

R para Ciência de Dados: Exploração e Visualização de Dados

Departamento de Estatística
Instituto de Matemática e Estatística
Universidade Federal da Bahia

Preparando o ambiente

Durante o curso

- Usaremos nas aulas: posit.cloud.
 - Recomendamos instalar e usar R com versão pelo menos 4.1: cran.r-project.org.
 - usaremos o *framework* [tidyverse](https://www.tidyverse.org):
 - Instalação: `install.packages("tidyverse")`
-

Na sua casa

- **IDE** recomendadas: [RStudio](https://rstudio.com) e [VSCode](https://code.visualstudio.com).
 - Caso você queira usar o [VSCode](https://code.visualstudio.com), instale a extensão da linguagem R: [REditorSupport](https://marketplace.visualstudio.com/items?itemName=RobertoHernandez.REditorSupport).
- Outras linguagens interessantes: [python](https://python.org) e [julia](https://julialang.org).
 - [python](https://python.org): linguagem interpretada de propósito geral, contemporânea do R, simples e fácil de aprender.
 - [julia](https://julialang.org): linguagem interpretada para análise de dados, lançada em 2012, promete simplicidade e velocidade.

A linguagem R

A precursora da linguagem 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 da linguagem 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.

Motivos para 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

- **Nível *cheguei agora aqui*:** [zen do R](#).
- **Nível Iniciante:** [R Tutorial na W3Schools](#).
- **Nível Iniciante:** [Hands-On Programming with R](#).
- **Nível Intermediário:** [R for Data Science](#).
- **Nível Avançado:** [Advanced R](#).

Em pt-br

- **Curso R:** material.curso-r.com.
- **ecoR:** ecor.ib.usp.br.

O que você pode fazer quando estiver em apuros?

- consultar a documentação do R:

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

- Peça ajuda a um programador mais experiente.
- Consulte o pt.stackoverflow.com.
- Use ferramentas de busca como o [google](https://www.google.com) e duckduckgo.com.

```
log("G")
```

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

Operações básicas

Soma

```
1 + 1
```

```
## [1] 2
```

Subtração

```
2 - 1
```

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

Multiplicação

```
3 * 3
```

```
## [1] 9
```

Divisão

```
3 / 2
```

```
## [1] 1.5
```

Potenciação

```
2^3
```

```
## [1] 8
```

Operações básicas

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 $5^{1.23}$

Pacotes na linguagem R

- *códigos* criados pela comunidade
- disponibilizados principalmente na plataforma cran.r-project.org

instalação:

```
install.packages(pacman)
```

carregando pacotes

Pacotes precisam estar instalados

```
library(pacman)
```

Carregando pacotes com pacman

- Se os pacotes estão instalados: `pacman` carrega os pacotes
- Se os pacotes **não** estão instalados: `pacman` instala e depois carrega os pacotes
- Omite mensagens dos pacotes
- Pacote separados por vírgula

```
p_load(glue, readxl, writexl, janitor, gt, ggthemes, readODS,  
       patchwork, tidyverse)
```

Funções na linguagem R

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

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

nome da função *parênteses* *argumentos posicionais* *argumentos nomeados* *parênteses*
`nome_funcao` `(` `valor1,` `valor2,` `nome1 = valor3,` `nome2 = valor4` `)`

example:

```
read_xlsx('dados/brutos/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 ENEM 2022 (cada pessoa tem sua cidade) usando a função `read_xlsx` do pacote `readxl`.

Os dados no R

- **Tipo de dados:** `character` (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`.
 - `array` é uma estrutura de dados multidimensional para armazenar sequências de matrizes (ou sequência de *arrays*). Para detalhes sobre `array`, consulte [Arrays in R](#).
- **Estrutura de dados Heterôgenea:** `list` e `data.frame` (`tibble` no `tidyverse`).

Tipo de dados no R

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"
```

Estrutura de dados homogênea

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)`.

Vetor de caracteres

```
vetor_nomes <- c("Gilberto", "Sassi")  
vetor_nomes
```

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

```
vetor_texto_vazio <- vector("character", 3)  
vetor_texto_vazio
```

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

Vetores

Vetor de números reais

```
vetor_numerico <- c(0.2, 1.35)  
vetor_numerico
```

```
## [1] 0.20 1.35
```

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

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

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

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

Vetor de números inteiros

```
vetor_inteiros <- c(1L, 2L)  
vetor_inteiros
```

```
## [1] 1 2
```

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

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

Vetor lógico

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

```
## [1] TRUE FALSE
```

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

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

Estrutura de dados homogênea

Matriz

- Agrupamento de valores de mesmo tipo em um único objeto de dimensão 2.
- Criação de matriz:
 - `matrix(..., nrow = <integer>, ncol = <integer>)`
 - `cbind` e `rbind`
 - `diag(<vector>)`

Matriz de caracteres

```
matriz_texto <- matrix(c("a", "b", "c", "d"), nrow = 2)
matriz_texto
```

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

Matriz

Matriz de números reais

```
matriz_num_real <- diag(c(1.1, 2.3, 3.3))  
matriz_num_real
```

```
##           [,1] [,2] [,3]  
## [1,]  1.1  0.0  0.0  
## [2,]  0.0  2.3  0.0  
## [3,]  0.0  0.0  3.3
```

Matriz

Matriz de inteiros

```
matriz_inteiros <- cbind(c(1L, 2L), c(3L, 4L), nrow = 2)
matriz_inteiros
```

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

Matriz de valores lógicos

```
matriz_logica <- rbind(
  c(TRUE, FALSE),
  c(TRUE, TRUE)
)
matriz_logica
```

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

Matriz

Exercício

Crie as seguintes matrizes e vetores:

- 1 $\begin{pmatrix} \text{João} & \text{Joana} \\ \text{Josué} & \text{Joaquina} \end{pmatrix}$
- 2 $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 3.1 \end{pmatrix}$
- 3 $(\text{TRUE} \quad \text{TRUE} \quad \text{FALSE})$
- 4 $(0,1 \quad 0,2 \quad 0,3 \quad 0,4 \quad 0,5)$

Operações com vetores

Operações com vetores numéricos (double, integer e complex).

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

Slicing

Selecionando todos os elementos entre o primeiro e o quinto.

```
letras <- c("a", "b", "c", "d", "e", "f", "g", "h", "i")  
letras[1:5]
```

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

Operações com vetores

Adição (vetores numéricos)

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

```
## [1] 6 7 8 9 10
```

Subtração (vetores numéricos)

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

```
## [1] -5 -5 -5 -5 -5
```

Operadores com vetores

Multiplicação (vetores numéricos)

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

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

Divisão (vetores numéricos)

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

```
## [1] 0.1666667 0.2857143 0.3750000 0.4444444 0.5000000
```

Operações com vetores

Exercício

Realize as seguintes operações envolvendo vetores:

- ① $(1 \ 2 \ 3) + (0,1 \ 0,05 \ 0,33)$
- ② $(1 \ 2 \ 3) - (0,1 \ 0,05 \ 0,33)$
- ③ $(1 \ 2 \ 3) * (0,1 \ 0,05 \ 0,33)$
- ④ $(1 \ 2 \ 3) / (0,1 \ 0,05 \ 0,33)$

Operações com matrizes

Operações com matrizes numéricas (double, integer e complex).

- Operações básicas: adição, subtração, multiplicação e divisão (realizadas em cada elemento das matrizes).
- Outras operações elementares:
 - Multiplicação de matrizes (vide [multiplicação de matrizes](#)): `A %*% B`
 - Inversão de matrizes (vide [inversão de matrizes](#)): `solve(A)`
 - Matriz transposta (vide [matriz transposta](#)): `t(A)`
 - Determinante (vide [determinante de uma matriz](#)): `det(A)`
 - Solução de sistema de equações lineares (vide [sistema de equações lineares](#)): `solve(A, b)`

Operadores com matrizes

Exemplo

Cada pessoa de uma equipe de quatro pessoas divide a administração de duas regiões (A e B) conforme as matrizes abaixo:

Postos de venda da região A

```
postos_regiao_a <- rbind(  
  c(2, 4), # primeira linha  
  c(1, 5) # segunda linha  
)
```

Postos de venda da região B

```
postos_regiao_b <- cbind(  
  c(23, 44), # primeira coluna  
  c(19, 12) # segunda coluna  
)
```

Operações com matrizes

Soma de duas matrizes

Número total de postos de venda administrados por cada pessoa.

```
postos <- postos_regiao_a + postos_regiao_b  
postos
```

```
##           [,1] [,2]  
## [1,]      25  23  
## [2,]      45  17
```

Operações com matrizes

Subtração de matrizes

Número de postos de venda da região A, conhecendo o total de número de postos e o número de postos de venda região B.

```
# postos de venda da região A
postos_regiao_a <- postos - postos_regiao_b
postos_regiao_a
```

```
##           [,1] [,2]
## [1,]         2    4
## [2,]         1    5
```

Operações com matrizes

Suponha que cada pessoa consiga vender um determinado número de produtos por posto de venda conforme a matriz abaixo.

```
produtos_pessoa <- rbind(  
  c(4, 2),  
  c(1, 3)  
)
```

A quantidade de produtos vendidos por pessoa na região A é dado por:

```
vendas_regiao_a <- postos_regiao_a * produtos_pessoa  
vendas_regiao_a
```

```
##      [,1] [,2]  
## [1,]    8    8  
## [2,]    1   15
```

Operações com matrizes

Imagine que temos o número de produtos vendidos por pessoa na região A e sabemos o número de postos de cada pessoa na equipe na região A. Então, podemos descobrir o número de produtos vendidos por um posto para cada pessoa.

```
vendas_regiao_a / postos_regiao_a
```

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

Outras operações importantes com matrizes

Código em R	Descrição da operação
<code>A %o% B</code>	produto diádico $A \cdot B^T$
<code>crossprod(A, B)</code>	$A \cdot B^T$
<code>crossprod(A)</code>	$A \cdot A^T$
<code>diag(x)</code>	retorna uma matrix diagonal com diagonal igual a x
<code>diag(A)</code>	retorna um vetor com a diagona de A
<code>diag(k)</code>	retorna uma matriz diagona de ordem k

Operações com matrizes

Exercício

Realizei as seguinte operações envolvendo as matrizes:

$$① \begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} + \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$$

$$② \begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} - \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$$

$$③ \begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} \cdot \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$$

$$④ \begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} / \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix}$$

Estrutura de dados heterogênea

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 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:
 - `escolaridade`: escolaridade
 - `profissao`: variável qualitativa com os valores possíveis:
funcionário público, funcionário da iniciativa privada,
estudante e desempregado
- `matriz`: inclua uma matriz de números reais de dimensão 2×2

Estrutura de dados heterogênea data frame (tibble)

- Agrupamento de dados em tabela, onde: cada coluna é uma variável; cada linha é uma observação. Usamos a tabela tidy:
 - Cada variável em uma única coluna
 - Cada unidade observacional em uma única linha
- Criação de tibble: `tibble(...)` e `tribble(...)`.

tibble (data frame)

```
df <- tibble(  
  nome = c("João", "Josué", "Joaquim", "José"),  
  idade = c(20, 21, 23, 32)  
)  
glimpse(df)
```

```
## Rows: 4  
## Columns: 2  
## $ nome <chr> "João", "Josué", "Joaquim", "José"  
## $ idade <dbl> 20, 21, 23, 32
```

Operações em data frames

Operações em um tibble

Algumas funções úteis depois de aprender a carregar os dados no R.

Código em R	Descrição
<code>head()</code>	Mostra as primeiras linhas de um tibble
<code>tail()</code>	Mostra as últimas linhas de um tibble
<code>glimpse()</code>	Impressão de informações básicas dos dados
<code>add_case()</code> ou <code>add_row()</code>	Adiciona uma nova observação

```
head(df, n = 2)
```

```
## # A tibble: 2 x 2
##   nome   idade
##   <chr> <dbl>
## 1 João    20
## 2 Josué   21
```

```
df <- add_case(df, nome = "Josefina", idade = 31)
```

```
tail(df, n = 2)
```

```
## # A tibble: 2 x 2
##   nome      idade
##   <chr>    <dbl>
## 1 José      32
## 2 Josefina  31
```

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*
- adicione *Fulano* de *30 anos* ao tibble *df* para obter o seguinte *dataset*:

nome	idade
João	20
Josué	21
Joaquim	23
José	32
Josefina	31
Fulano	30

Operações com listas

Concatenação de listas

```
lista_1  <- list("a", "b")  
lista_2 <- list(1, 2)  
lista_final <- c(lista_1, lista_2)  
lista_final
```

```
##  [[1]]  
##  [1] "a"  
##  
##  [[2]]  
##  [1] "b"  
##  
##  [[3]]  
##  [1] 1  
##  
##  [[4]]  
##  [1] 2
```

Operações com listas

Slicing a lista

```
lista_final[1:2]
```

```
## [[1]]  
## [1] "a"  
##  
## [[2]]  
## [1] "b"
```

Acessando elemento pela ordem

```
# Acessando o segundo elemento da lista d.  
lista_final[[2]]
```

```
## [1] "b"
```

Acessando elemento usando \$

```
lista <- list(elemento_1 = 1, elemento_2 = "docente")  
lista$elemento_2
```

```
## [1] "docente"
```

Operações com listas

Slicing uma lista com ["nome"]

```
lista <- list(elemento_1 = 1, elemento_2 = "docente",  
             elemento_3 = list("olá"))  
lista["elemento_3"] # funciona como slicing
```

```
## $elemento_3  
## $elemento_3[[1]]  
## [1] "olá"
```

Obtendo os nomes dos elementos em um lista

```
lista <- list(c(1, 2, 3), elemento_1 = 1, elemento_2 = "docente",  
            elemento_3 = list("olá"))  
names(lista)
```

```
## [1] "" "elemento_1" "elemento_2" "elemento_3"
```

Listas

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`

Valores especiais em R

Valor	Descrição	O que é	Função para identificar
NA	Not Available	Valor faltante.	<code>is.na()</code>
NaN	Not a Number	Resultado do cálculo indefinido.	<code>is.nan()</code>
Inf	Infinito	Valor que excede o valor máximo que sua máquina aguenta.	<code>is.inf()</code>
NULL	Nulo	Valor indefinido de expressões e funções (diferente de NaN e NA)	<code>is.null()</code>

Parênteses 1: guia de estilo no R

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 (*built-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: `^`.

Para mais detalhes, consulte: [guia de estilo do tidyverse](#).

Parênteses 2: 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](#).

Carregando dados no R

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

- **Pacote:** `readxl` do `tidyverse` (instale com o comando `install.packages('readxl')`)
- Parâmetros das funções `read_xls` (para ler arquivos `.xls`) e `read_xlsx` (para ler 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 oficial do *tidyverse*: [documentação de `read_xl`](#).

Carregando dados no R

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

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

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,
## $ 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",
```

Carregando dados no R

As formatações dos arquivos csv

csv: *comma separated values* (valores separados por coluna).

O *separador* (caracter usado para separar colunas) 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!

Carregando dados no R

Leitura de arquivos no formato csv

- **Pacote:** readr do tidyverse (instale com o comando `install.packages('readr')`).
- Parâmetros 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")
glimpse(dados_mtcarros)
```

```
## Rows: 32
## Columns: 11
## $ milhas_por_galao <dbl> 21.0, 21.0, 22.8, 21.4, 18.7, 18.1,
## $ cilindros         <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4, 6, 6, 8,
## $ cilindrada        <dbl> 160.0, 160.0, 108.0, 258.0, 360.0,
## $ cavalos_forca     <dbl> 110, 110, 93, 110, 175, 105, 245, 6
## $ eixo              <dbl> 3.90, 3.90, 3.85, 3.08, 3.15, 2.76,
## $ peso              <dbl> 2.620, 2.875, 2.320, 3.215, 3.440,
## $ velocidade        <dbl> 16.46, 17.02, 18.61, 19.44, 17.02,
## $ forma             <dbl> 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0,
## $ transmissao       <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ marchas           <dbl> 4, 4, 4, 3, 3, 3, 3, 4, 4, 4, 4, 3,
## $ carburadores       <dbl> 4, 4, 1, 1, 2, 1, 4, 2, 2, 4, 4, 3,
```

Carregando dados no R

Leitura de arquivos no formato ods

- **Pacote:** readODS (instale com o comando `install.packages('readODS')`).
- Parâmetros 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](#).

Carregando dados no R

Leitura de arquivos no formato ods

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

glimpse(dados_dentes)
```

```
## Rows: 60
## Columns: 3
## $ comprimento <dbl> 4.2, 11.5, 7.3, 5.8, 6.4, 10.0, 11.2, 11
## $ suplemento <chr> "Vitamina C", "Vitamina C", "Vitamina C"
## $ dose <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
```

Carregando dados no R

Exercício

- Leia os dados do ENEM de sua cidade no formato `.xlsx` (coloque o arquivo na pasta `dados/brutos`).
- Leia os dados do ENEM de sua cidade no formato `.csv` (coloque o arquivo na pasta `dados/brutos`).
- Leia o arquivo `crescimento_dentes.ods` (o arquivo já está na pasta `dados/brutos`).

Salvando dados no R

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

|>

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)
```

```
log(sqrt(sum(x2)))
```

é exatamente a mesma coisa que executar

```
x2 |> 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(  
  acrescete(  
    "tigela vazia",  
    "farinha"  
  ),  
  "ovos"  
)
```

- Passo3:

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

- Passo4:

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

- Passo 5:

```
misture(  
  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 no R

Gráficos e Tabelas

Alguns conceitos básicos

- **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 entrevistadas pelo datafolha.

Classificação de variáveis

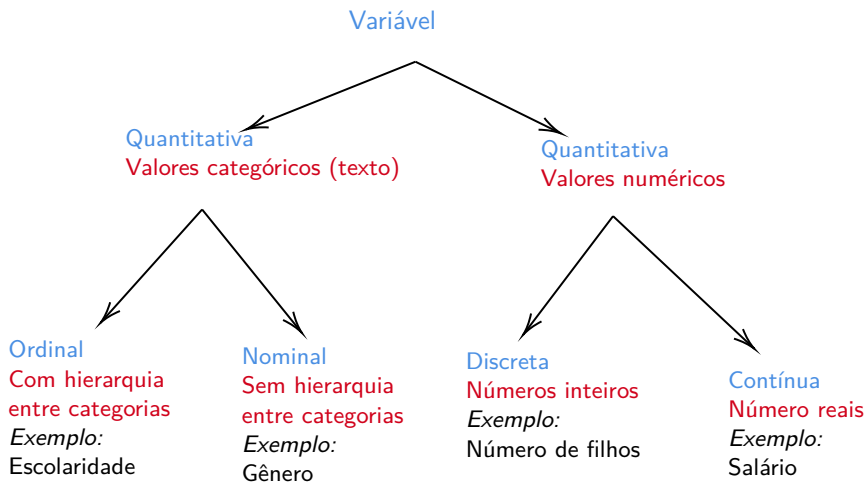


Figura 1: Classificação de variáveis.

Tabela de frequências

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 frequências

Variável qualitativa

- **Pacote:** `tabyl`, `adorn_totals` e `adorn_pct_formatting` do `janitor` (instale com o comando `install.packages('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 frequências

Variável qualitativa

```
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 frequências

Variável qualitativa

Exercício

Para o conjunto de dados ENEM 2022 (cada pessoa tem sua cidade) , construa a tabela 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 frequências

Variável quantitativa discreta

Muito semelhante a tabela de frequências 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 $\{x_1, \dots, x_k\}$ são os números que são valores únicos de X na amostra.

Tabela de frequências

Variável quantitativa discreta

```
tab <- tabyl(dados_mtcarrros, 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 frequências

Variável quantitativa discreta

Exercício

Para os dados do ENEM 2022 (cada pessoa tem sua cidade), construa a tabela de frequências para a variável q005: número de pessoas que moram na casa da(o) candidata(o).

Tabela de frequências

Variável quantitativa contínua

X: variável quantitativa contínua

Tabela 6: Tabela de frequências para a variável quantitativa contínua.

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

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

Recomendações:

- use l_0, l_1, \dots, 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 frequências

Variável quantitativa contínua

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

```
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
  )
)
```

Tabela de frequências

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 do ENEM (cada pessoa tem sua cidade), construa as seguintes tabelas de frequências:

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

Gráficos no R

- **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](#)

Gráficos no R

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](#).

Gráficos no R

Gráfico de barras no ggplot2

- **função:** `geom_bar()`. Para porcentagem: `geom_bar(x = <variável no eixo x>, y = ..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()
```

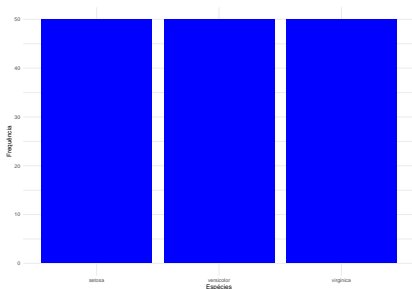


Gráfico de barras

Variável qualitativa

Exercício

Para o conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), 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 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 frequências

Variável quantitativa discreta

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

```
tab <- tabyl(dados_mtcarrros, 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_mtcarrros) +  
  geom_bar(  
    mapping = aes(carburadores, after_stat(100 * prop)),  
    fill = "#002f81"  
  ) +  
  labs(x = "Número de carburadores", y = "Porcentagem") +  
  theme_minimal()
```

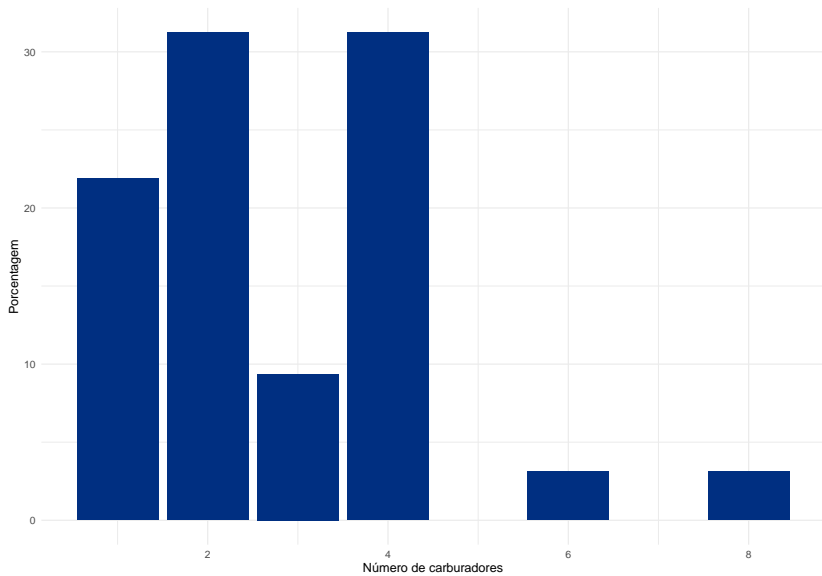


Gráfico de barras

Variável quantitativa discreta

Exercício

- Para a variável q005 do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), construa o gráfico de barras onde o eixo y é a frequência absoluta.
- Para a variável q005 do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), construa o gráfico de barras onde o eixo y é a frequência relativa.
- Para a variável q005 do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), construa o gráfico de barras onde o eixo y é a porcentagem.

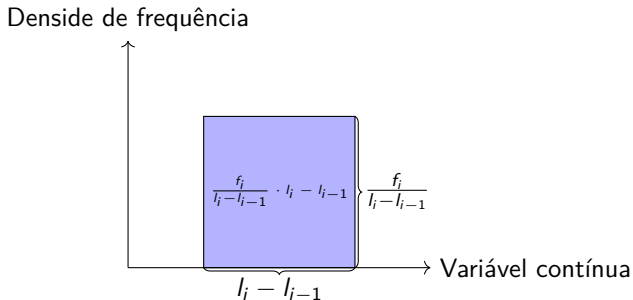
Histograma

Para variáveis 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 $[l_{i-1}, l_i)$, $i = 1, \dots, n$, será representada por um barra com área f_i , $i = 1, \dots, n$.
- 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

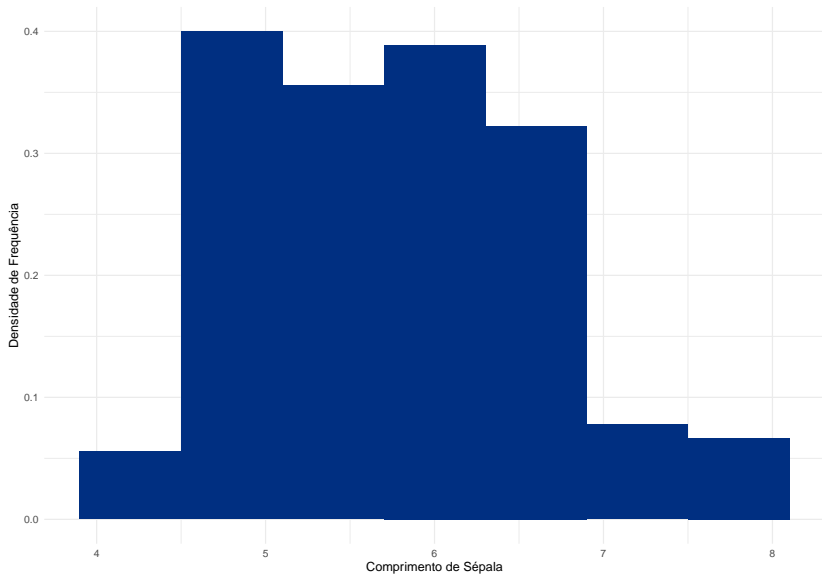
Histograma

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



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 ENEM 2022 (cada pessoa tem sua cidade), construa o histograma onde os intervalos tem o mesmo tamanho igual a 100.
- Para a variável `nu_nota_cn` do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), 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:** $\bar{x} = \frac{x_1 + \dots + x_n}{n}$.
 - **Mediana:** valor que divide a sequência ordenada de valores em duas partes iguais.
-

Medidas de dispersão

A ideia é medir a homogeneidade dos valores.

- **Variância:** $s^2 = \frac{(x_1 - \bar{X})^2 + \dots + (x_n - \bar{X})^2}{n - 1}$.
- **Desvio padrão:** $s = \sqrt{s^2}$ (mesma unidade dos dados).
- **Coefficiente de variação** $cv = \frac{s}{\bar{x}} \cdot 100\%$ (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 ENEM 2022 (cada pessoa tem sua cidade) por gênero (`tp_sexo`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_cn` do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade) por gênero (`tp_sexo`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_mt` do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade) 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 ENEM 2022 (cada pessoa tem sua cidade) por raça (`tp_cor_raca`).

Ideia

$q(p)$ é um valor que satisfaz;

- $100 \cdot p\%$ das observações é no máximo $q(p)$
- $100 \cdot (1 - p)\%$ das observações é no mínimo $q(1 - p)$

Alguns quantis especiais

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

Quantis

```
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 ENEM 2022 (cada pessoa tem sua cidade) 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 ENEM 2022 (cada pessoa tem sua cidade) 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 ENEM 2022 (cada pessoa tem sua cidade) 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 ENEM 2022 (cada pessoa tem sua cidade) 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: $dq = q_3 - q_1$

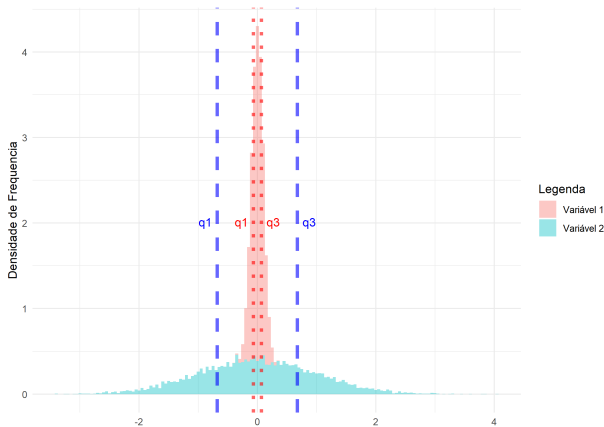


Diagrama de caixa (ou *boxplot*)

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

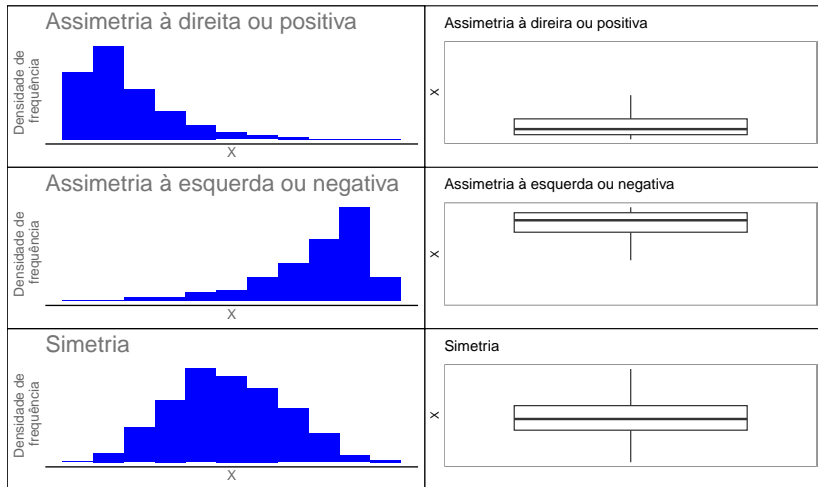


Diagrama de caixa (ou *boxplot*)

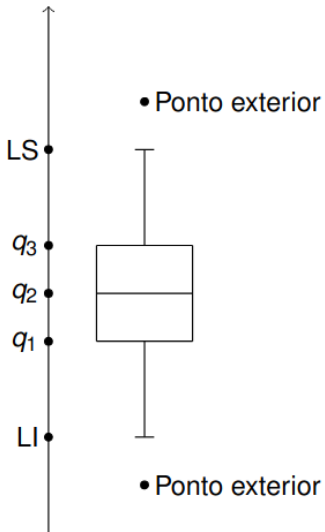


Diagrama de caixa (ou *boxplot*)

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

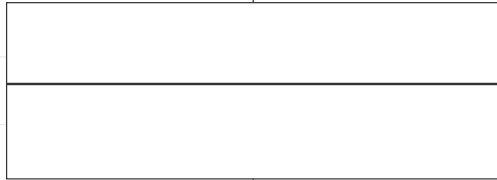
Comprimento de Sépala

8

7

6

5



Gráficos lado a lado com patchwork

- patchwork permite que colocar gráficos lado a lado com
 - +: figuras ao lado
 - \: figuras embaixo
- Para mais detalhes, 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
```

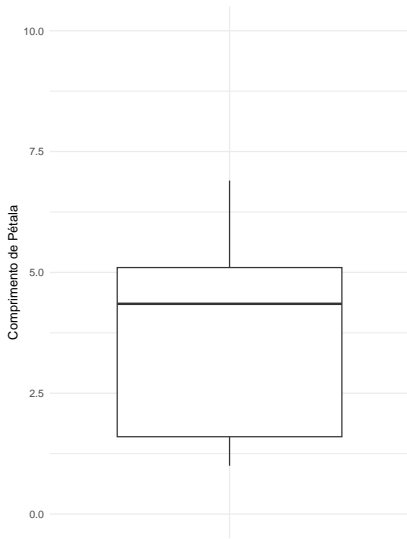
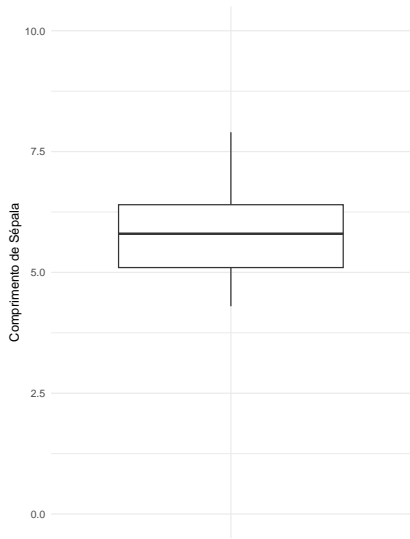


Diagrama de caixa

Exercício

Para o conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), 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`.

Gráficos

Duas variáveis

Ideia: estudar a associação entre duas variáveis quantitativas.

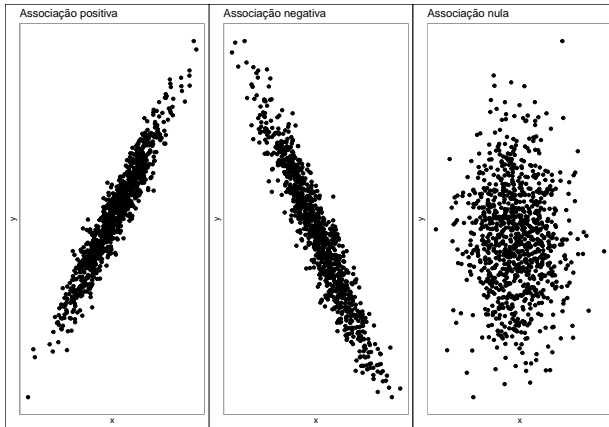


Gráfico de dispersão

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

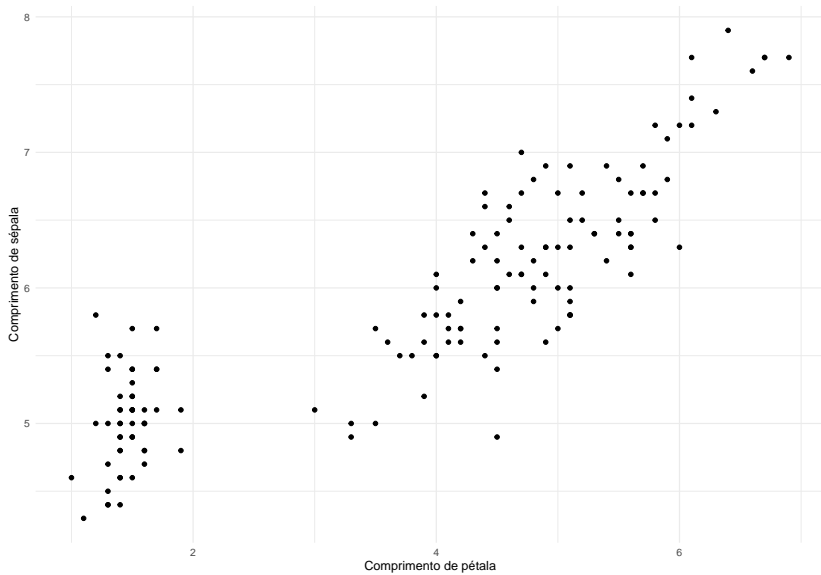


Gráfico de dispersão

Exercício

Para o conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), 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:

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

Desejamos estudar a associação entre X e Y .

Associação entre X e Y

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

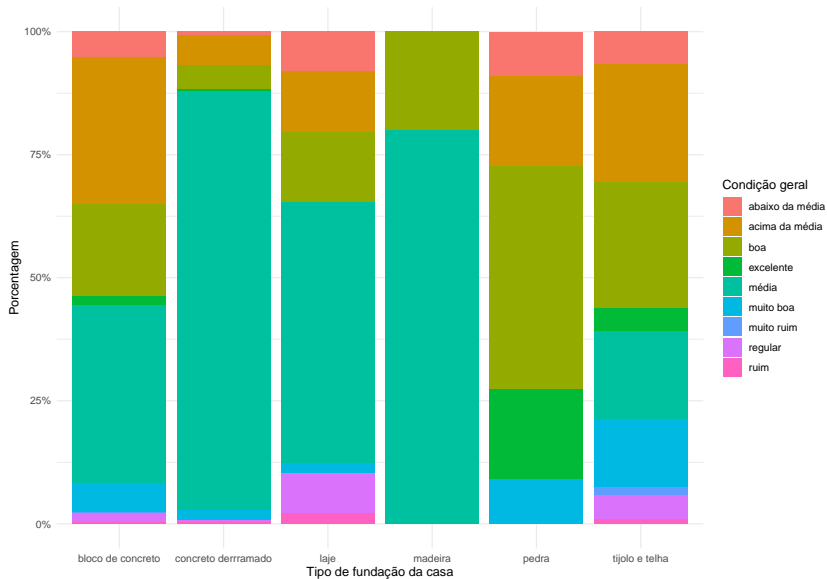
- **não associados:** se ao conhecermos o valor de Y para um elemento da população, **continuamos** com a percentagem $100 \cdot f_i \%$ deste elemento ter valor de X igual a A_i
- **associados:** se ao conhecermos o valor de Y para um elemento da população, **alteramos** a percentagem $100 \cdot f_i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

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()
```

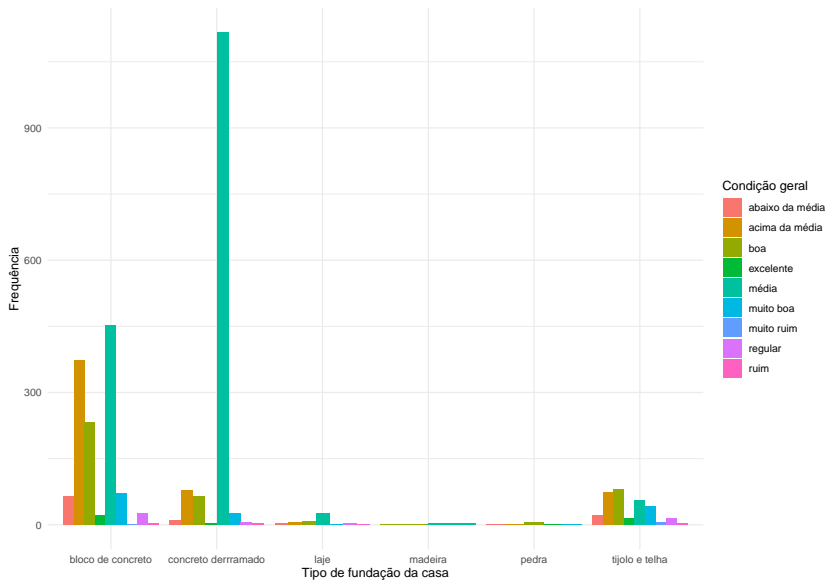


Associação entre duas variáveis qualitativas

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 = "Frequência",
       fill = "Condição geral") +
  theme_minimal()
```



Associação entre duas variáveis qualitativas

Gráfico de barras

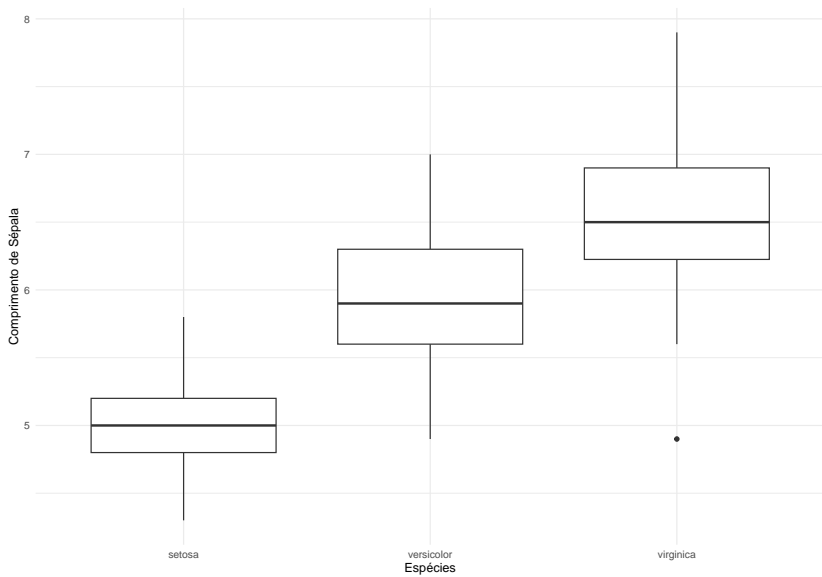
Exercício

- Verifique se existe associação entre as variáveis `q006` e `tp_cor_raca` do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade) usando gráfico de barras usando o `position=fill`.
- Verifique se existe associação entre as variáveis `q006` e `tp_sexo` do conjunto de dados ENEM 2022 (cada pessoa tem sua cidade) usando gráfico de barras 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()
```



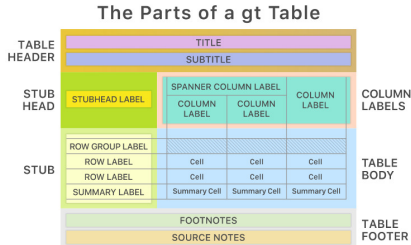
Comparação de medianas usando Diagrama de caixa Exercício

- Para o conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), compare a variável `nu_nota_mt` por raça (`tp_cor_raca`).
- Para o conjunto de dados ENEM 2022 (cada pessoa tem sua cidade), compare a variável `nu_nota_cn` por raça (`tp_cor_raca`).
- Coloque os dois gráficos acima lado a lado usando o pacote `patchwork`.

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
```

Salvando tabelas com o pacote gt

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 gtnos 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 ENEM 2022 (cada pessoa tem sua cidade) 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:** Elaboração 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: Elaboração 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
<i>Espécies principais</i>						
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: Elaboração 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

especies	m_petala	Dispersão		Quantis		
		dp_petala	cv_petala	q1_petala	q2_petala	q3_petala
<i>Espécies principais</i>						
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: Elaboração 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

especies	Dispersão		Quantis			m_petala
	dp_petala	cv_petala	q1_petala	q2_petala	q3_petala	
<i>Espécies principais</i>						
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: Elaboração 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

Espécies	Dispersão		Quantis			Média
	Desvio padrão	CV	Q1	Q2	Q3	
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: Elaboração própria.

Atualizando as colunas

Exercício

Para o objeto `gt_tab`, garanta 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

Espécies	Dispersão		Quantis			Média
	Desvio padrão	CV	Q1	Q2	Q3	
<i>Espécies principais</i>						
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: Elaboração 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.