

# Ciência de Dados para Cultura usando R | Exploração e Visualização

Observatório da Economia Criativa  
OBEC

Professor: Gilberto Sassi

Planilhas

# Planilhas eletrônicas

- Inspiração em planilha de papel da contabilidade;
- Surgiram na década de 1970 com LANPAR, VisiCalc e Lotus-1-2-3;
- Informações são distribuídas por linhas e colunas;
- Linhas são identificadas com números inteiros;
- Colunas são identificadas com letras;
- Cada linha um transação;
- Cada coluna um tipo de informação diferente.



Figura 1: Planilha de papel usada antes do surgimento das planilhas eletrônicas na década de 1970.

## Planilhas Google

- Planilha eletrônica baseada a Web criada em 2006 pela 2Web Technologies;
- Vinculada a uma conta Google;
- São permitidas diversas abas;
- É permitido inputar valores diretamente nas células;
- É permitido validação de valores;
- Importação de dados.

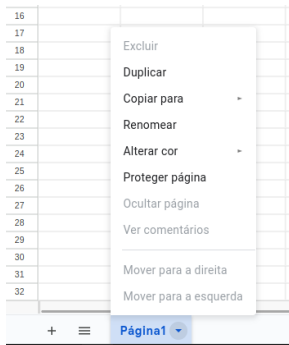


Figura 2: Abas no Planilhas Google.

# Planilhas Google

## Exercício

### Exercício 1

- a. Crie uma nova planilha em uma pasta de um projeto ou pesquisa (crie este diretório);
  - b. Crie duas abas;
  - c. Renomeie as duas abas;
  - d. Anote na primeira aba as informações sobre o seu núcleo familiar: nome, data de nascimento e idade;
  - e. Anote na segunda aba as informações sobre seus melhores amigos: nome, profissão e idade.
- 

### Exercício 2

- a. Importe a planilha de dados do ENEM 2023 (cada pessoa tem sua cidade).

# Planilhas Google

Algumas formatações possíveis:

- Cor, Negrito, Itálico, Sublinhado e Bordas;
- Alinhamento e Mesclagem;
- Formatação condicional;
- Formatação de números.



Figura 3: Formatação de texto e número

# Planilhas Google

## Fórmulas

- Operações básicas
- Filtro
- Fixação de colunas e/ou linhas com \$
- Fórmulas:
  - média
  - soma
  - cont.se (tabela de contingência e tabela de distribuição de frequências)
  - soma.se
  - máximo
  - mínimo



Figura 4: Inclusão de fórmula

# Planilhas Google

## Exercício

### Exercício

- Calcule a média de notas em matemática (`nu_nota_mt`) por raça;
- Conte a quantidade por categoria da variável `tp_cor_raca` incluindo frequência e porcentagem;
- Calcule a nota média de `nu_nota_lc` por renda familiar (`q006`);
- Calcule a nota para ingresso na Universidade Federal de Pindorama que usa a nota calculada pela seguinte fórmula:

$$nota\_final = \frac{3 \cdot nu\_nota\_mt + 3 \cdot nu\_nota\_lc + 2 \cdot nu\_nota\_cn + 2 \cdot nu\_nota\_ch}{10}.$$

# Planilhas Google

## Gráficos

- Inclusão de gráficos de barra
- Inclusão de rótulos nas barras
- Cores HTML ([imagecolorpicker.com](http://imagecolorpicker.com))



Figura 5: Gráfico de barras.

# Planilhas Google

## Exercício

### Exercício

- Construa um gráfico de barras para tp\_cor\_raca;
- Construa um gráfico de barras para tp\_faixa\_etaria;
- Construa um gráfico de barras para q006;
- Construa um gráfico de barras para q002.

# Planilhas Google

## **Tabela dinâmica**

- Tabela de distribuição de frequência
- Média e mediana

Clique em [Inserir > Tabela dinâmica](#).

# Planilhas Google

## Exercício

### Exercício

- Calcule a média de notas em matemática (`nu_nota_mt`) por raça;
- Conte a quantidade por categoria da variável `tp_cor_raca` incluindo frequência e porcentagem;
- Calcule a nota média de `nu_nota_1c` por renda familiar (`q006`);
- Calcule a nota média de `nu_nota_mt` por renda familiar (`q006`) para as categorias `parda` e `preta` da variável `tp_cor_raca`.

A linguagem  $\mathbb{R}$

# Preparando o ambiente

## Durante o curso

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

## Na sua casa

- **IDE** recomendadas: [RStudio](#) e [VSCode](#).
  - Caso você queira usar o [VSCode](#), instale a extensão da linguagem R: [REditorSupport](#).
- Outras linguagens interessantes para Ciência de Dados: [python](#) e [julia](#).
  - [python](#): linguagem interpretada de propósito 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

## 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](http://material.curso-r.com).
- **ecoR:** [ecor.ib.usp.br](http://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](http://pt.stackoverflow.com).
- Use ferramentas de busca como o [google](http://google.com) e [duckduckgo.com](http://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](https://cran.r-project.org)
- 

## **instalação:**

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

---

## **carregando pacotes**

Pacotes precisam estar instalados

```
p_load(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 2023 (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`.

## 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
```

# Fator

Podemos fixar o conjunto de valores possíveis de uma variável qualitativa (e especificar uma ordem implícita) usando `factor`.

Principais vantagens:

- Evita os erros de digitação.
- Introduce uma ordenação que pode ser útil para construir gráficos e tabelas.
- Necessário para funções de modelagem estatística (que não veremos neste curso).

Vamos usar o pacote `forcats`.

## Fator

Função `fct` do pacote `forcats`: transforma uma variável qualitativa (`chr`) em fator (`fct`).

- `x`: primeiro vetor de texto;
- `levels`: valores possíveis da variável qualitativa, onde a ordem de inputação é a ordem implícita. Se não fornecida, `fct` usará a ordem de aparição;
- Todos os valores de `x` precisam ser um elemento de `levels`;

Um erro é produzido se um elemento do vetor não está nos níveis.

```
tp_cor_raca <- c("Branca", "Preta", "Amarela", "Índigena")
niveis <- c("Branca", "Preta", "Amarela")
```

```
fct_cor_raca <- fct(tp_cor_raca, levels = niveis)
```

```
# Error in `fct()`:
# ! All values of `x` must appear in `levels` or `na`
# i Missing level: "Índigena"
```

# 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))  
matriz_inteiros
```

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

## Matriz de valores lógicos

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

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

# Vetor, Fator e 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  $(0, 1 \quad 0, 2 \quad 0, 3 \quad 0, 4 \quad 0, 5)$

# Vetor, Fator e Matriz

## Exercício

Crie os seguintes fatores:

- 1 Crie um fator com a coluna `especies` para o conjunto de dados `iris.xlsx`;
- 2 Crie um fator com a coluna `tp_cor_raca` para a sua cidade. Dica: use o dicionário `dicionario_enem_2023.xlsx`;
- 3 Crie um fator com a coluna `tp_sexo` para a sua cidade. Dica: use o dicionário `dicionario_enem_2023.xlsx`; Crie um fator com a coluna `q006` para a sua cidade. Dica: use o dicionário `dicionario_enem_2023.xlsx`;
- 4 Crie um fator com a coluna `tp_sexo` para a sua cidade.

# Operações com vetores

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

- *Slicing*: extrair parte de um vetor
  - Operações básicas (operação, subtração, multiplicação e divisão ) realizada em cada elemento do 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] -5 -5 -5 -5 -5
```

### 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  $(1 \ 2 \ 3) \cdot (0,1 \ 0,05 \ 0,33)$ ;
- 4  $(1 \ 2 \ 3) / (0,1 \ 0,05 \ 0,33)$ ;
- 5 Recupere as/os coordenadoras/es do seguinte vetor:  
(Daniele Rodrigo Amanda Gilberto).

# 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

## Primeira matriz

```
matriz_1 <- rbind(  
  c(2, 4), # primeira linha  
  c(1, 5) # segunda linha  
)
```

## Segunda matriz

```
matriz_2 <- cbind(  
  c(23, 44), # primeira coluna  
  c(19, 12) # segunda coluna  
)
```

# Operações com matrizes

## Soma de duas matrizes

```
soma_matriz <- matriz_1 + matriz_2  
soma_matriz
```

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

# Operações com matrizes

## Subtração de matrizes

```
subtracao_matriz <- matriz_1 - matriz_2  
subtracao_matriz
```

```
#      [,1] [,2]  
# [1,]  -21  -15  
# [2,]  -43   -7
```

# Operações com matrizes

## Produto de matrizes

```
produto_matriz <- matriz_1 * matriz_2  
produto_matriz
```

```
#      [,1] [,2]  
# [1,]  46  76  
# [2,]  44  60
```

# Operações com matrizes

## Divisão de matrizes

```
divisao_matrizes <- matriz_1 / matriz_2  
divisao_matrizes
```

```
#           [,1]      [,2]  
# [1,] 0.08695652 0.2105263  
# [2,] 0.02272727 0.4166667
```

## Operações com matrizes

*Slicing*: extrair parte da matriz.

```
nome_vetor[vetor_linhas, vetor_colunas]
```

onde `vetor_linhas` é um vetor de posições na linha, e `vetor_colunas` é um vetor de posições na coluna.

---

```
equipe <- cbind(  
  c("Daniele", "Gilberto", "Ellen"),  
  c("Rodrigo", "Amanda", "Jalinson")  
)  
equipe
```

```
#      [,1]      [,2]  
# [1,] "Daniele" "Rodrigo"  
# [2,] "Gilberto" "Amanda"  
# [3,] "Ellen"    "Jalinson"
```

## Pegando a pessoa na primeira linha e na segunda coluna

```
equipe[1, 2]
```

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

## Pegando a pessoa a primeira e terceira linha, e primeira e a segunda coluna

```
equipe[c(1, 3), c(1, 2)]
```

```
#      [,1]      [,2]  
# [1,] "Daniele" "Rodrigo"  
# [2,] "Ellen"   "Jalinson"
```

## Pegando a terceira linha

```
equipe[3, ]
```

```
# [1] "Ellen"   "Jalinson"
```

## Pegando a segunda coluna

```
equipe[, 2]
```

```
# [1] "Rodrigo" "Amanda"  "Jalinson"
```

## 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

Faça as seguintes 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  $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix} / \begin{pmatrix} 0,1 & 0 \\ 0 & 0,5 \end{pmatrix};$

5 Recupere a primeira linha de  $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix};$

6 Recupere a segunda coluna de  $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix};$

7 Recupere o elemento da primeira linha e da segunda coluna de  $\begin{pmatrix} 1 & 0 \\ 2 & 0,5 \end{pmatrix}.$

# 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  $2 \times 2$

# Lista

Recuperando valores de uma lista:

- Pela posição: `lista[[2]]`;
- Pelo nome do campo: `lista$nome_do_campo`;

---

## Pela posição

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

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

## Pela posição

```
lista_info$nome
```

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

## Lista

*Slicing*: extrair parte de um lista (produz uma lista).

```
lista[vetor_de_posicoes]
```

onde `vetor_de_posicoes` é um vetor de posições da lista.

---

```
sub_lista <- lista_info[c(2, 4)]  
sub_lista
```

```
# $nome  
# [1] "Fulano"  
#  
# $cpf  
# [1] "12345678900"
```

# Lista

## Exercício

### Exercício:

- 1 Recupere o terceiro elemento de `informacoes_pessoais`;
- 2 Recupere a matriz de `informacoes_pessoais`;
- 3 Crie uma nova lista a partir de `informacoes_pessoais` com apenas nome e idade.

## 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 ;
- Cada coluna de um data frame é um vetor;
- Cada coluna tem um único tipo de dados;
- tibble usada para criar data frame.

## 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 frame

## Operações em um tibble

- Notação de lista pode ser usada em data frame;
- Notação de matriz pode ser usada em data frame;
- Em cada coluna, podemos usar notação de vetor.

---

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

---

## Notação de lista

Produz vetor.

```
primeira_coluna <- df[[1]]  
primeira_coluna
```

```
# [1] "João"      "Josué"      "Joaquim" "José"
```

```
idade <- df$idade  
idade
```

```
# [1] 20 21 23 32
```

## Notação de matriz

Produz um data frame.

```
primeiras_linhas <- df[c(1, 3), ]  
primeiras_linhas
```

```
# # A tibble: 2 x 2  
#   nome      idade  
#   <chr>    <dbl>  
# 1 João      20  
# 2 Joaquim   23
```

Produz vetor.

```
primeira_coluna <- df[[1]]  
primeira_coluna[1:2]
```

```
# [1] "João" "Josué"
```

```
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
```

## data frame

### Exercício

Realize as seguintes operações no *dataset* iris (disponível no R - não é necessário carregar no R):

- imprima um resumo sobre o *dataset* iris
- pegue as 5 primeiras linhas de iris
- pegue as 5 últimas linhas de iris
- pegue a terceira coluna de iris
- pegue a coluna Species de iris
- use a função `tibble` para criar o seguinte data frame:

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

## 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.
  - Coloque espaço entre operadores básicos: `+`, `-`, `*`, `==`, `/` e outros. Exceção: `^` e `**`.

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.

No OBEC, já fazemos isso.

Para mais detalhes, consulte esse guia do [curso-r: diretórios e .Rproj](#).

Carregando dados no R

# 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`

```
p_load(tidyverse)
p_load(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, 4
# $ largura_sepala      <dbl> 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2
# $ comprimento_petala <dbl> 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5, 1
# $ largura_petala     <dbl> 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2, 0
# $ 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 ;

Dados públicos geralmente usam esse formato.

***Preste atenção em como o seus dados estão armazenados!***

---

	Sistema imperial (UK, US, AU, NZ)	Sistema métrico (quase todo planeta)
Separador da parte decimal e inteira de um número	.	,
Agrupador de milhar	,	.
Separador de colunas em arquivos .csv	;	,
Função para ler arquivos .csv (pacote readr)	<code>read_csv</code>	<code>read_csv2</code>

---

## 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_mtcarrros <- read_csv2("dados/brutos/mtcarrros.csv")  
glimpse(dados_mtcarrros)
```

```
# 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, 2  
# $ cavalos_forca <dbl> 110, 110, 93, 110, 175, 105, 245, 62  
# $ 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, 3  
# $ velocidade <dbl> 16.46, 17.02, 18.61, 19.44, 17.02, 2  
# $ 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, 0.5, 0.
```

## 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

# 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 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(x**2)))
```

é exatamente a mesma coisa que executar

```
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(  
  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()
```

Sobre questionário

# Questionário

- Tenha o objetivo e a pergunta da pesquisa!
  - As questões do questionários precisam informações úteis para a pergunta da pesquisa;
  - A literatura da área do conhecimento te ajuda a elaborar as questões;
- Não fazer uma amostra auto-selecionada: resultados não confiáveis;
- As perguntas precisam ser explícitas, óbvias, simples, e de entendimento rápido;
- Preferencialmente use perguntas curtas ou com instruções extremamente breves;
- Não deixe questões ambíguas ou de dupla interpretação.

- **EVITE A TODO CUSTO QUESTÕES DE RESPOSTA ABERTA;**
- **EVITE A TODO CUSTO QUESTÕES DE SELEÇÃO MÚLTIPLA;**
- É preferível usar valores numéricos (evite categorizar os valores nas questões).

# Estatística Descritiva no R

# 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.*

## 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.

## Tipo de amostras

**Probabilística:** sorteio dos elementos da população.

- **Amostra aleatória simples:** sorteio dos elementos da população;
  - **Com reposição:** elementos da população podem ser escolhidos mais de uma vez;
  - **Sem reposição:** elementos da população podem ser escolhidos uma única vez;
- **Amostra sistemática:** elementos escolhidos de maneira ordenada e aleatória (quinta lâmpada na linha de produção);
- **Amostra Estratificada:** População é dividida em grupos distintos relevantes chamados de estratos e elementos são sorteados dos estratos;
- **Amostra por conglomerados:** População é dividida em seções chamados de conglomerados, sorteia-se os conglomerados e todos elementos dos conglomerados sorteados entram na amostra;

Podemos usar técnicas de inferência estatística.

**Não-probabilística:** elementos são selecionados por conveniência com critérios estabelecidos pelo/a pesquisador/a.

- **Conveniência:** elementos são escolhidos pela simplicidade de coleta sem qualquer sorteio (enquete disponível na internet);
- **Julgamento:** pesquisador escolhe intencionalmente os elementos que irão compor a amostra.

# Classificação de variáveis

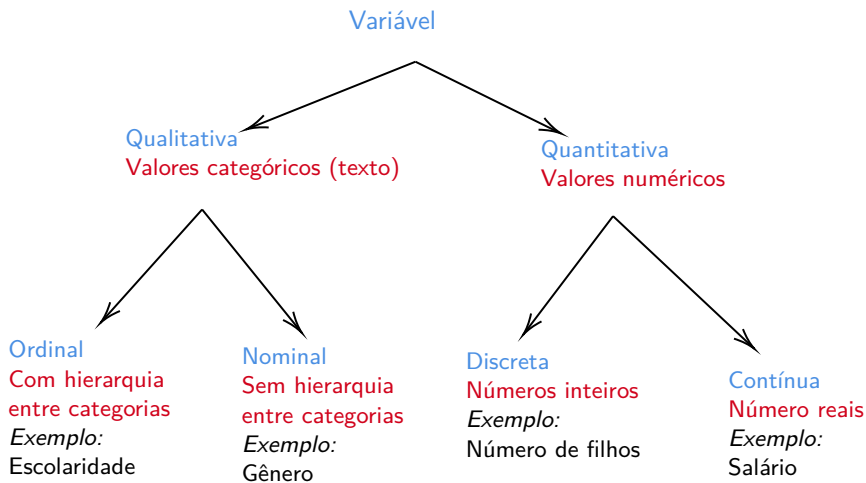


Figura 6: Classificação de variáveis.

## 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` e `adorn_totals` 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

Para mais detalhes, consulte a documentação oficial do *janitor*:  
[documentação de `tabyl`](#).

## Tabela de frequências

### Variável qualitativa

```
tab <- tabyl(dados_iris, especies) |> adorn_totals()  
tab
```

```
#   especies   n  percent  
#   setosa    50 0.3333333  
# versicolor  50 0.3333333  
#  virginica  50 0.3333333  
#      Total 150 1.0000000
```

# Tabela de frequências

## Variável qualitativa

### Exercício

Para o conjunto de dados ENEM 2023 (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()  
tab
```

#	carburadores	n	percent
#	1	7	0.21875
#	2	10	0.31250
#	3	3	0.09375
#	4	10	0.31250
#	6	1	0.03125
#	8	1	0.03125
#	Total	32	1.00000

# Tabela de frequências

## Variável quantitativa discreta

### Exercício

Para os dados do ENEM 2023 (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 7: Tabela de frequências para a variável quantitativa contínua.

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

- 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()
```

#	comprimento_sepala_int	n	percent
#	[4.3,4.81)	16	0.10666667
#	[4.81,5.33)	30	0.20000000
#	[5.33,5.84)	34	0.22666667
#	[5.84,6.36)	28	0.18666667
#	[6.36,6.87)	25	0.16666667
#	[6.87,7.39)	10	0.06666667
#	[7.39,7.9]	7	0.04666667
#	Total	150	1.00000000

# 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](#).

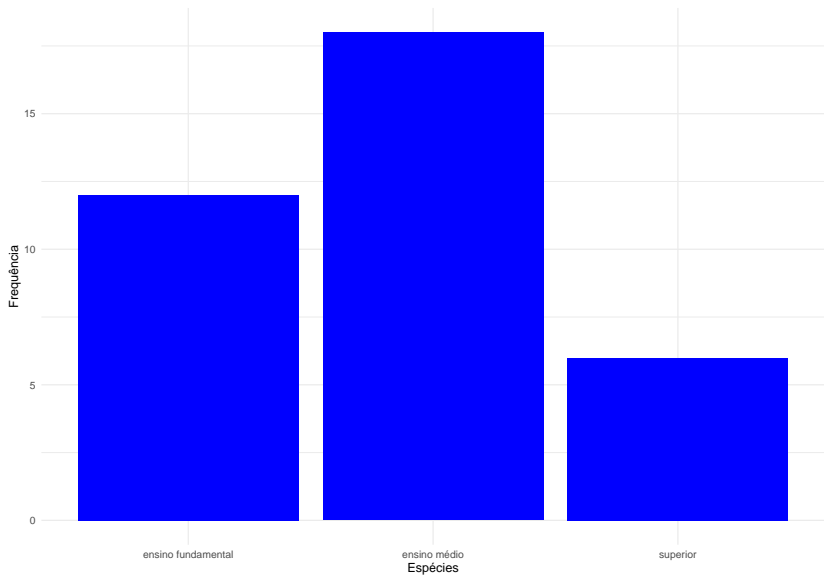
## Gráfico de barras no ggplot2

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

```
companhia <- read_xlsx("dados/brutos/companhia_mb.xlsx")
ggplot(companhia) +
  geom_bar(mapping = aes(escolaridade), fill = "blue") +
  labs(x = "Espécies", y = "Frequência") +
  theme_minimal()
```



# Gráfico de barras

## Variável qualitativa

### Exercício

Para o conjunto de dados ENEM 2023 (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()  
tab
```

#	carburadores	n	percent
#	1	7	0.21875
#	2	10	0.31250
#	3	3	0.09375
#	4	10	0.31250
#	6	1	0.03125
#	8	1	0.03125
#	Total	32	1.00000

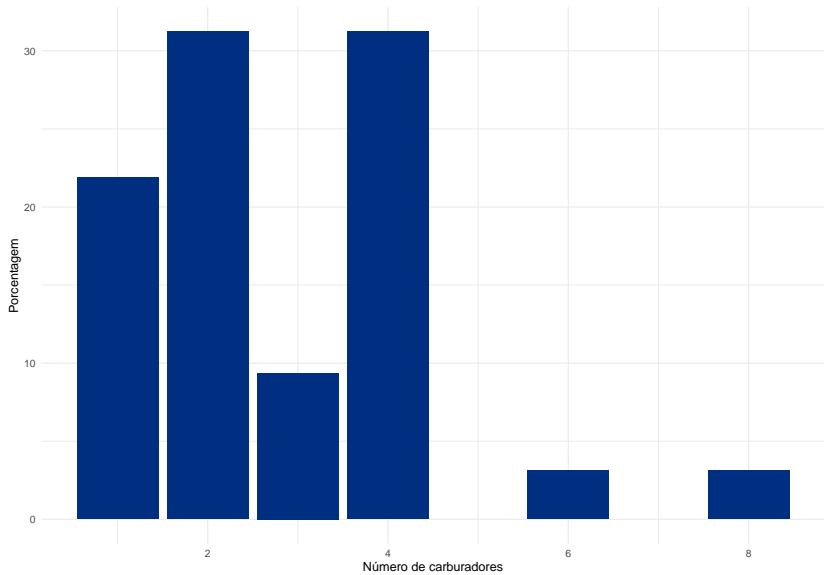
# Gráfico de barras

## 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 2023 (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 2023 (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 2023 (cada pessoa tem sua cidade), construa o gráfico de barras onde o eixo y é a porcentagem.

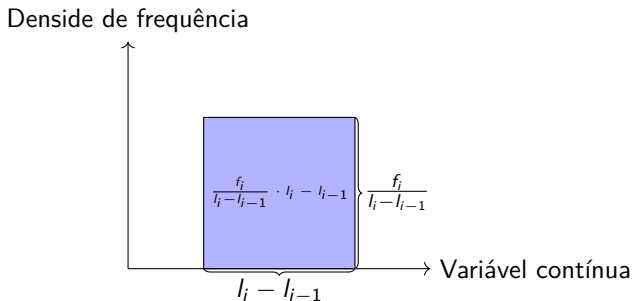
# Histograma

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

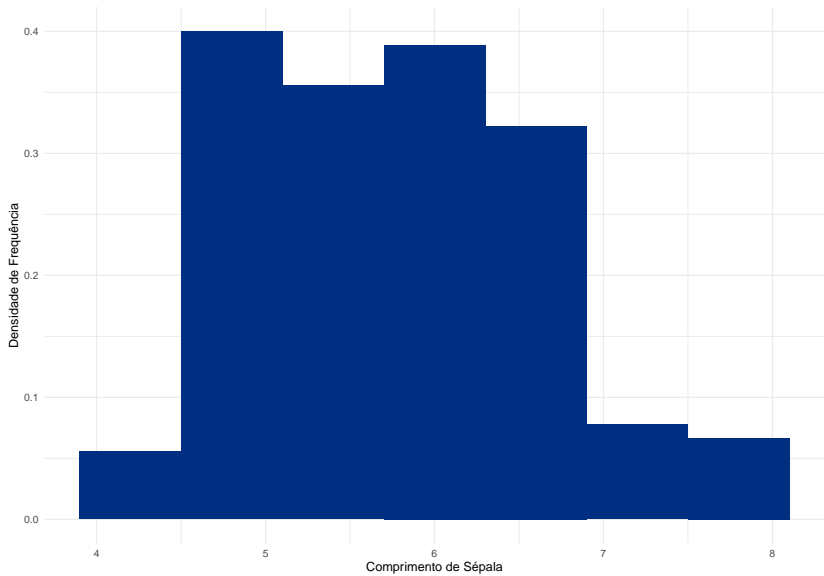
# Histograma

Figura 7: 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 2023 (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 2023 (cada pessoa tem sua cidade), construa o histograma usando a regra de Sturge.

## Medidas resumen

# 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.

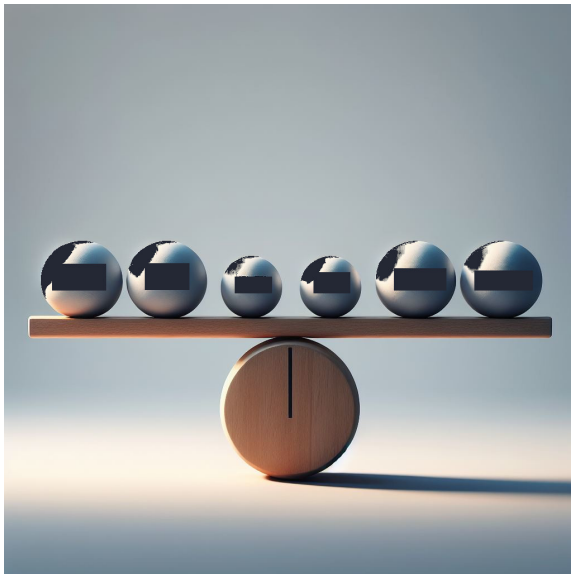
- **Média:**  $\bar{x} = \frac{x_1 + \dots + x_n}{n}$ .
  - mean
- **Mediana:** valor que divide a sequência ordenada de valores em duas partes iguais.
  - median
- **Moda:** valor mais frequente (apenas para variável qualitativa e variável quantitativa discreta)
  - `get_mode <- \ (x) names(table(x))[which.max(table(x))]`

## Medidas de dispersão

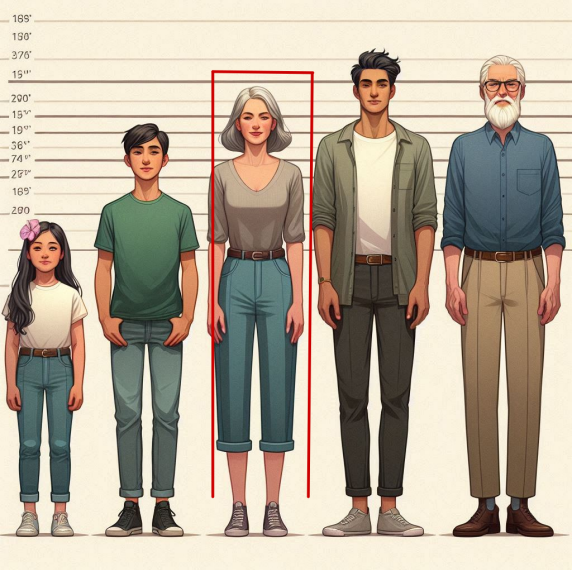
A ideia é medir a homogeneidade dos valores.

- **Variância:**  $s^2 = \frac{(x_1 - \bar{X})^2 + \dots + (x_n - \bar{X})^2}{n - 1}$ .
  - var
- **Desvio padrão:**  $s = \sqrt{s^2}$  (mesma unidade dos dados).
  - sd
- **Coefficient de variação:**  $cv = \frac{s}{\bar{x}}$  (sem unidade dos dados).
  - sd(x) / mean(x)

# Média



# Mediana



## Medidas de posição

```
get_mode <- \(x) names(table(x))[which.max(table(x))]  
companhia_mb <- read_xlsx("dados/brutos/companhia_mb.xlsx")  
resumo_mb <- companhia_mb |>  
  summarise(  
    media_salarial = mean(salario),  
    moda_procedencia = get_mode(procedencia),  
    mediana_filhos = median(numero_filhos)  
  )  
resumo_mb
```

```
# # A tibble: 1 x 3  
#   media_salarial moda_procedencia mediana_filhos  
#           <dbl> <chr>                <dbl>  
# 1           11.1 outra                    0
```

## Motivação dispersão

Medidas de posição podem ser insuficientes para pegar analisar variáveis.

---

```
notas_estudantes <- read_xlsx("dados/brutos/notas_estudantes.xlsx")
medias <- notas_estudantes |>
  summarise(
    media_mt = mean(matematica),
    media_pt = mean(portugues),
    media_cn = mean(ciencias_naturais),
    media_ch = mean(ciencias_humanas)
  )
medias
```

```
# # A tibble: 1 x 4
#   media_mt media_pt media_cn media_ch
#   <dbl>     <dbl>     <dbl>     <dbl>
# 1         5         5         5         5
```

```
medianas <- notas_estudantes |>
  summarise(
    media_mt = median(matematica),
    media_pt = median(portugues),
    media_cn = median(ciencias_naturais),
    media_ch = median(ciencias_humanas)
  )
medianas
```

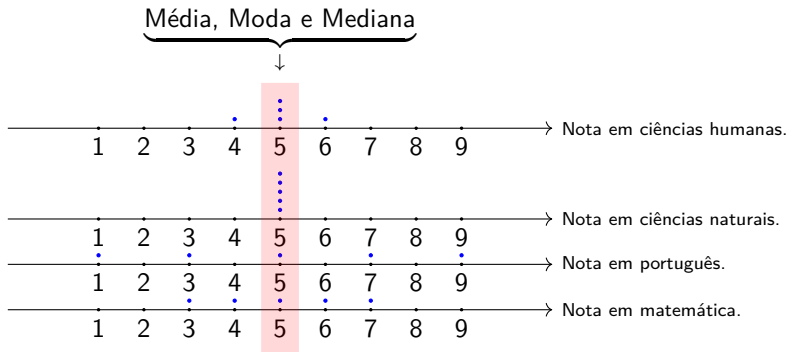
```
# # A tibble: 1 x 4
#   media_mt media_pt media_cn media_ch
#   <dbl>    <dbl>    <dbl>    <dbl>
# 1         5         5         5         5
```

```
get_mode <- \(x) names(table(x))[which.max(table(x))]  
modas <- notas_estudantes |>  
  summarise(  
    media_mt = get_mode(matematica),  
    media_pt = get_mode(portugues),  
    media_cn = get_mode(ciencias_naturais),  
    media_ch = get_mode(ciencias_humanas)  
  )  
modas
```

```
# # A tibble: 1 x 4  
#   media_mt media_pt media_cn media_ch  
#   <chr>    <chr>    <chr>    <chr>  
# 1 3        1         5         5
```

## Os desempenhos das/os estudantes nas 4 disciplinas são iguais?

Figura 8: Representação gráfica para nota em matemática, português, ciências naturais e ciências humanas.



Não! A dispersão/distribuição das notas são diferentes!

## Medidas de dispersão:

- Desvio padrão
  - mensura dispersão/distribuição dos valores observados
  - valores pequenos para valores próximos da média (**homogeneidade**)
  - valores grandes para valores distantes da média (**heterogeneidade**)
- Coeficiente de variação:
  - mensura dispersão/distribuição dos valores observados
  - sem unidade de mensuração
  - valores pequenos para valores próximos da média (**homogeneidade**)
  - valores grandes para valores distantes da média

```
dp <- notas_estudantes |>
  summarise(
    dp_mt = sd(matematica),
    dp_pt = sd(portugues),
    dp_cn = sd(ciencias_naturais),
    dp_ch = sd(ciencias_humanas)
  )
dp
```

```
# # A tibble: 1 x 4
#   dp_mt dp_pt dp_cn dp_ch
#   <dbl> <dbl> <dbl> <dbl>
# 1  1.58  3.16     0 0.707
```

```
cv <- notas_estudantes |>
  summarise(
    cv_mt = sd(matematica) / mean(matematica),
    cv_pt = sd(portugues) / mean(portugues),
    cv_cn = sd(ciencias_naturais) / mean(ciencias_naturais),
    cv_ch = sd(ciencias_humanas) / mean(ciencias_naturais)
  )
cv
```

```
# # A tibble: 1 x 4
#   cv_mt cv_pt cv_cn cv_ch
#   <dbl> <dbl> <dbl> <dbl>
# 1 0.316 0.632     0 0.141
```

## 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 2023 (cada pessoa tem sua cidade) por gênero (`tpsexo`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_cn` do conjunto de dados ENEM 2023 (cada pessoa tem sua cidade) por gênero (`tpsexo`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_mt` do conjunto de dados ENEM 2023 (cada pessoa tem sua cidade) por raça (`tpcor_raca`).
- Calcule média, mediana, o desvio padrão e coeficiente de variação para a variável `nu_nota_cn` do conjunto de dados ENEM 2023 (cada pessoa tem sua cidade) por raça (`tpcor_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

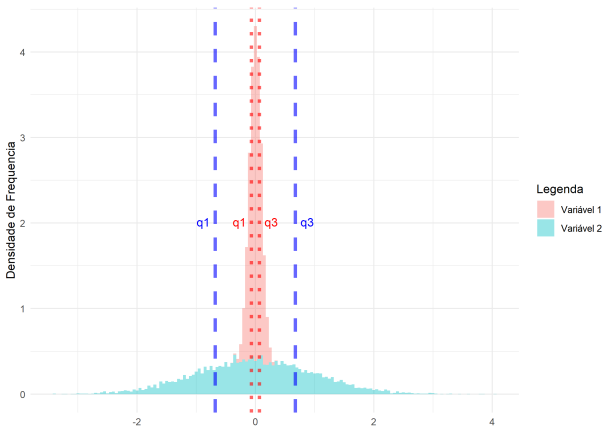
- Calcule o primeiro quartil, segundo quartil e o terceiro quartil para a variável `nu_nota_mt` do conjunto de dados ENEM 2023 (cada pessoa tem sua cidade) por gênero (`tp_sexo`). Inclua uma coluna com a frequência da variável `tp_sexo`.
- Calcule o primeiro quartil, segundo quartil e o terceiro quartil para a variável `nu_nota_cn` do conjunto de dados ENEM 2023 (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 2023 (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 2023 (cada pessoa tem sua cidade) por raça (`tp_cor_raca`). Inclua uma coluna com a frequência da variável `tp_cor_raca`.

Diagrama de caixa

# 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:**

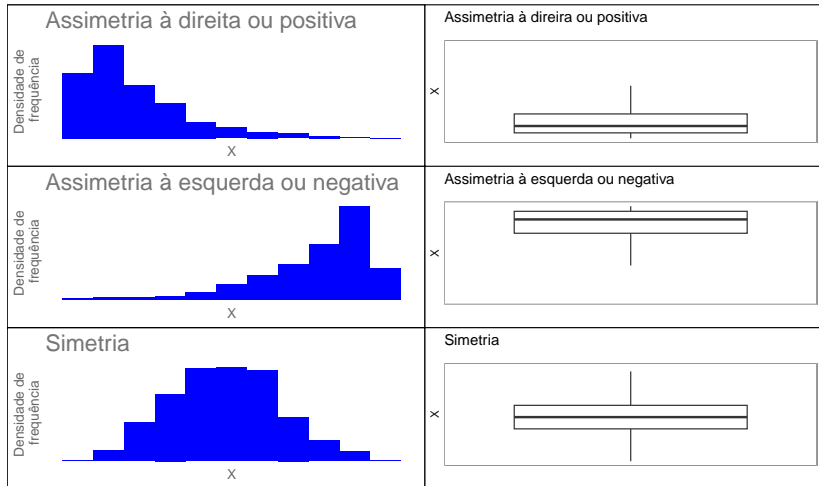
- frequências diminuem à direita no histograma
  - $q_2$  perto  $q_1$ :  $q_2 - q_1 < q_3 - q_2$
- 

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

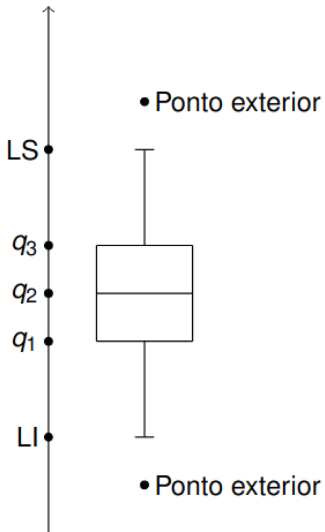
- frequências diminuem à esquerda no histograma
- $q_2$  perto  $q_3$ :  $q_2 - q_1 > q_3 - q_2$

# Diagrama de caixa (ou *boxplot*)

## Assimetria

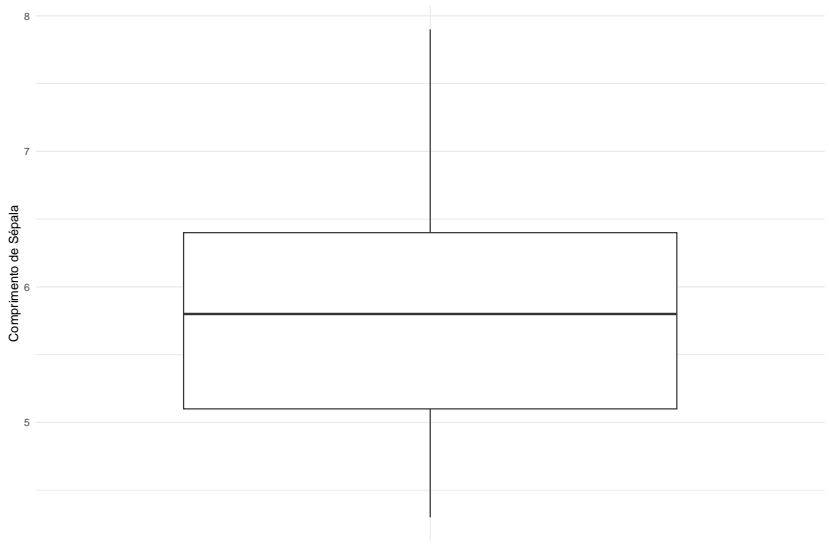


## Diagrama de caixa (ou *boxplot*)



## Diagrama de caixa (ou *boxplot*)

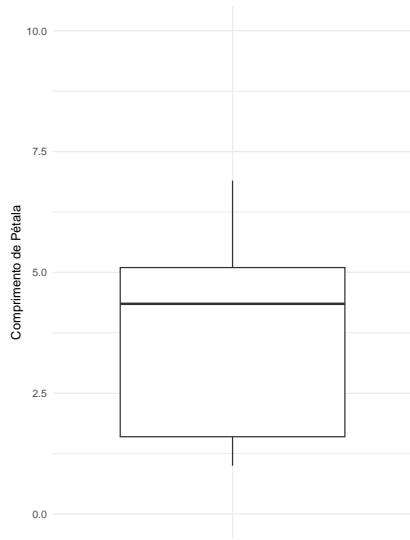
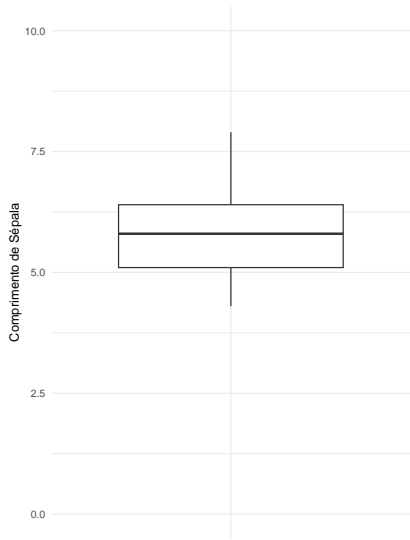
```
ggplot(dados_iris) +  
  geom_boxplot(aes(x = "", y = comprimento_sepala)) +  
  labs(x = "", y = "Comprimento de Sépala") +  
  theme_minimal()
```



## Gráficos lado a lado com patchwork

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

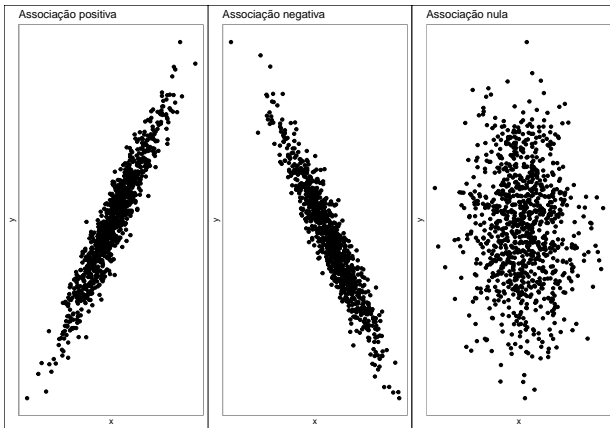
### Exercício

Para o conjunto de dados ENEM 2023 (cada pessoa tem sua cidade), construa o diagrama de caixa para as variáveis `nu_notas_mt` e `nu_notas_cn` e os coloque lado a lado usando o pacote `patchwork`.

Associação entre duas variáveis

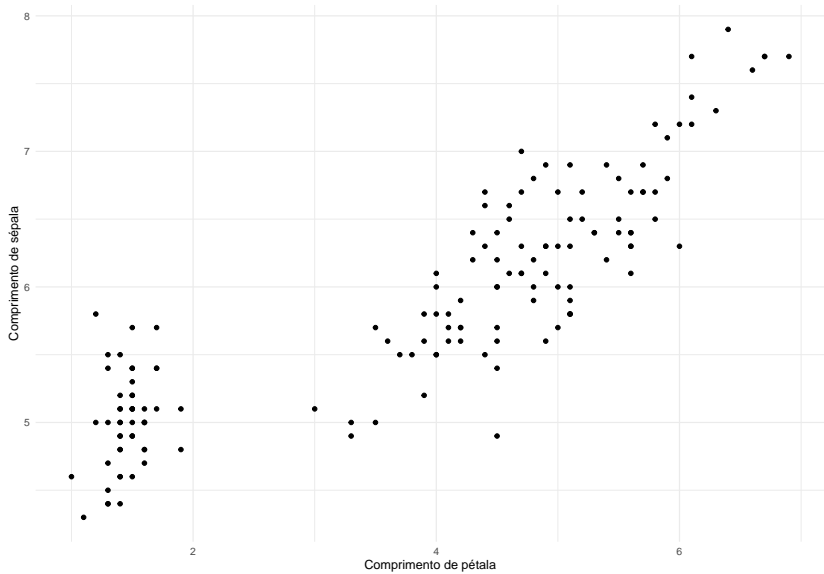
# Gráficos Duas variáveis

**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()
```



# Coeficiente de correlação linear de Pearson

Mensura a associação entre duas variáveis quantitativas.

- está entre  $-1$  e  $1$
- valores negativos indicam associação negativa
- valores positivos indicam associação positiva
- valores próximos de  $0$  indicam a ausência de associação

Usamos a função `cor` para calcular a correlação entre duas variáveis quantitativas.

## Coeficiente de correlação linear de Pearson

A largura e comprimento de pétalas estão associadas?

```
dados_iris <- read_xlsx("dados/brutos/iris.xlsx")
coef_cor <- cor(
  dados_iris$comprimento_petala,
  dados_iris$largura_petala
)
coef_cor
```

```
# [1] 0.9628654
```

Aparentemente, largura e comprimento das pétalas das flores de iris estão associadas.

# Associação entre duas variáveis quantitativas

## Exercício

Para o conjunto de dados ENEM 2023 (cada pessoa tem sua cidade), construa o gráfico de dispersão entre as variáveis `nu_nota_mt` e `nu_nota_cn`. Calcule coeficiente de correlação linear de Pearson.

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:

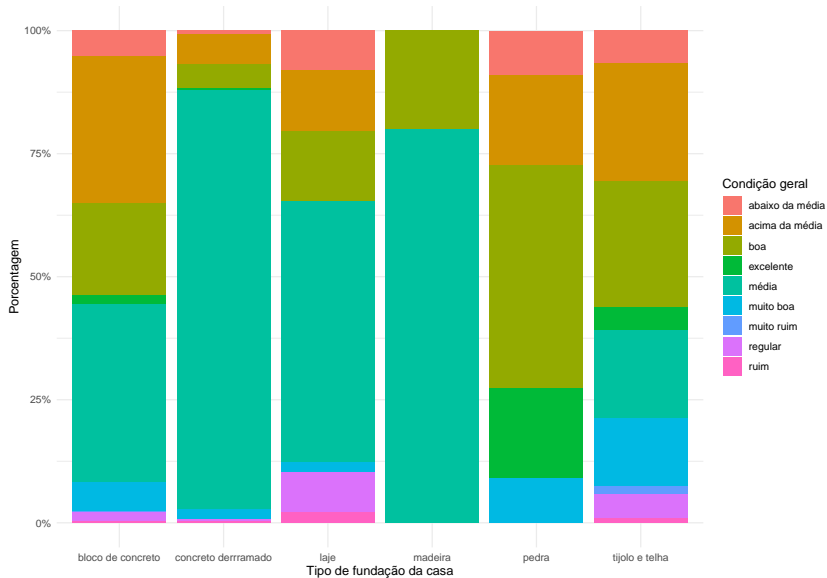
- **não associados:** se ao conhecermos o valor de  $Y$  para um elemento da população, **continuamos** com a percentagem  $100 \cdot f_i \%$  deste elemento ter valor de  $X$  igual a  $A_i$
- **associados:** se ao conhecermos o valor de  $Y$  para um elemento da população, **alteramos** a percentagem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

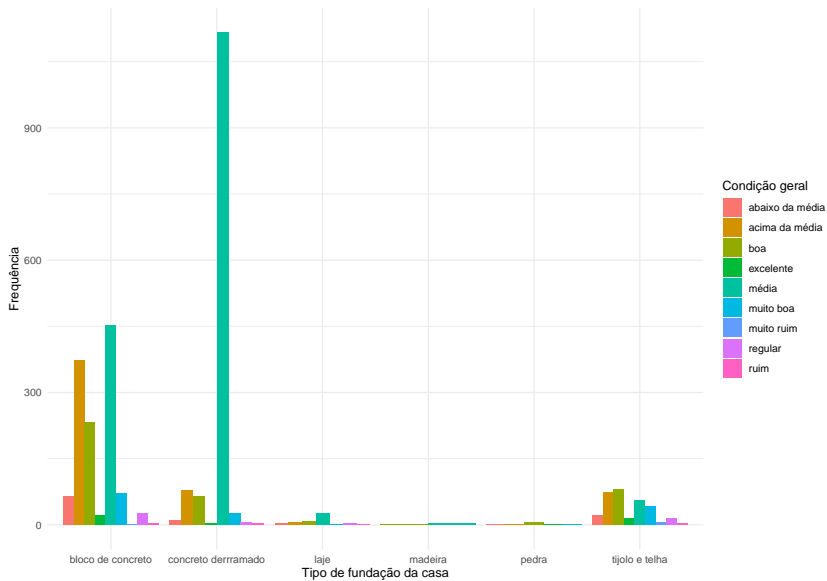


# Associação entre duas variáveis qualitativas

## 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()
```



# Coeficiente V de Cramer

Mensura a associação entre duas variáveis qualitativas.

- está entre 0 e 1
- valores próximos de 0 indicam ausência de associação
- valores próximos de 1 indicam associação

Vamos usar a função `CramerV` do pacote `DescTools`.

## Coeficiente V de Cramer

Existe associação entre o tipo de fundação da casa (`fundacao_tipo`) e a condição geral da casa (`geral_condicao`)?

```
coef_associacao <- CramerV(  
  dados_casas$fundacao_tipo,  
  dados_casas$geral_condicao,  
  correct = T  
)  
coef_associacao
```

```
# [1] 0.2535774
```

Aparentemente, existe uma fraca associação entre `fundacao_tipo` e `geral_condicao`.

# Associação entre duas variáveis qualitativas

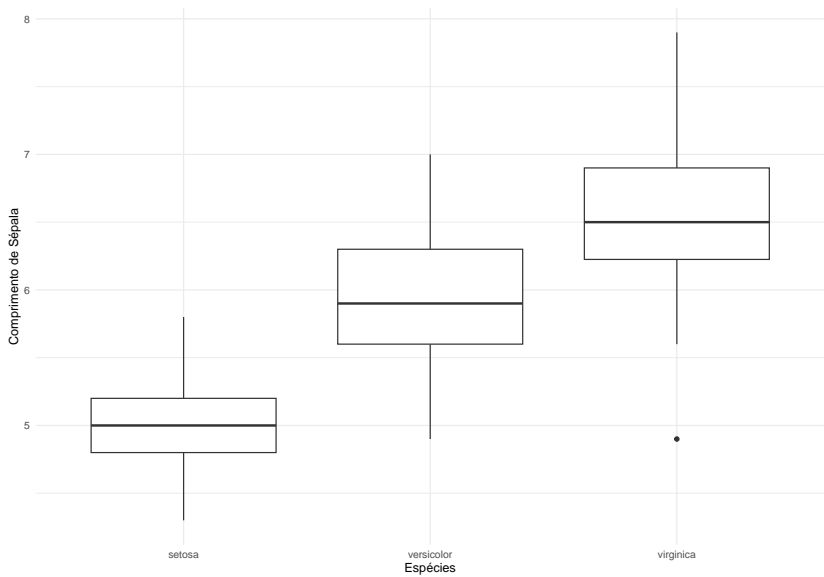
## Exercício

- Verifique se existe associação entre as variáveis `q006` e `tp_cor_raca` do conjunto de dados ENEM 2023 (cada pessoa tem sua cidade) usando gráfico de gráficos usando o `position=fill`. Calcule o coeficiente  $V$  de Cramer.
- Verifique se existe associação entre as variáveis `q006` e `tp_sexo` do conjunto de dados ENEM 2023 (cada pessoa tem sua cidade) usando gráfico de gráficos usando o `position=dodge`. Calcule o coeficiente  $V$  de Cramer.

## 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 2023 (cada pessoa tem sua cidade), compare a variável `nu_nota_mt` por raça (`tp_cor_raca`).
- Para o conjunto de dados ENEM 2023 (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`.