

R para Ciência de Dados

Ciência de Dados para Docentes usando o R

Profa Carolina Paraíba e Prof Gilberto Sassi

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

Julho de 2023

Ciência de Dados para Docentes usando o R

- Introdução ao R
- Estatística Descritiva
 - Conceitos Básicos
 - Classificação de Variáveis
 - Tabelas e gráficos para variável qualitativa
 - Tabelas e gráficos para variável quantitativa discreta
 - Tabelas e gráficos para variável quantitativa contínua
 - Medidas Resumo
 - Boxplot
 - Gráfico temporal
 - Gráfico de dispersão
- Introdução à Regressão Linear Simples

Preparando o ambiente

- Você precisa de um computador para acompanhar as aulas.
- Usaremos nas aulas: RStudio Cloud.
- No seu dia-a-dia, recomendamos instalar o R com versão pelo menos 4.1: cran.r-project.org.
- **IDE** recomendada: *RStudio* e
- Neste curso, usaremos o *framework* **tidyverse**:
 - Instale o framework a partir do repositório CRAN:
`install.packages("tidyverse")`
- Outras linguagens interessantes: `python` e `julia`.
 - `python`: linguagem interpretada de 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: uma introdução

O precursor do R: S.

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

História do R

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

Porque usar o R

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

Onde estudar fora da 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.

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 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

Divisão

3 / 2

[1] 1.5

Potenciação

2^3

[1] 8

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`.
- **Estrutura de dados Heterôgenea:** `list` e `data.frame` (`tibble` no `tidyverse`).

Tipo de dados no R

Número inteiro

```
a <- 1L  
typeof(a)  
## [1] "integer"
```

Número real

```
b <- 1.2  
typeof(b)  
## [1] "double"
```

Número complexo

```
d <- 1 + 1i  
typeof(d)  
## [1] "complex"
```

Tipo de dados no R

Número lógico

```
typeof(TRUE)
```

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

Caracter

```
cor <- "Vermelho"
```

```
typeof(cor)
```

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

Estrutura de dados homogênea

Vetor

- Agrupamento de valores de mesmo tipo em um único objeto.
- Criação de vetor: `c(...)` e `vector('<tipo de dados>', <comprimento do vetor>)`, `seq(from = a, to = b, by = c)`.

Vetor de caracteres

```
cores <- c("Vermelho", "Verde")
```

```
cores
```

```
## [1] "Vermelho" "Verde"
```

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

```
b
```

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

Estrutura de dados homogênea

Vetor de números reais

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

```
a
```

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

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

```
b
```

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

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

```
d
```

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

Estrutura de dados homogênea

Vetor de números inteiros

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

```
a
```

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

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

```
b
```

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

```
d <- 1:4
```

```
d
```

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

Estrutura de dados homogênea

Vetor lógico

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

```
a
```

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

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

```
b
```

```
## [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>)` ou `diag(<vector>)`.

Matriz de caracteres

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

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


Estrutura de dados homogênea

Matriz de números reais

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

```
##      [,1] [,2]
## [1,]  0.0  1.0
## [2,]  0.5  1.5
```

Matriz de inteiros

```
a <- matrix(1L:4L, nrow = 2)
a
```

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

Estrutura de dados homogênea

Matriz de valores lógicos

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

```
a
```

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

Estrutura de dados homogênea

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 (não precisa ser vetor numérico).

Slicing

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

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

Estrutura de dados homogênea

Adição (vetores numéricos)

```
a <- 1:5
```

```
b <- 6:10
```

```
a + b
```

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

Subtração (vetores numéricos)

```
a <- 1:5
```

```
b <- 6:10
```

```
b - a
```

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

Estrutura de dados homogênea

Multiplicação (vetores numéricos)

```
a <- 1:5  
b <- 6:10  
b * a
```

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

Divisão (vetores numéricos)

```
a <- 1:5  
b <- 6:10  
b / a
```

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

Estrutura de dados homogênea

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

- Operações básicas (operação, subtração, multiplicação e divisão) realizada em cada elemento das matrizes.
- Multiplicação de matrizes (vide multiplicação de matrizes), inversão de matrizes (vide inversão de matrizes), matriz transposta (vide matriz transposta), determinante (vide determinante de uma matriz) e solução de sistema de equações lineares (vide sistema de equações lineares).

Estrutura de dados heterogênea

- Agrupamento de dados em tabela, onde: cada coluna é uma variável; cada linha é uma observação.
- Criação de tibble: `tibble(...)` e `tribble(...)`.

tibble (data frame)

```
library(tidyverse)

a <- tibble(variavel_1 = c(1, 2), variavel_2 = c("a", "b"))
glimpse(a)

## Rows: 2
## Columns: 2
## $ variavel_1 <dbl> 1, 2
## $ variavel_2 <chr> "a", "b"
a

## # A tibble: 2 x 2
##   variavel_1 variavel_2
##       <dbl> <chr>
## 1         1      1 a
## 2         2      2 b
```


Estrutura de dados heterogênea

Operações em um tibble

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

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

Estrutura de dados heterogênea

Concatenação de listas

```
a <- list("a", "b")  
b <- list(1, 2)  
d <- c(a, b)  
d
```

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

Estrutura de dados heterogênea

Slicing a lista

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

Acessando o valor de elemento em uma lista

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

Acessando elementos em uma lista usando \$

```
d <- list(elemento_1 = 1, elemento_2 = "docente")
d$elemento_2
## [1] "docente"
```

Estrutura de dados heterogênea

Slicing uma lista com ["nome"]

```
d <- list(elemento_1 = 1, elemento_2 = "docente",
          elemento_3 = list("olá"))
d["elemento_3"]

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

Obtendo os nomes dos elementos em um lista

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

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

Valores especiais no R

Valores especiais	Descrição	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: qualquer pessoa precisa entender o que este objeto é ou faz.

Parênteses 1: guia de estilo no R

- Use a convenção do R:
 - Use apenas letras minúsculas, números e *underscore* (comece sempre com letras minúsculas).
 - Nomes de objetos precisam ser substantivos e precisam descrever o que este objeto é ou faz (seja conciso, direto e significativo).
 - Evite ao máximo os nomes que já são usados (*buit-in*) no R.
 - 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.
 - codigos: código fonte do seu projeto.
 - figuras: figuras criadas no seu projeto.
 - resultados: outros arquivos que não são figuras.
 - antigo: arquivos da versão anterior do projeto.
 - notas: notas de reuniões e afins.
 - relatorio (ou artigo): documento final de seu projeto.
 - 'referencias': livros, artigos e qualquer coisa que são referências em seu projeto.
- Para mais detalhes, consulte esse guia do curso-r: diretórios e `.Rproj`.

Lendo 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`.

Lendo dados no R

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

```
library(readxl)
```

```
library(tidyverse)
```

```
dados <- read_xlsx("../dados/brutos/estudantes.xlsx")
```

Salvando dados no R

Salvar no formato **.xlsx**

- **Pacote:** `writexl` do tidyverse (instale com o comando `install.packages('writexl')`)
- Parâmetros da função `write_xlsx` (para ler arquivos `.xlsx`):
 - `path`: caminho até o arquivo.
 - `col_names`: Argumento lógico com valor padrão igual a `TRUE`. Indica se a primeira linha tem o nome das variáveis.
 - `format_headers`: Argumento lógico com valor padrão igual a `TRUE`. Indica que os nomes das colunas no arquivo `.xlsx` estarão centralizados e em negrito.
- Para mais detalhes, consulte a documentação oficial do *writexl*: documentação de `writexl`.

Salvando dados no R

Salvar no formato .xlsx

```
library(writexl)

dados_selecionado <- dados |>
  select(sexo, idade, nota)

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

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:** característica numérica da amostra. Em geral, usamos uma estatística 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 e letras minúsculas do alfabeto latino para representar o valor observado da variável em um elemento da amostra.

Conceitos Básicos

Exemplo:

- **População:** Todos os residentes da cidade de Salvador com 25 anos ou mais.
- **Amostra:** 5 residentes da cidade de Salvador com 25 anos ou mais *selecionados segundo um plano de amostragem probabilística*.
- **Variável:** salário em R\$ (denotado pela letra X).
- **Parâmetro:** *salário médio* da população de residentes da cidade de Salvador com 25 anos ou mais (denotado pela letra grega μ).
- **Estatística:** *salário médio* da amostra de 20 residentes da cidade de Salvador com 25 anos ou mais.

Conceitos Básicos

Exemplo (continuação):

Suponha que foi selecionada uma amostra de $n = 5$ residentes da cidade de Salvador com 25 anos ou mais para os quais foi observada a variável salário em R\$.

Tabela 3: Salário em R\$ de uma amostra de 5 residentes da cidade de Salvador com 25 anos ou mais.

Elemento da amostra	Salário
1	843.95
2	876.98
3	1055.87
4	907.05
5	912.93

Conceitos Básicos

Exemplo (continuação):

Para este exemplo, temos que:

- Variável: X : salário em R\$.
- Valores observados de X : x_i : valor observado da variável no i -ésimo elemento da amostra, $i = 1, 2, 3, 4, 5$: 843.95; 876.98; 1055.87; 907.05; 912.93
- Parâmetro: μ : salário médio dos residentes da cidade de Salvador com 25 anos ou mais.
- Estatística: média amostral: $\bar{x} = \frac{x_1 + x_2 + x_3 + x_4 + x_5}{n}$.

Conceitos Básicos

Exemplo (continuação):

Média amostral:

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + x_3 + x_4 + x_5}{n} \\ &= \frac{843.95 + 876.98 + 1055.87 + 907.05 + 912.93}{5} \\ &= 919.356.\end{aligned}$$

Classificação de Variáveis

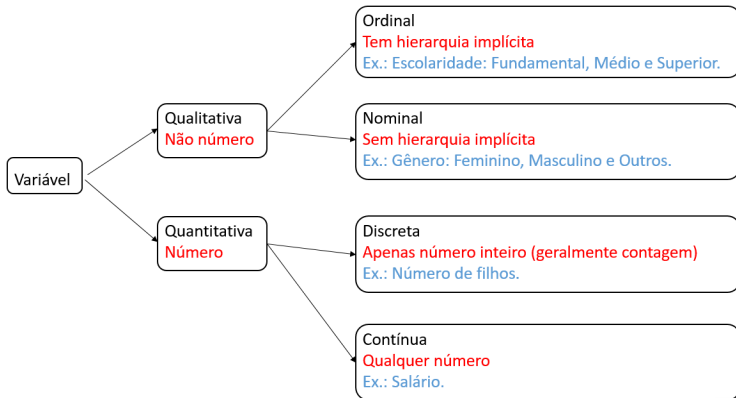


Figura 1: Classificação de variáveis.

Tabelas e gráficos para 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.

Tabelas e gráficos para variável qualitativa

- **Pacote:** `tabyl` do `janitor` (instale com o comando `install.packages('janitor')`).
- Parâmetros da função `tabyl`:
 - `dat`: *data frame* ou vetor com os valores da variável que desejamos tabular.
 - `var1`: nome da primeira variável.
 - `var2`: nome da segunda variável (opcional).
- Para mais detalhes, consulte a documentação oficial do *janitor*: documentação de `tabyl`.

Tabelas e gráficos para variável qualitativa

Tabela de frequência:

```
tab <- tabyl(dados, sexo) |>
  adorn_totals() |>
  adorn_pct_formatting(digits = 2) |>
  rename (
    "Sexo" = sexo, "Frequência" = n,
    "Porcentagem" = percent)
```

Tabelas e gráficos para variável qualitativa

Tabela de frequência:

```
tab
```

##	Sexo	Frequência	Porcentagem
##	feminino	29	29.00%
##	masculino	71	71.00%
##	Total	100	100.00%

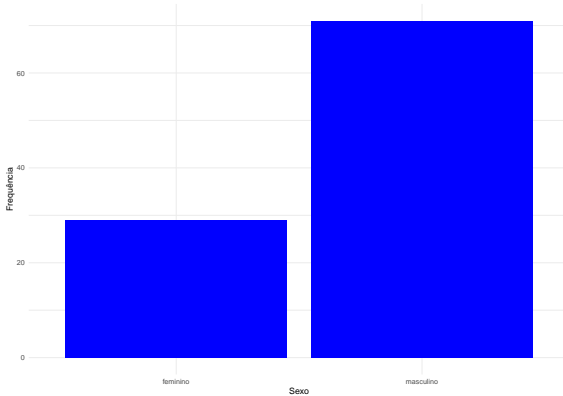
Tabelas e gráficos para variável qualitativa

Gráfico de barras:

```
ggplot(dados) +  
  geom_bar(mapping = aes(sexo), fill = "blue") +  
  labs(x = "Sexo", y = "Frequência") +  
  theme_minimal()
```

Tabelas e gráficos para variável qualitativa

Gráfico de barras:



Tabelas e gráficos para variável quantitativa discreta

A primeira coisa que fazemos é contar!

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 e $\{x_1, x_2, \dots, x_k\}$ são os números que são os valores únicos de X na amostra.

Tabelas e gráficos para variável quantitativa discreta

Tabela de frequência:

```
tab <- tabyl(dados, disciplinas) |>
  adorn_totals() |>
  adorn_pct_formatting(digits = 2) |>
  rename (
    "Disciplinas" = disciplinas, "Frequência" = n,
    "Porcentagem" = percent)
```

Tabelas e gráficos para variável quantitativa discreta

Tabela de frequência:

tab

##	Disciplinas	Frequência	Porcentagem
##	2	6	6.00%
##	3	21	21.00%
##	4	45	45.00%
##	5	28	28.00%
##	Total	100	100.00%

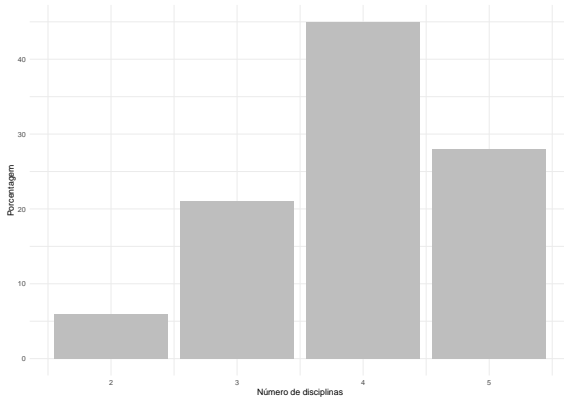
Tabelas e gráficos para variável quantitativa discreta

Gráfico de barras:

```
ggplot(dados) +  
  geom_bar(aes(disciplinas, after_stat(100 * prop))  
    fill = "grey") +  
  labs(x = "Número de disciplinas",  
    y = "Porcentagem") +  
  theme_minimal()
```

Tabelas e gráficos para variável quantitativa discreta

Gráfico de barras:



Tabelas e gráficos para 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	$f_k = \frac{n_k}{n_1 + \dots + n_k}$	$p_k = f_k \cdot 100$

Tabelas e gráficos para variável quantitativa contínua

Em que:

- $\min = l_0 \leq l_1 \leq \dots \leq l_{k-1} \leq l_k = \max$ (min é o menor valor do suporte da variável X e max é o maior valor do suporte da variável 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 (ou pelo regulador).

Recomendação: 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 $k = \lceil 1 + \log_2(n) \rceil$.

Tabelas e gráficos para variável quantitativa contínua

Tabela de frequência:

```
k <- ceiling(1 + log(nrow(dados)))

dados <- mutate(
  dados,
  idade_int = cut(
    idade, breaks = k,
    include.lowest = TRUE,
    right = FALSE))
```


Tabelas e gráficos para variável quantitativa contínua

Tabela de frequência:

```
tab <- tabyl(dados, idade_int) |>
  adorn_totals() |>
  adorn_pct_formatting(digits = 2) |>
  rename(
    "Idade" = idade_int,
    "Frequência absoluta" = n,
    "Porcentagem" = percent)
```

Tabelas e gráficos para variável quantitativa contínua

Tabela de frequência:

tab

##	Idade	Frequência absoluta	Porcentagem
##	[18,19.5)	2	2.00%
##	[19.5,21)	4	4.00%
##	[21,22.5)	31	31.00%
##	[22.5,24)	21	21.00%
##	[24,25.5)	31	31.00%
##	[25.5,27]	11	11.00%
##	Total	100	100.00%

Tabelas e gráficos para variável quantitativa contínua

Tabela de frequência:

```
limites <- c(18, 20, 22, 24, 26, 28)

dados <- dados |>
  mutate(idade_int = cut(dados$idade,
                        breaks = limites,
                        include.lowest = T,
                        right = F))

tab <- dados |>
  tabyl(idade_int) |>
  adorn_totals() |>
  adorn_pct_formatting(digits = 2) |>
  rename("Idade" = idade_int,
         "Frequência absoluta" = n,
         "Porcentagem" = percent)
```

Tabelas e gráficos para variável quantitativa contínua

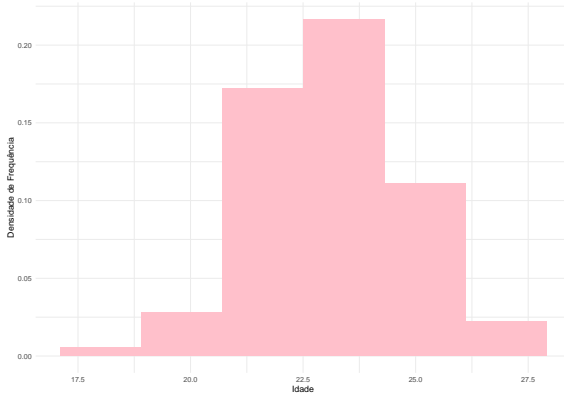
Histograma:

```
k <- ceiling(1 + log(nrow(dados)))

ggplot(dados) +
  geom_histogram(
    aes(x = idade, y = after_stat(density)),
    bins = k,
    fill = "pink") +
  theme_minimal() +
  labs(x = "Idade", y = "Densidade de Frequência")
```

Tabelas e gráficos para variável quantitativa contínua

Histograma:



Tabelas e gráficos para variável quantitativa contínua

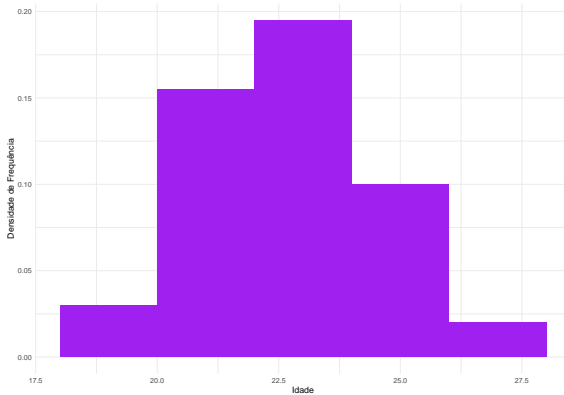
Histograma:

```
limites <- c(18, 20, 22, 24, 26, 28)
```

```
ggplot(dados) +  
  geom_histogram(  
    aes(x = idade, y = after_stat(density)),  
    breaks = limites,  
    fill = "purple") +  
  theme_minimal() +  
  labs(x = "Idade", y = "Densidade de Frequência")
```

Tabelas e gráficos para variável quantitativa contínua

Histograma:



Medidas Resumo

As medidas resumo são obtidas apenas para variáveis quantitativas discretas ou contínuas.

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).
- **Coeficiente de variação** $cv = \frac{s}{\bar{x}} \cdot 100\%$ (adimensional, ou

Medidas Resumo

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

```
sum_idade <- dados |>
  summarise(
    media = mean(idade),
    mediana = median(idade),
    dp = sd(idade),
    cv = dp / media)
```

Medidas Resumo

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

```
sum_idade
```

```
## # A tibble: 1 x 4
##   media mediana    dp      cv
##   <dbl>   <dbl> <dbl>  <dbl>
## 1  23.2      23  1.81  0.0780
```

Medidas Resumo

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

```
dados |> group_by(sexo) |>  
  summarise(  
    media = mean(idade),  
    mediana = median(idade),  
    dp = sd(idade),  
    cv = dp / media)
```

```
## # A tibble: 2 x 5  
##   sexo      media mediana    dp      cv  
##   <chr>    <dbl>   <dbl> <dbl>  <dbl>  
## 1 feminino  23.3      23  1.89  0.0812  
## 2 masculino 23.2      23  1.79  0.0772
```

Medidas Resumo

- **Quantil:** denotado por $q(p)$, é um valor que satisfaz:
 - $p \times 100\%$ das observações é no máximo $q(p)$;
 - $(1 - p) \times 100\%$ das observações é no mínimo $q(p)$.
- **Quartis:** dividem o conjunto de dados em quatro partes.
 - Primeiro quartil: $q_1 = q(1/4)$.
 - Segundo quartil: $q_2 = q(1/2)$.
 - Terceiro quartil: $q_3 = q(3/4)$.

Medidas Resumo

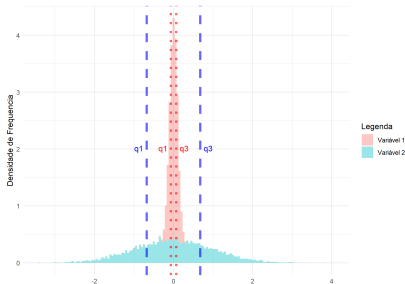
```
dados |> group_by(sexo) |>  
  summarise(  
    q1 = quantile(idade, 0.25),  
    q2 = quantile(idade, 0.5),  
    q3 = quantile(idade, 0.75),  
    frequencia = n())
```

```
## # A tibble: 2 x 5  
##   sexo      q1      q2      q3 frequencia  
##   <chr>    <dbl> <dbl> <dbl>      <int>  
## 1 feminino    22     23     25         29  
## 2 masculino    22     23     24         71
```

Boxplot

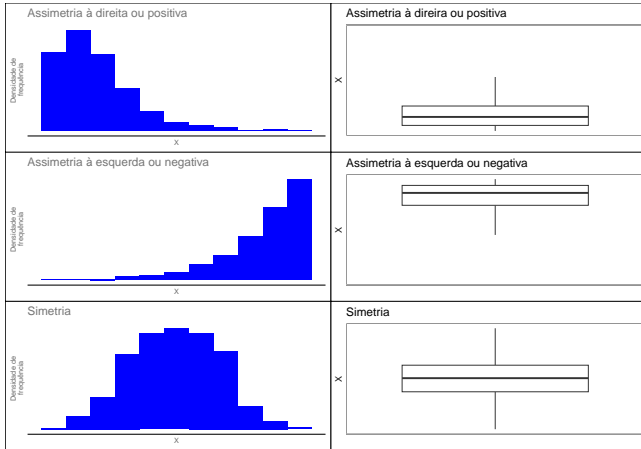
medida de dispersão: uma distância pequena entre q_1 e q_3 indica homogeneidade dos dados.

Diferença de quartis: $dq = q_3 - q_1$.



Boxplot

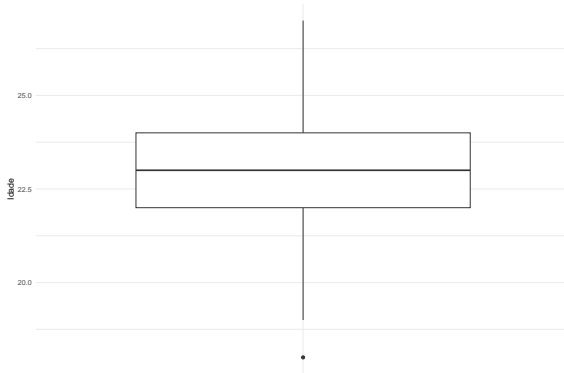
Assimetria:



Boxplot

```
ggplot(dados) +  
  geom_boxplot(aes(x = "", y = idade)) +  
  labs(x = "", y = "Idade") +  
  theme_minimal()
```


Boxplot



Boxplot

```
ggplot(dados) +  
  geom_boxplot(aes(x = sexo, y = idade,  
                   fill = sexo)) +  
  labs(x = "Sexo", y = "Idade") +  
  theme_minimal()
```

Boxplot

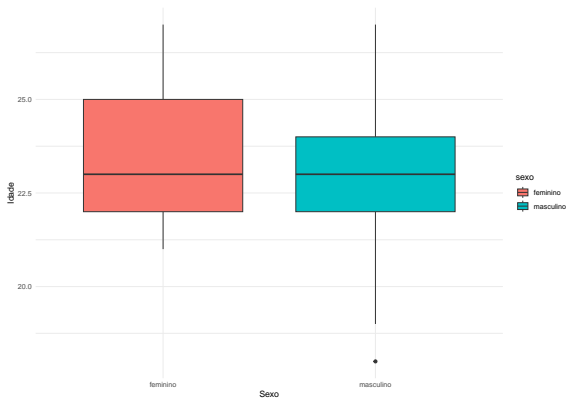


Gráfico temporal

Uma *série temporal* é uma sequência de medições da mesma variável coletada ao longo do tempo. Na maioria das vezes, as medições são feitas em intervalos de tempo regulares.

Uma característica importante de uma série temporal é que ela é uma sequência de observações onde a ordenação é importante. A ordenação é muito importante porque há dependência e alterar a ordem pode alterar o significado dos dados.

Uma série temporal pode ser representada em um gráfico de dispersão onde o *tempo* está no eixo x e as medições da variável de interesse estão no eixo y. Usualmente, os pontos sucessivos são unidos por retas.

Gráfico temporal

```
dados <- read_xlsx("../dados/brutos/economia_us.xlsx")  
glimpse(dados)
```

```
## Rows: 574  
## Columns: 6  
## $ date      <chr> "1/7/1967", "1/8/1967", "1/9/1967",  
## $ pce       <dbl> 506.7, 509.8, 515.6, 512.2, 517.4, 5  
## $ pop       <dbl> 198712, 198911, 199113, 199311, 1994  
## $ psavert   <dbl> 12.6, 12.6, 11.9, 12.9, 12.8, 11.8,  
## $ uempmed   <dbl> 4.5, 4.7, 4.6, 4.9, 4.7, 4.8, 5.1, 4  
## $ unemploy  <dbl> 2944, 2945, 2958, 3143, 3066, 3018,
```

Gráfico temporal

```
library(lubridate)

dados_econ <- dados |>
  mutate(date = dmy(date))
```

Gráfico temporal

```
ggplot(dados_econ, aes(x = date, y = psavert)) +  
  geom_line() +  
  scale_x_date(date_labels = "%m-%Y") +  
  xlab("")
```

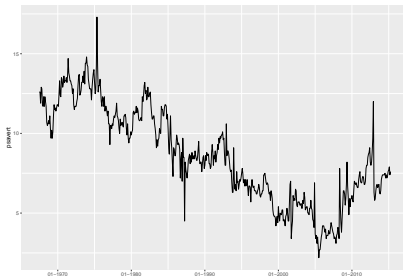


Gráfico temporal

```
library(plotly)

p <- ggplot(dados_econ, aes(x = date, y = psavert))
  geom_line() +
  scale_x_date(date_labels = "%m-%Y") +
  xlab("")

ggplotly(p)
```


Gráfico de dispersão

Um *gráfico de dispersão* é uma representação gráfica de duas variáveis quantitativas onde a variável explicativa está no eixo x e a variável resposta está no eixo y e cada par de valores (x, y) é representado por um ponto. Ao analisar um gráfico de dispersão, buscamos responder as seguintes questões:

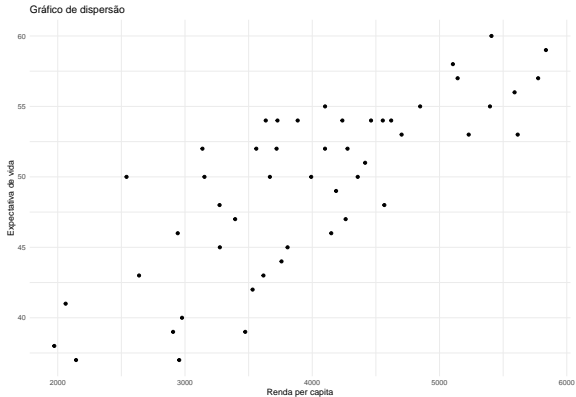
- Qual é a direção da relação?
- A relação é linear ou não linear?
- A relação é fraca, moderada ou forte?
- Existem valores atípicos ou extremos?

Gráfico de dispersão

```
dados <- read_xlsx("../dados/brutos/renda_vida.xlsx")

ggplot(dados) +
  geom_point(aes(x = renda, y = vida)) +
  labs(x = "Renda per capita",
       y = "Expectativa de vida",
       title = "Gráfico de dispersão") +
  theme_minimal()
```

Gráfico de dispersão



Coeficiente de correlação linear de Pearson

O *coeficiente de correlação linear de Pearson* é uma medida numérica da força de associação linear entre duas variáveis quantitativas.

Sejam x_1, x_2, \dots, x_n n valores observados da variável aleatória quantitativa X e sejam y_1, y_2, \dots, y_n n valores observados da variável aleatória quantitativa Y . A correlação amostral, r , entre X e Y é definida por

$$r = \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2}}.$$

Coeficiente de correlação linear de Pearson

- O coeficiente de correlação linear é adimensional e é tal que $-1 \leq r \leq 1$.
- Se $r > 0$, temos que as duas variáveis possuem uma relação linear positiva.
- Se $r < 0$, temos que as duas variáveis possuem uma relação linear negativa.
- Quando $r = 0$, temos uma ausência de relação linear entre as duas variáveis.

```
cor(dados$renda, dados$vida)
```

```
## [1] 0.7843596
```

Introdução à Regressão Linear Simples

Problema de Análise de Regressão: estabelecer e determinar uma função que descreva a relação entre uma variável, chamada de variável resposta e denotada por Y , e um conjunto de variáveis observáveis, chamadas de variáveis preditoras, explicativas ou covariáveis e denotadas por X_1, X_2, \dots, X_p .

Uma vez estabelecida e determinada a relação funcional entre a variável resposta e as covariáveis, a Análise de Regressão pode explorar esta relação para obter informações sobre Y a partir do conhecimento de X_1, X_2, \dots, X_p . Os modelos de regressão podem, então, serem usados para predição, estimação, testes de hipótese e para modelar relações casuais.

Introdução à Regressão Linear Simples

Modelo de Regressão Linear Simples:

Seja:

- Y a variável resposta;
- y_1, y_2, \dots, y_n , n valores observados da variável resposta Y ;
- X a variável preditora;
- x_1, x_2, \dots, x_n , n valores observados da variável preditora.

As observações (x_i, y_i) , $i = 1, 2, \dots, n$ são pares de valores observados de (X, Y)

Introdução à Regressão Linear Simples

É muito pouco provável que as coordenadas $(x_1, y_1), \dots, (x_n, y_n)$ forneçam exatamente uma linha reta: haverá algum erro que deve ser considerado na construção do modelo. Assim, o modelo de regressão linear simples é descrito por

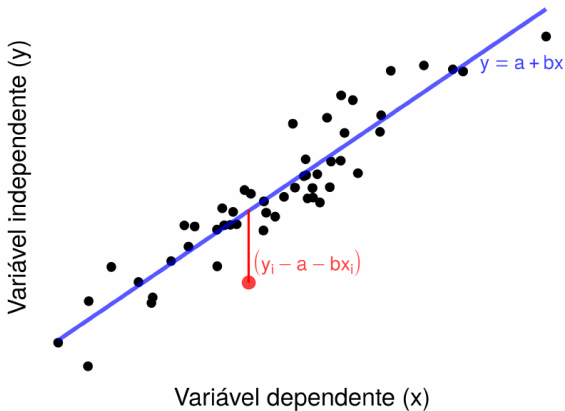
$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad (1)$$

onde β_0 é o intercepto, β_1 é o parâmetro de inclinação e ϵ_i é o erro aleatório do valor de y_i com relação à reta $\beta_0 + \beta_1 x_i$, para $i = 1, 2, \dots, n$.

β_0 e β_1 são parâmetros (populacionais) desconhecidos que devem ser estimados utilizando os métodos de estimação de Inferência Estatística.

Modelo de Regressão Linear Simples

Ilustração dos erros em regressão linear simples:



Introdução à Regressão Linear Simples

Suposições do modelo de regressão linear simples:

No modelo de regressão linear simples usual, os ϵ_i 's são variáveis aleatórias sujeitas às seguintes condições:

- O valor esperado de cada erro é zero: $E(\epsilon_i) = 0, i = 1, \dots, n.$
- Os erros têm a mesma variância: $Var(\epsilon_i) = \sigma^2, i = 1, \dots, n.$
- Os erros são não correlacionados:
 $Cov(\epsilon_i, \epsilon_j) = 0, i \neq j, i, j = 1, 2, \dots, n.$

Introdução à Regressão Linear Simples

De uma maneira mais simples, podemos enunciar as suposições do modelo de regressão linear simples como segue:

- Linearidade: a variável resposta Y tem uma relação (aproximadamente) linear com a variável preditora X .
- Homoscedasticidade: para cada valor de X , a distribuição dos erros tem a mesma variância. Isso significa dizer que o nível de erro no modelo é aproximadamente o mesmo independente do valor da variável preditora.
- Independência dos erros: os erros não devem ser correlacionados. Idealmente, não deve ocorrer nenhum padrão entre resíduos consecutivos.

Por último, fazemos uma suposição extra:

- Normalidade: os erros do modelo são normalmente distribuídos.

Introdução à Regressão Linear Simples

Estimação:

Os parâmetros β_0 e β_1 são desconhecidos e devem ser estimados utilizando os dados amostrais observados.

O método de mínimos quadrados (MMQ) é mais utilizado do que qualquer outro procedimento de estimação em modelos de regressão e fornece os estimadores de β_0 e β_1 tal que a soma de quadrados das diferenças entre as observações y_i 's e a linha reta ajustada seja mínima.

Assim, de todos os possíveis valores de β_0 e β_1 , os estimadores de mínimos quadrados (EMQ) serão aqueles que minimizam a soma de quadrados dos erros, que é dada por

$$\sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2. \quad (2)$$

Introdução à Regressão Linear Simples

Usando o MMQ, as estimativas de β_0 e β_1 são, respectivamente

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (3)$$

e

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n y_i x_i - n \bar{y} \bar{x}}{\sum_{i=1}^n x_i^2 - n \bar{x}^2}, \quad (4)$$

onde $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ e $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ são as médias amostrais dos y_i 's e x_i 's, respectivamente.

Introdução à Regressão Linear Simples

Reta ajustada (modelo ajustado):

Uma vez as estimativas de β_0 e β_1 tenham sido obtidas, teremos a reta de regressão linear ajustada.

O modelo de regressão linear simples ajustado é

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i, \quad i = 1, \dots, n, \quad (5)$$

que é a estimativa pontual da média de Y_i para um particular x_i .

Introdução à Regressão Linear Simples

Interpretação dos parâmetros:

A inclinação de uma reta é a mudança na variável y sobre a mudança na variável x . Se a mudança na variável x é um, então a inclinação é interpretada como a mudança em y para um incremento de uma unidade em x . Essa mesma interpretação pode ser aplicada ao parâmetro de inclinação da reta de regressão linear simples ajustada. Assim, temos que:

- $\hat{\beta}_1$ representa o aumento estimado em Y para cada aumento de uma unidade em X . Se o valor de $\hat{\beta}_1$ é negativo, então temos um incremento negativo.
- $\hat{\beta}_0$ é o intercepto da linha de regressão com o eixo- y . Então, quando $X = 0$ é um valor que faz sentido para os dados estudados, $\hat{\beta}_0$ é a estimativa do valor de Y quando $X = 0$.