimento de Pétala") +
  ylim(c(0, 10)) +
  theme_minimal()
sepala + petala
```



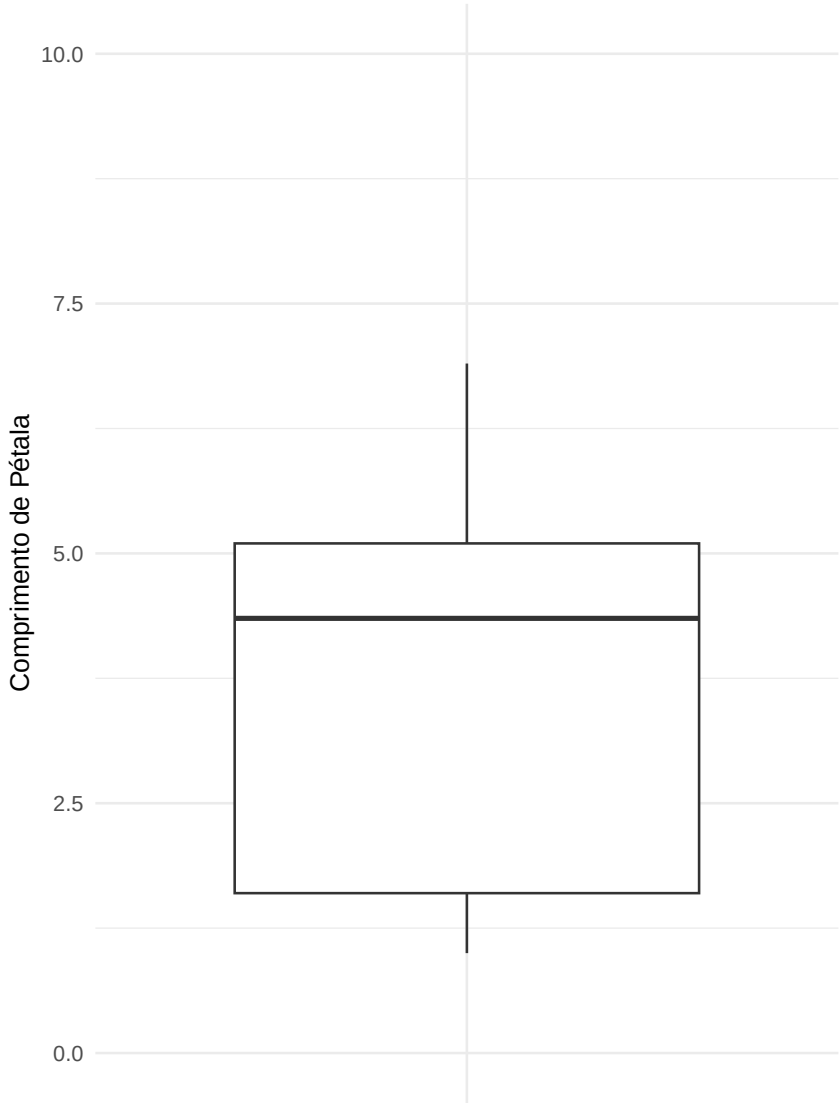


Diagrama de caixa
Exercício

Para o conjunto de dados `amostra_enem_salvador.xlsx`, construa o diagrama de caixa para as variáveis `nu_nota_mt` e `nu_nota_cn` e os coloque lado a lado usando o pacote `patchwork`.

Associção entre duas variáveis

Departamento de Estatística
IME - UFBA

Gráficos
Duas variáveis

Ideia: estudar a associação entre duas variáveis quantitativas.

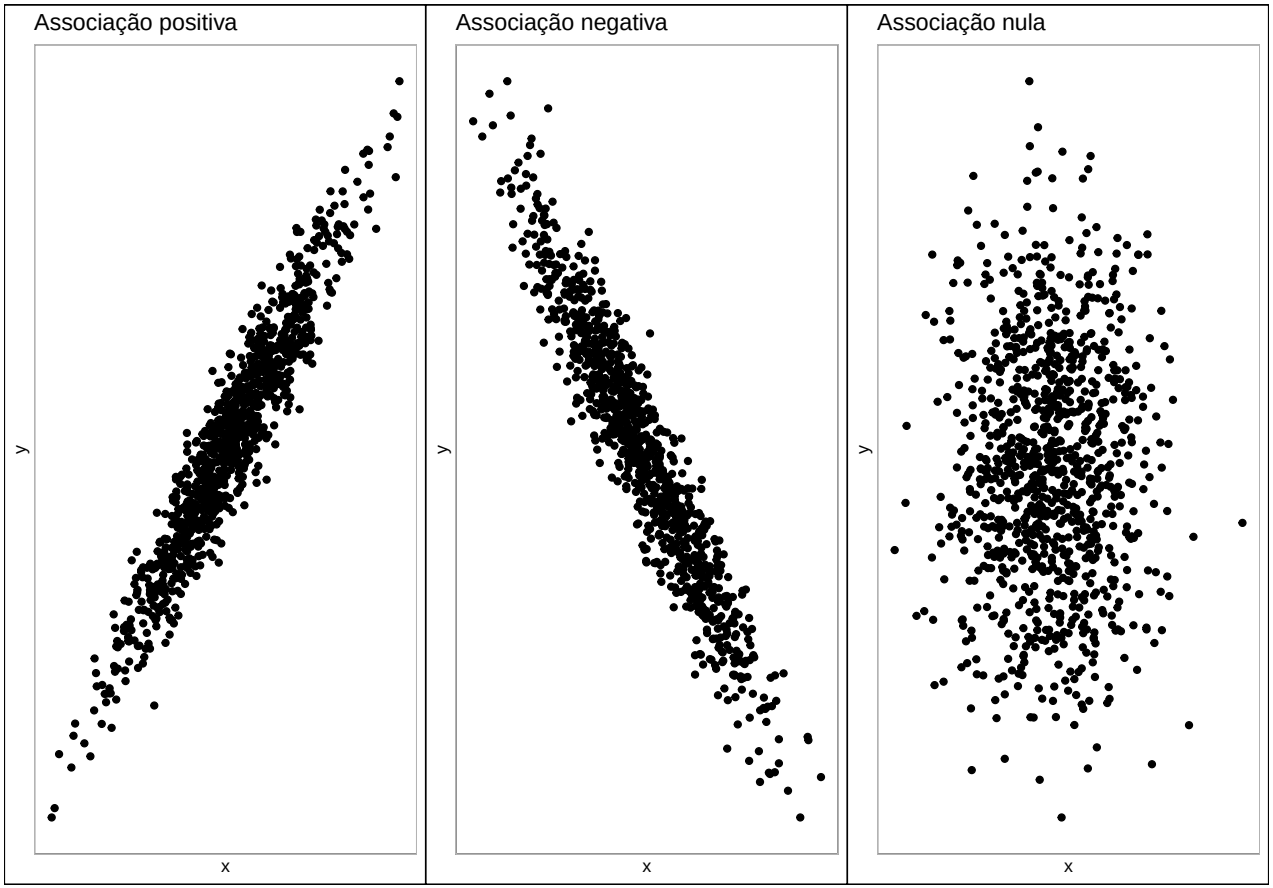


Gráfico de dispersão

```
ggplot(dados_iris) +
  geom_point(aes(comprimento_petala, comprimento_sepala)) +
  labs(
    x = "Comprimento de pétala",
    y = "Comprimento de sépala"
  ) +
  theme_minimal()
```

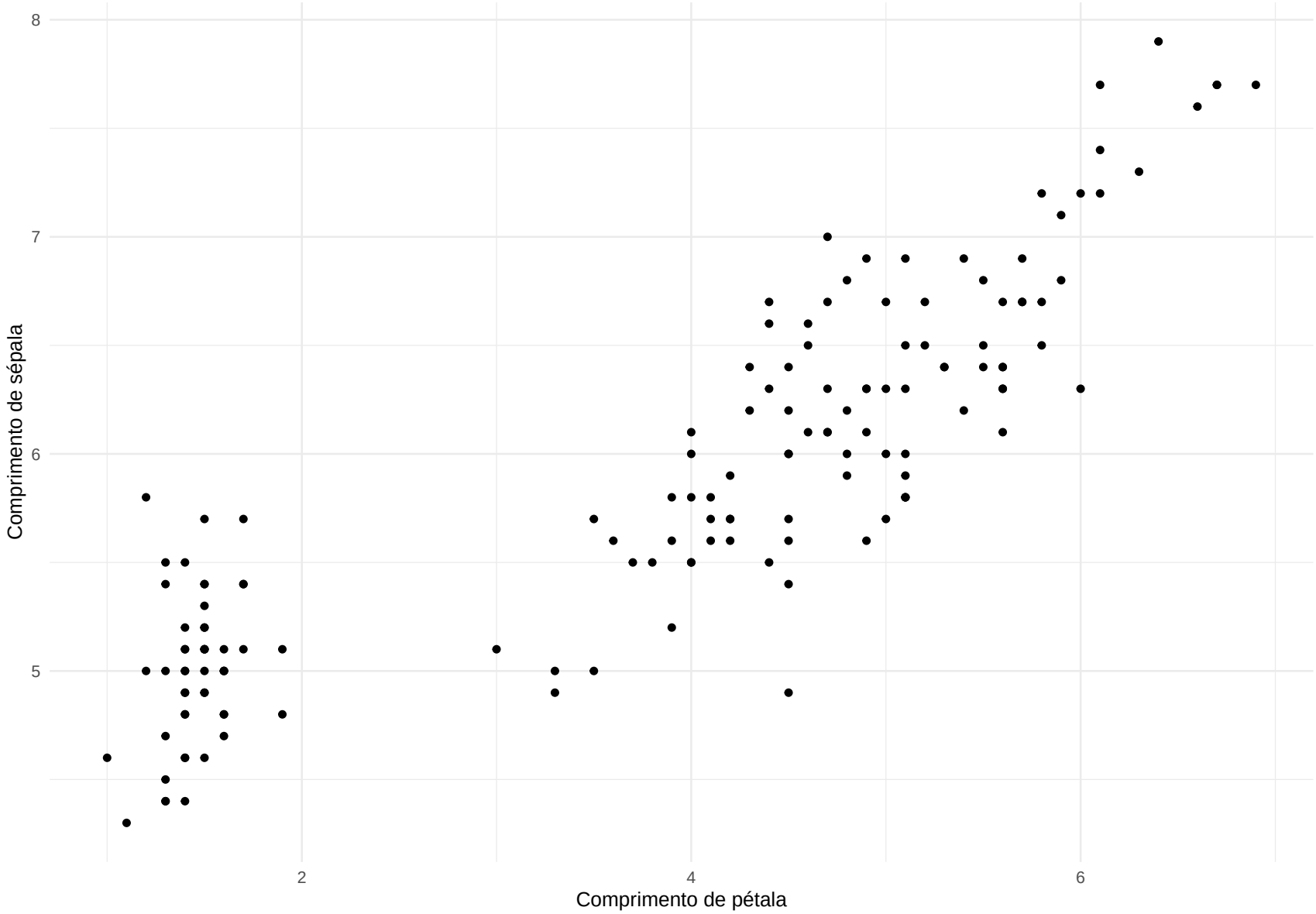


Gráfico de dispersão
Exercício

Para o conjunto de dados `amostra_enem_salvador.xlsx`, construa o gráfico de dispersão entre as variáveis `nu_nota_mt` e `nu_nota_cn`.

Inclua o argumento nomeado `alpha = 0.1` na função `geom_point` para incluir opacidade no gráfico de dispersão. Isso ajuda quando temos amostra de tamanho médio e grande.

Associação entre duas variáveis qualitativas

Ideia

Sejam X e Y duas variáveis qualitativas com os seguintes valores possíveis:

- $X : A_1, \cdots, A_r$
- $Y : B_1, \cdots, B_s$

Desejamos estudar a associação entre X e Y.

Associação entre X e Y

Suponha que A_i tenha porcentagem $100 \cdot f_i \cdot \%$. Então, X e Y são:

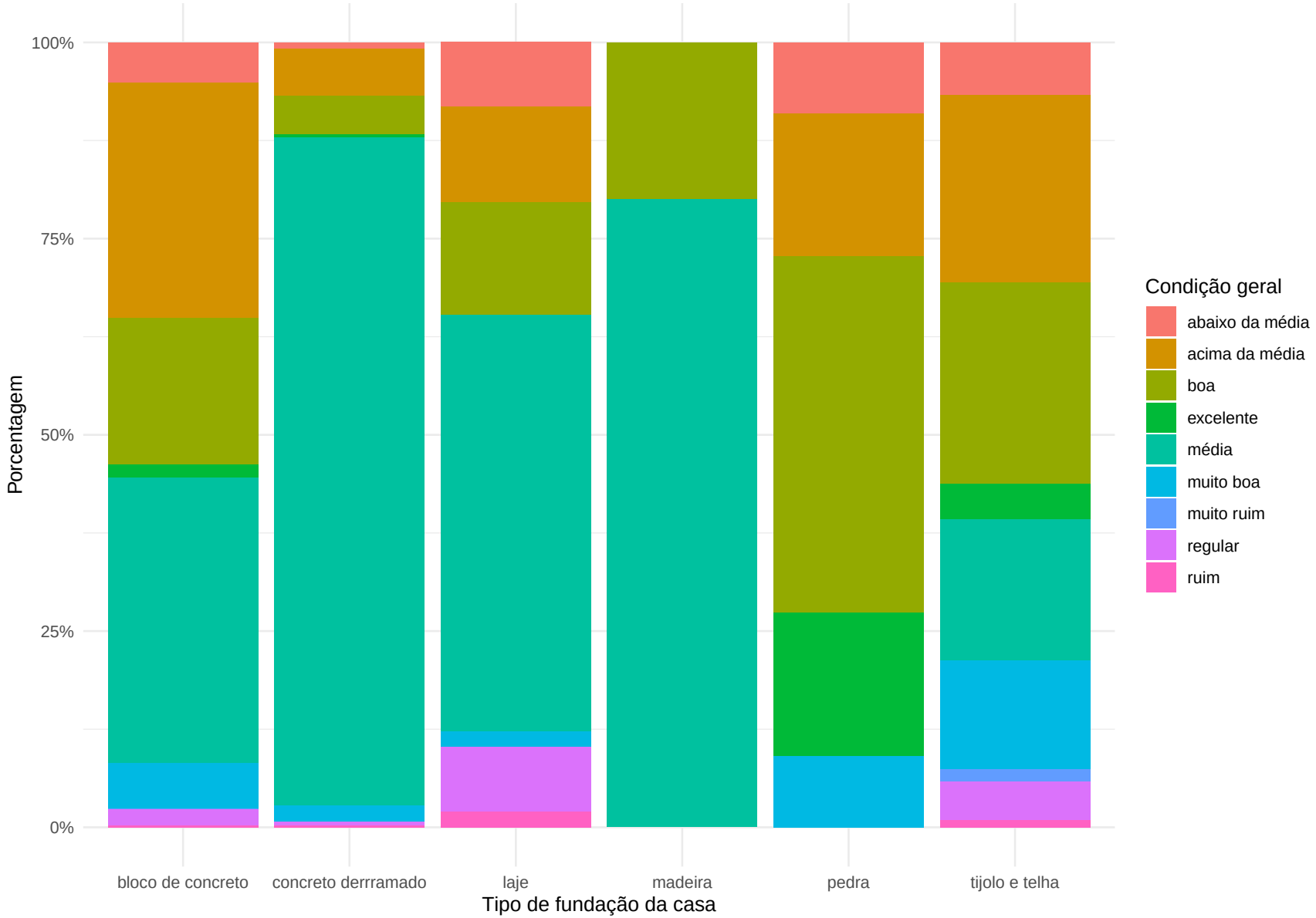
- **não associados:** se ao conhecermos o valor de Y para um elemento da população, **continuamos** com a porcentagem $100 \cdot f_i\%$ deste elemento ter valor de X igual a A_i
- **associados:** se ao conhecermos o valor de Y para um elemento da população, **alteramos** a porcentagem $100 \cdot fi\%$ deste elemento ter valor de X igual a A_i

Departamento de Estatística
IME - UFBA

Associação entre duas variáveis qualitativas
Gráfico de barras

Vamos checar a associação entre `fundacao_tipo` e `geral_condicao`.

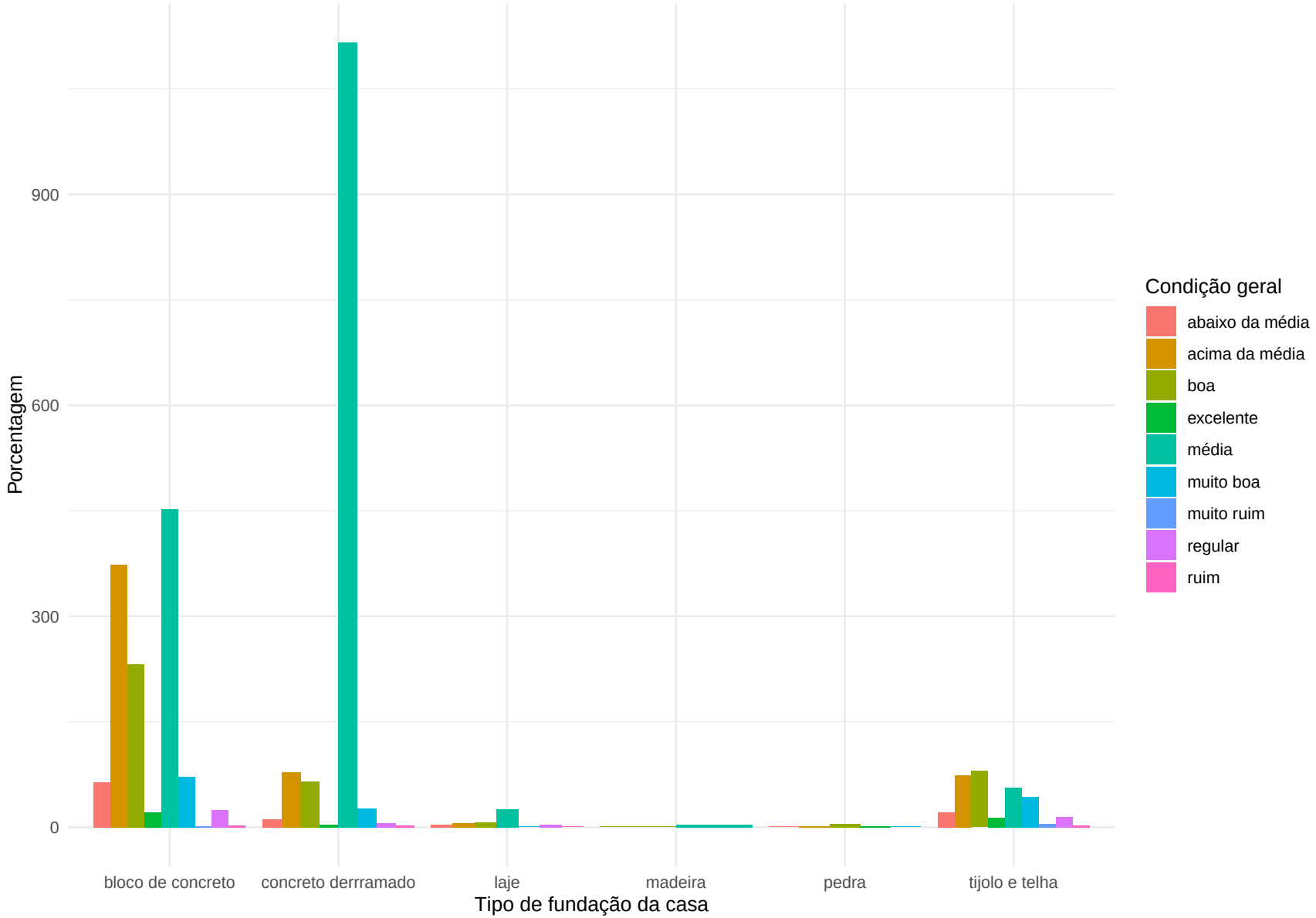
```
dados_casas <- read_xlsx("dados/brutos/casas.xlsx")
ggplot(dados_casas) +
  geom_bar(aes(x = fundacao_tipo, fill = geral_condicao),
           position = "fill") +
  labs(x = "Tipo de fundação da casa", y = "Porcentagem",
       fill = "Condição geral") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal()
```



Associação entre duas variáveis qualitativas
Gráfico de barras

Podemos agrupar as barras por grupos para analisar a associação entre duas variáveis qualitativas.

```
dados_casas <- read_xlsx("dados/brutos/casas.xlsx")
ggplot(dados_casas) +
  geom_bar(aes(x = fundacao_tipo, fill = geral_condicao),
           position = "dodge") +
  labs(x = "Tipo de fundação da casa", y = "Porcentagem",
       fill = "Condição geral") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal()
```



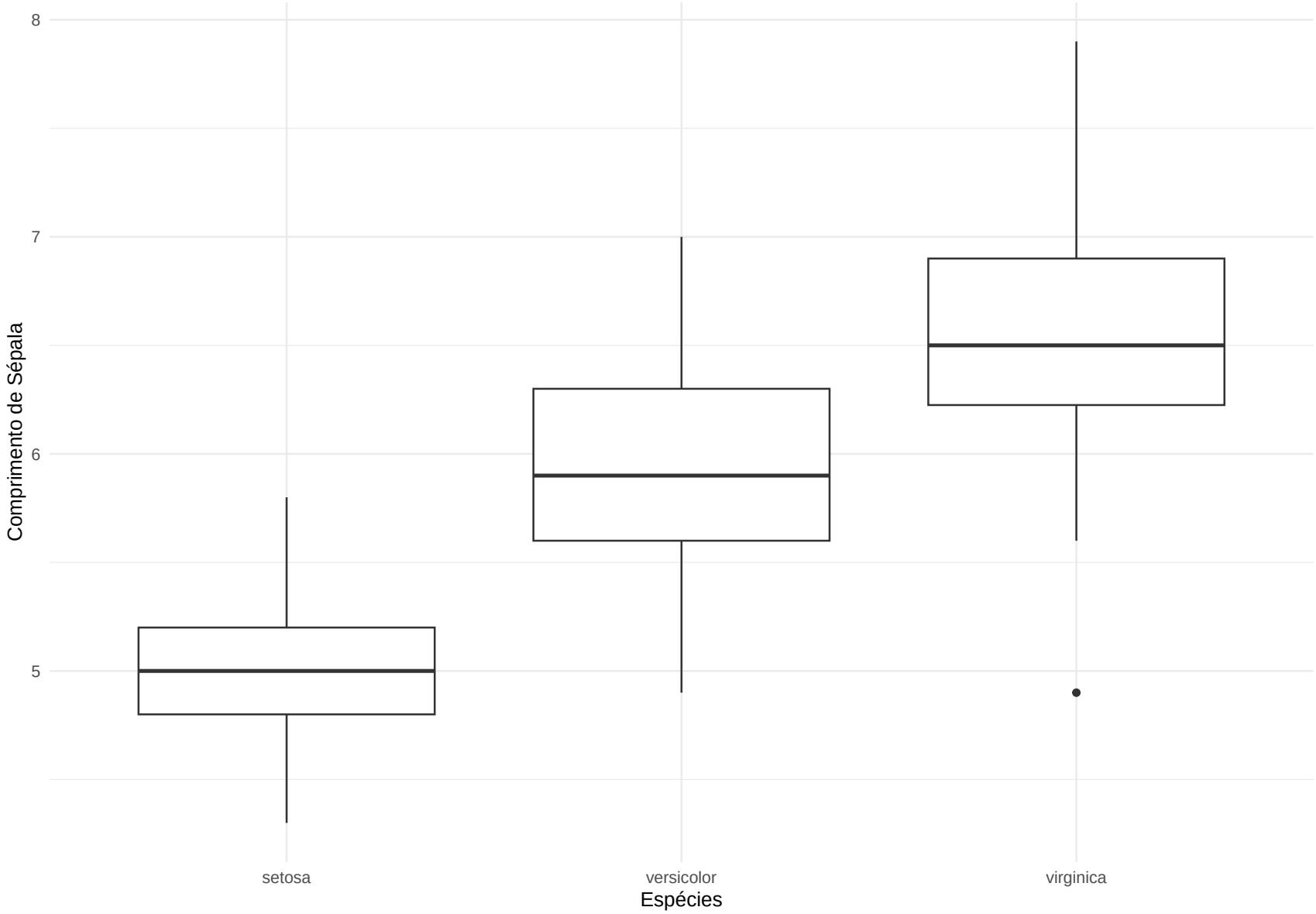
Associação entre duas variáveis qualitativas
Gráfico de barras
Exercício

- Verifique se existe associação entre as variáveis `q006` e `tp_cor_raca` do conjunto de dados `amostra_enem_salvador.xlsx` usando gráfico de gráficos usando o `position=fill`.
- Verifique se existe associação entre as variáveis `q006` e `tp_sexo` do conjunto de dados `amostra_enem_salvador.xlsx` usando gráfico de gráficos usando o `position=dodge`.

Comparação de medianas usando Diagrama de caixa

Podemos comparar medianas de diferentes grupos usando o diagrama de caixa.

```
ggplot(dados_iris) +
  geom_boxplot(aes(x = especies, y = comprimento_sepala)) +
  labs(x = "Espécies", y = "Comprimento de Sépala") +
  theme_minimal()
```



Comparação de medianas usando Diagrama de caixa
Exercício

- Para o conjunto de dados `amostra_enem_salvador.xlsx`, compare a variável `nu_nota_mt` por raça (`tp_cor_raca`).
- Para o conjunto de dados `amostra_enem_salvador.xlsx`, compare a variável `nu_nota_cn` por raça (`tp_cor_raca`).
- Coloque os dois gráficos acima lado a lado usando o pacote `patchwork`.

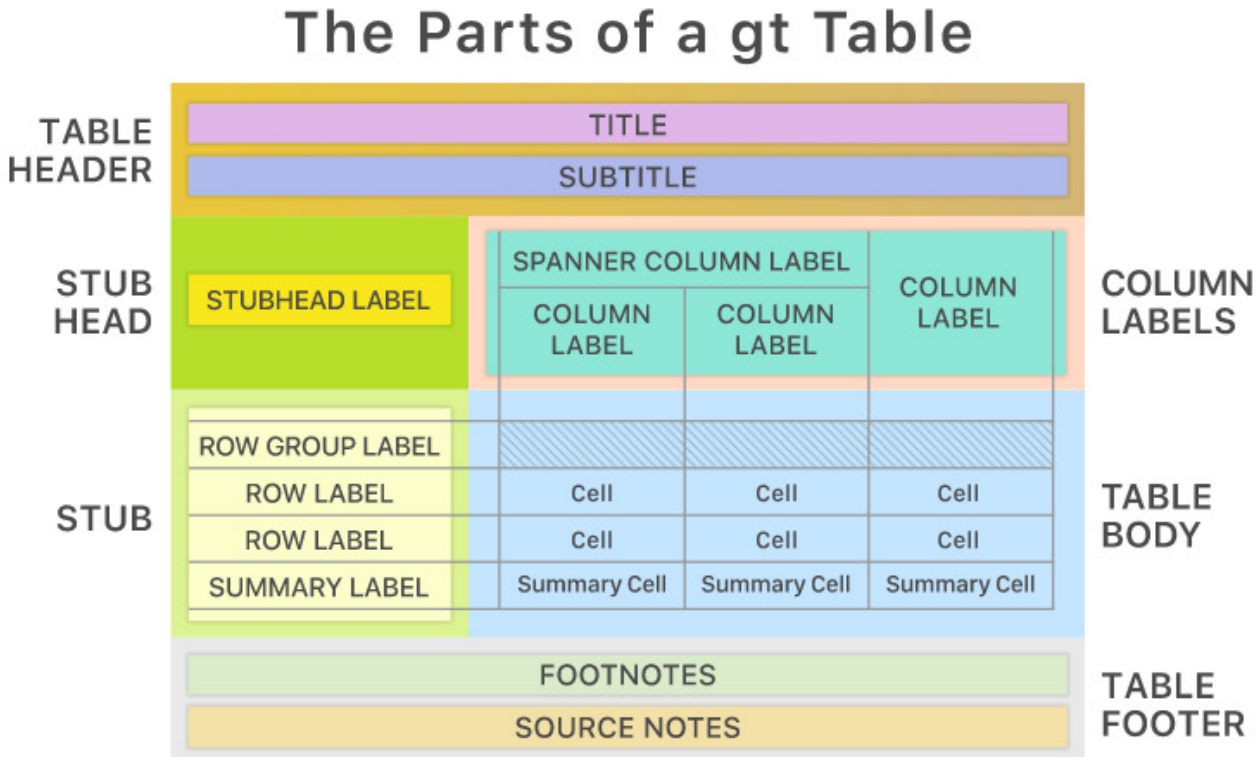
Customizando tabelas usando o pacote gt

Departamento de Estatística
IME - UFBA

Salvando tabelas com o pacote gt

Vamos usar o pacote gt para customizar a apresentação de uma tabela.

A ideia do pacote gt é melhorar apresentação por camadas.



Para mais detalhes, visite documentação do pacote gt

Salvando tabelas com o pacote gt

Vamos usar um exemplo para ensinar como usar o pacote gt.

```
tab <- dados_iris |>
  group_by(especies) |>
  summarise(
    m_petala = mean(comprimento_petala),
    dp_petala = sd(comprimento_petala),
    q1_petala = quantile(comprimento_petala, probs = 0.25),
    q2_petala = quantile(comprimento_petala, probs = 0.5),
    q3_petala = quantile(comprimento_petala, probs = 0.75),
    cv_petala = dp_petala / m_petala
  )
tab
```

```
# A tibble: 3 x 7
  especies   m_petala dp_petala q1_petala q2_petala q3_petala cv_petala
  <chr>         <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
1 setosa         1.46     0.174       1.4       1.5       1.58     11.9
2 versicolor     4.26     0.470       4         4.35      4.6      11.0
3 virginica      5.55     0.552       5.1       5.55      5.88      9.94
```

Cabeçalho da tabela: legenda e sub-legenda da tabela.

- `tab_header`: permite incluir legenda (`title`) e sub-legenda na tabela (`subtitle`)
- `gtsave`: permite salvar objeto `gt`nos formatos `.html`, `.tex` e `.docx`.
- `md`: permite formatação usando a sintaxe `markdown`.
  - Para mais detalhes sobre markdown, consulte *cheatsheet* do markdown

```
gt_tab <- gt(tab) |>
  tab_header(
    title = md("**Comprimento de pétala**"),
    subtitle = md("_Algumas estatísticas descritivas_")
  )
gtsave(gt_tab, "output/tabela.html")
gtsave(gt_tab, "output/tabela.tex")
gtsave(gt_tab, "output/tabela.docx")
```

Salvando tabelas com o pacote gt
Exercício

1. Calcule a média, o desvio padrão, o primeiro quartil, o segundo quartil e o terceiro quartil para a variável `nu_nota_mt` por raça (`tp_cor_raca`) do conjunto de dados `amostra_enem_salvador.xlsx`e salve o resultado em objeto `tab`.
2. Crie um objeto `gt` com nome `gt_tab` a partir da tabela em `tab`.
3. Inclua uma legenda com o texto "Nota em matemática por raça" e sublegenda "Edição 2021" com a função `tab_header`.

Salvando tabelas com o pacote gt

- `tab_source`: inclusão de _fonte de dados_dentes

```
gt_tab <- gt_tab |>
  tab_source_note(
    source_note = md("**Fonte:** Elboração própria.")
  )
gt_tab
```

Comprimento de pétala

Algumas estatísticas descritivas

especies	m_petala	dp_petala	q1_petala	q2_petala	q3_petala	cv_petala
setosa	1.462	0.1736640	1.4	1.50	1.575	11.878522
versicolor	4.260	0.4699110	4.0	4.35	4.600	11.030774
virginica	5.552	0.5518947	5.1	5.55	5.875	9.940466

Fonte: Elboração própria.

Salvando tabelas com o pacote gt
Exercício

Inclua *fonte de dados* usando a função `tab_source_note` como texto "Fonte: elaboração própria." no objeto `gt_tab`.

Rótulo (legenda) para grupo de linhas

`tab_row_group`: permite colocar um *rótulo* para um grupo de linhas.

```
gt_tab <- gt_tab |>
  tab_row_group(
    rows = c(1, 3),
    label = md("_Espécies principais_")
  )
gt_tab
```

Comprimento de pétala

Algumas estatísticas descritivas

especies	m_petala	dp_petala	q1_petala	q2_petala	q3_petala	cv_petala
Espécies principais						
setosa	1.462	0.1736640	1.4	1.50	1.575	11.878522
virginica	5.552	0.5518947	5.1	5.55	5.875	9.940466
versicolor	4.260	0.4699110	4.0	4.35	4.600	11.030774

Fonte: Elboração própria.

Rótulo (legenda) para grupo de linhas
Exercício

Inclua um *rótulo* para as linhas `pardas` e `pretas` com o texto "negras" no objeto `gt_tab`.

Rótulo (legenda) para grupo de colunas

`tab_spanner`: permite *rótulo* para grupo de colunas.

```
gt_tab <- gt_tab |>
  tab_spanner(
    columns = c(
      q1_petala,
      q2_petala,
      q3_petala
    ),
    label = "Quantis"
  ) |>
  tab_spanner(
    columns = c(dp_petala, cv_petala),
    label = "Dispersão"
  )
gt_tab
```

Comprimento de pétala

Algumas estatísticas descritivas

		Dispersão		Quantis		
especies	m_petala	dp_petala	cv_petala	q1_petala	q2_petala	q3_petala
Espécies principais						
setosa	1.462	0.1736640	11.878522	1.4	1.50	1.575
virginica	5.552	0.5518947	9.940466	5.1	5.55	5.875
versicolor	4.260	0.4699110	11.030774	4.0	4.35	4.600

Fonte: Elboração própria.

Rótulo (legenda) para grupo de colunas
Exercício

Inclua um *rótulo* pra as colunas do primeiro quartil, segundo quartil e terceiro quartil com o texto "Quartis" no objeto `gt_tab`.

Movendo as colunas na tabela

- `cols_move_to_start`: move uma ou mais colunas para o início da tabela.
- `cols_move_to_end`: move uma ou mais colunas para o fim da tabela.
- `cols_move`: move uma ou mais colunas para depois um determinada coluna.

```
gt_tab <- gt_tab |>
  cols_move_to_start(
    columns = c(especies, dp_petala, cv_petala)
  ) |>
  cols_move_to_end(
    columns = m_petala
  ) |>
  cols_move(
    after = cv_petala,
    columns = c(q1_petala, q2_petala, q3_petala)
  )
gt_tab
```

Comprimento de pétala

Algumas estatísticas descritivas

	Dispersão		Quantis			
especies	dp_petala	cv_petala	q1_petala	q2_petala	q3_petala	m_petala
Espécies principais						
setosa	0.1736640	11.878522	1.4	1.50	1.575	1.462
virginica	0.5518947	9.940466	5.1	5.55	5.875	5.552
versicolor	0.4699110	11.030774	4.0	4.35	4.600	4.260

Fonte: Elboração própria.

Movendo as colunas na tabela
Exercício

Deixe as colunas de `gt_tab` na seguinte ordem: *raça*, *média*, *primeiro quartil*, *segundo quartil*, *terceiro quartil* e *desvio padrão* usando as funções `cols_move_to_start`, `cols_move` e `cols_move_to_end`.

Atualizando as colunas

`cols_label`: permite atualizar os *rótulos* das colunas.

```
gt_tab <- gt_tab |>
  cols_label(
    especies = md("**Espécies**"),
    dp_petala = "Desvio padrão",
    cv_petala = "Coeficiente de variação",
    q1_petala = md("*Q1*"),
    q2_petala = md("*Q2*"),
    q3_petala = md("*Q3*"),
    m_petala = "Média"
  )
gt_tab
```

Comprimento de pétala

Algumas estatísticas descritivas

	Dispersão		Quantis			
Espécies	Desvio padrão	CV	*Q1*	*Q2*	*Q3*	Média
Espécies principais						
setosa	0.1736640	11.878522	1.4	1.50	1.575	1.462
virginica	0.5518947	9.940466	5.1	5.55	5.875	5.552
versicolor	0.4699110	11.030774	4.0	4.35	4.600	4.260

Fonte: Elboração própria.

Atualizando as colunas
Exercício

Para o objeto `gt_tab`, garante que as colunas tenham os seguintes nomes: *Raça*, *Média*, *Desvio padrão*, *Primeiro quartil*, *Segundo quartil* e *Terceiro quartil*.

Formatação de valores

`fmt_number`: formatação de valores numéricos de uma ou mais colunas.

```
gt_tab <- gt_tab |>
  fmt_number(
    columns = c(
      dp_petala, q1_petala, q2_petala,
      q3_petala, m_petala
    ),
    decimals = 2,
    dec_mark = ",",
    sep_mark = "."
  ) |>
  fmt_number(
    columns = cv_petala,
    decimals = 2,
    dec_mark = ",",
    sep_mark = ".",
    patter = "{x} \\%"
  )
gt_tab
```

Comprimento de pétala

Algumas estatísticas descritivas

	Dispersão		Quantis			
Espécies	Desvio padrão	CV	*Q1*	*Q2*	*Q3*	Média
Espécies principais						
setosa	0,17	11,88 %	1,40	1,50	1,58	1,46
virginica	0,55	9,94 %	5,10	5,55	5,88	5,55
versicolor	0,47	11,03 %	4,00	4,35	4,60	4,26

Fonte: Elboração própria.

Formatação de valores
Exercício

No objeto `gt_tab`, para as colunas numéricas coloque “,” para o separador de casa decimal e “.” para o agrupador de milhar.