Tópico 3: Manipulação e Visualização de Dados com R

MRS

3. Manipulação e Visualização de Dados

- Após importar os dados, o próximo passo é manipulá-los e visualizá-los para gerar insights.
- Pacotes úteis dentro de **tidyverse**:
  - `dplyr`: para transformação dos dados.
  - `ggplot2`: para visualização gráfica.

Importação de dados

- É possível importar dados de diversos formatos.
- Formatos comuns: `.txt`, `.csv`, `.xlsx`, `.json`, `.xml`.

Exemplo: Importar de TXT

```
# Base R
dados_txt_base <- read.table("_dados/dados.txt", header = TRUE, sep = "\t")
head(dados_txt_base)
```

```
    nome idade sexo
1    Ana    23    F
2  Bruno    30    M
3  Carla    27    F
4 Daniel    35    M
5    Eva    29    F
```

```r
# readr (tidyverse)
library(readr)
dados_txt_readr <- read_tsv("_dados/dados.txt", show_col_types = FALSE)
head(dados_txt_readr)
```

```
# A tibble: 5 x 3
  nome   idade sexo
  <chr>  <dbl> <chr>
1 Ana       23 F
2 Bruno     30 M
3 Carla     27 F
4 Daniel    35 M
5 Eva       29 F
```

Exemplo: Importar de CSV

```r
# Base R
dados_csv_base <- read.csv("_dados/dados.csv", header = TRUE)
head(dados_csv_base)
```

```
    nome idade sexo
1    Ana    23    F
2  Bruno    30    M
3  Carla    27    F
4 Daniel    35    M
5    Eva    29    F
```

```r
# readr (tidyverse)
library(readr)
dados_csv_readr <- read_csv("_dados/dados.csv", show_col_types = FALSE)     # read_csv2() co
head(dados_csv_readr)
```

```
# A tibble: 5 x 3
  nome   idade sexo
  <chr>  <dbl> <chr>
1 Ana       23 F
2 Bruno     30 M
3 Carla     27 F
4 Daniel    35 M
5 Eva       29 F
```

Exemplo: Importar de Excel

```
# readxl (tidyverse)
library(readxl)

dados_excel_base <- read_excel("_dados/dados.xlsx")   # Lê a primeira folha do ficheiro

head(dados_excel_base)
```

```
# A tibble: 5 x 3
  nome   idade sexo
  <chr>  <dbl> <chr>
1 Ana       23 F
2 Bruno     30 M
3 Carla     27 F
4 Daniel    35 M
5 Eva       29 F
```

Exemplo: Importar de JSON (NDJSON - JSON por linha)

```
# rsjon
# install.packages("rjson")
library(rjson)
dados_json_rjson <- fromJSON(file = '_dados/dados.json')

dados_df = as.data.frame(do.call(rbind, dados_json_rjson))  # para apresentar como tabela
head(dados_df)
```

```
    nome idade sexo
1    Ana    23    F
2  Bruno    30    M
3  Carla    27    F
4 Daniel    35    M
5    Eva    29    F
```

Exemplo: Importar de XML

```r
# xml2 + dplyr (tidyverse)
library(xml2)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
# Ler e transformar em lista
dados_xml_raw <- read_xml("_dados/dados.xml")
dados_xml_list <- as_list(dados_xml_raw)

# Extrair dados e transformar em tibble
# tibble --> data frame mais moderno, suportado pelo pacote tibble do tidyverse
dados_xml_xml2 <- lapply(dados_xml_list[[1]], function(x) {
  sapply(x, function(y) as.character(y))
}) %>% bind_rows()

head(dados_xml_xml2)
```

```
# A tibble: 5 x 3
  nome   idade sexo
  <chr>  <chr> <chr>
1 Ana    23    F
2 Bruno  30    M
3 Carla  27    F
4 Daniel 35    M
5 Eva    29    F
```

Limpeza e transformação com dplyr

- `dplyr` oferece funções para manipulação eficiente dos dados.

Principais funções

- `filter()`: filtra linhas
- `select()`: seleciona colunas
- `mutate()`: cria ou modifica colunas
- `arrange()`: ordena dados
- `group_by()` + `summarise()`: agrupa e resume

Exemplos

```
library(readr)

dados <- read_csv("_dados/dados.csv", show_col_types = FALSE)

dados
```

```
# A tibble: 5 x 3
  nome   idade sexo
  <chr>  <dbl> <chr>
1 Ana       23 F
2 Bruno     30 M
3 Carla     27 F
4 Daniel    35 M
5 Eva       29 F
```

```
library(dplyr)

dados_filtrados <- dados %>%
  filter(idade > 30) %>%
  select(nome, idade) %>%
  mutate(idade_2030 = idade + 5)

dados_filtrados
```

```
# A tibble: 1 x 3
  nome   idade idade_2030
  <chr>  <dbl>      <dbl>
1 Daniel    35         40
```

Agrupamento e resumo

```r
library(dplyr)

dados %>%
  group_by(sexo) %>%
  summarise(media_idade = mean(idade, na.rm = TRUE))  # ignora os NAs
```

```
# A tibble: 2 x 2
  sexo  media_idade
  <chr>       <dbl>
1 F            26.3
2 M            32.5
```

Visualização com ggplot2

- `ggplot2` permite criar gráficos personalizados com uma de camadas.

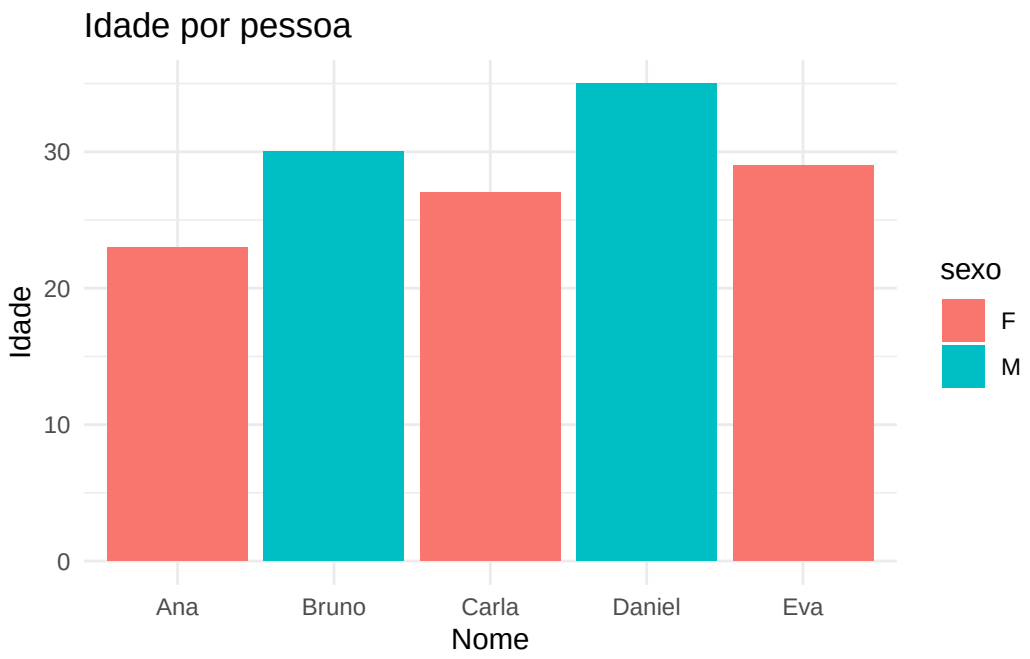
Estrutura básica

```r
# install.packages("ggplot2")
# library(ggplot2)

# ggplot(dados, aes(x = variavel_x, y = variavel_y)) + geom_<tipo_de_grafico>()
```

Exemplos

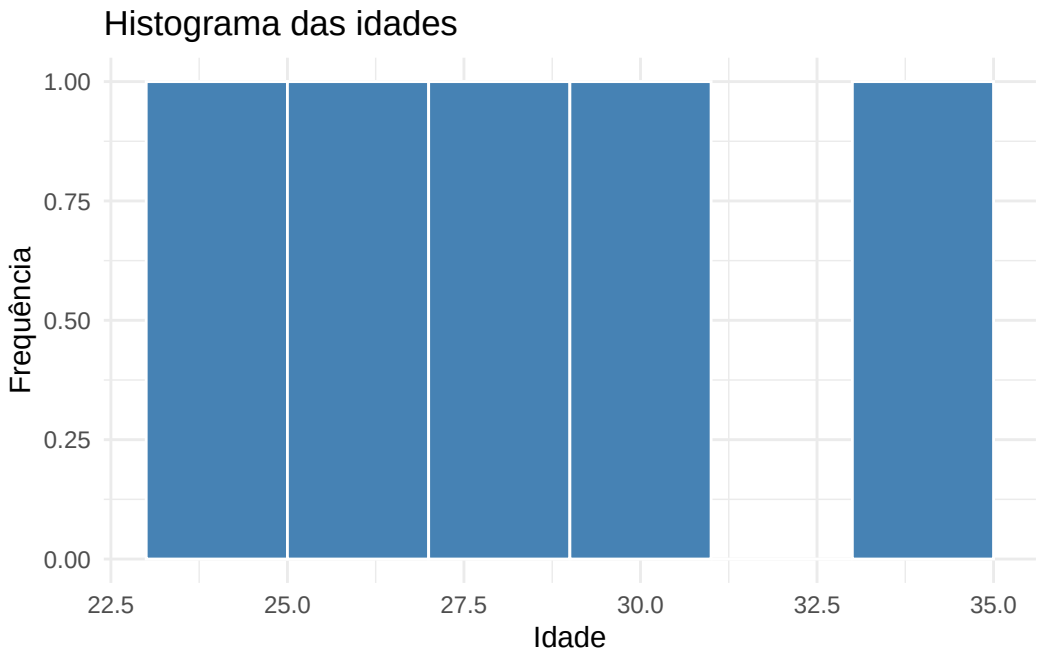
Gráfico de barras

```r
library(ggplot2)

# ggplot(dados, aes(x = sexo)) + geom_bar()

ggplot(dados, aes(x = nome, y = idade, fill = sexo)) +
  geom_bar(stat = "identity") +
  labs(title = "Idade por pessoa", x = "Nome", y = "Idade") +
  theme_minimal()
```



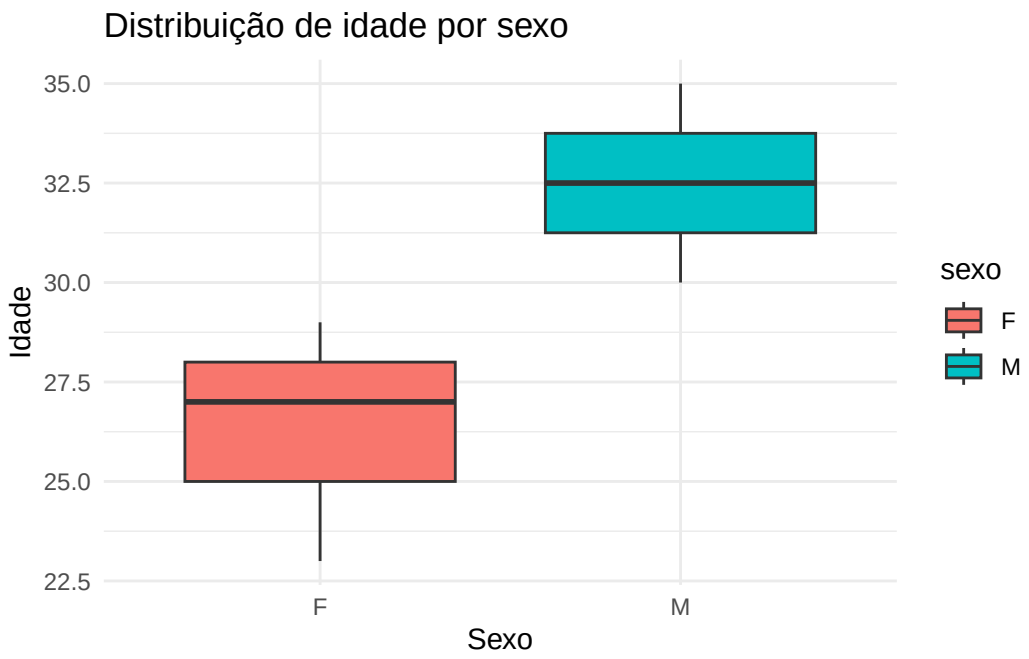
Histograma

```
library(ggplot2)

# ggplot(dados, aes(x = sexo)) + geom_histogram(binwidth = 5)

ggplot(dados, aes(x = idade)) +
  geom_histogram(binwidth = 2, fill = "steelblue", color = "white") +
  labs(title = "Histograma das idades", x = "Idade", y = "Frequência") +
  theme_minimal()
```



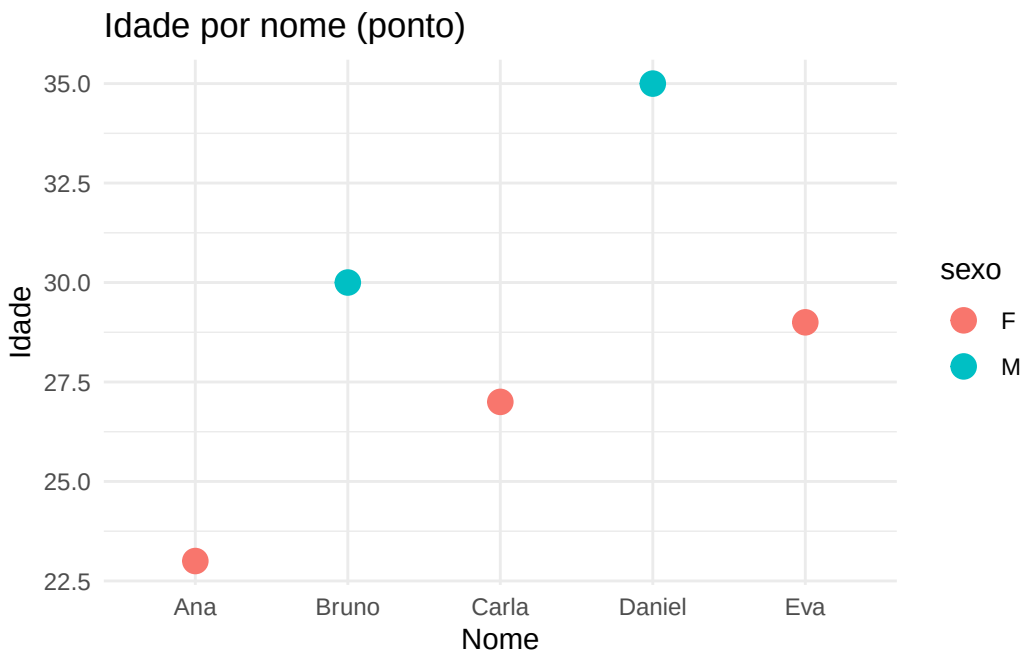
Boxplot

```
library(ggplot2)

# ggplot(dados, aes(x = genero, y = sexo)) + geom_boxplot()

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



Dispersão

```
# ggplot(dados, aes(x = altura, y = peso)) + geom_point()

ggplot(dados, aes(x = nome, y = idade, color = sexo)) +
  geom_point(size = 4) +
  labs(title = "Idade por nome (ponto)", x = "Nome", y = "Idade") +
  theme_minimal()
```



Exercício prático

1. Crie um dataset com colunas: `nome`, `idade`, `cidade`, `genero` e `salario`.
2. Filtre apenas as pessoas com salário acima de 2500.
3. Adicione uma coluna `salario_2026` com o crescimento de 25% do salário.
4. Calcule a média de salário por cidade.
5. Crie um gráfico de barras da contagem por cidade.
6. Crie um boxplot dos salários por género.

Solução

```
library(tidyverse)  # pacotes usados: tibble, dplyr, ggplot2
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.1
```

```
-- Conflicts --------------------------------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```r
# 1. Dataset pessoas
pessoas <- tibble(
  nome = c("Ana", "Bruno", "Carla", "Daniel", "Eva", "Felipe", "Gabriela", "Henrique", "Isabe
  idade = c(25, 32, 28, 40, 30, 27, 35, 45, 29, 33),
  cidade = c("Lisboa", "Porto", "Lisboa", "Coimbra", "Faro", "Lisboa", "Porto", "Coimbra", "F
  genero = c("F", "M", "F", "M", "F", "M", "F", "M", "F", "M"),
  salario = c(2100, 3200, 2700, 4000, 2300, 2600, 3100, 2900, 2200, 3500)
)

# 2. Filtrar salários > 2500
pessoas %>%
  filter(salario > 2500) %>%
  select(nome, idade, cidade, salario)
```

```
# A tibble: 7 x 4
  nome     idade cidade  salario
  <chr>    <dbl> <chr>     <dbl>
1 Bruno       32 Porto      3200
2 Carla       28 Lisboa     2700
3 Daniel      40 Coimbra    4000
4 Felipe      27 Lisboa     2600
5 Gabriela    35 Porto      3100
6 Henrique    45 Coimbra    2900
7 João        33 Lisboa     3500
```

```r
# 3. Nova coluna com o crescimento de 25% do salário
pessoas %>%
  mutate(salario_2026 = salario * 1.25)
```

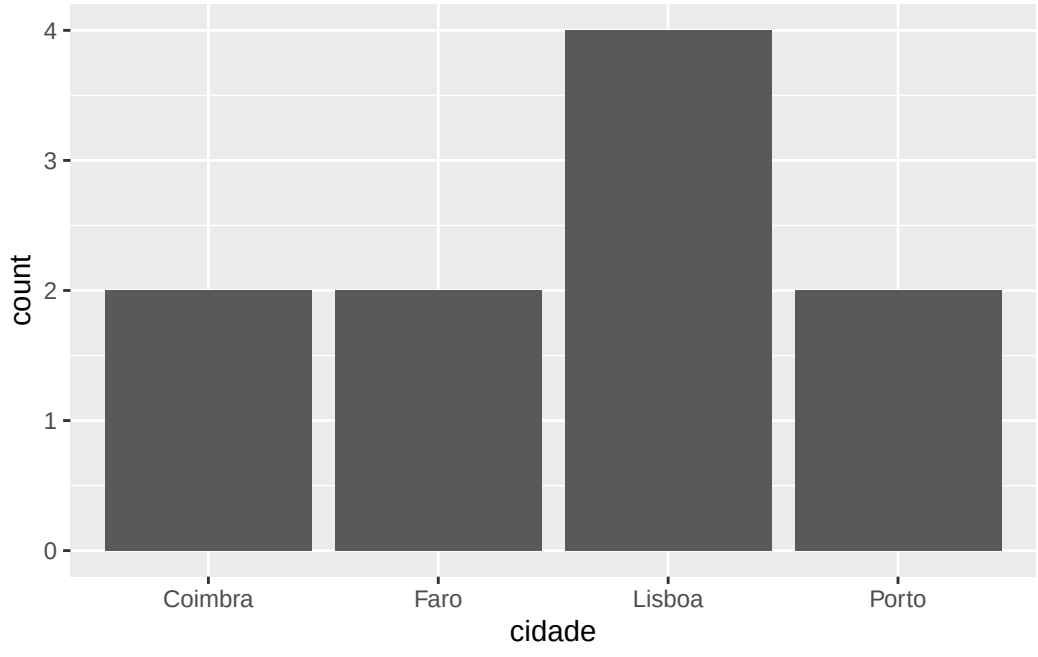
```
# A tibble: 10 x 6
   nome     idade cidade  genero salario salario_2026
   <chr>    <dbl> <chr>   <chr>    <dbl>        <dbl>
 1 Ana         25 Lisboa  F         2100         2625
 2 Bruno       32 Porto   M         3200         4000
 3 Carla       28 Lisboa  F         2700         3375
 4 Daniel      40 Coimbra M         4000         5000
```

```
 5 Eva         30 Faro    F          2300          2875
 6 Felipe      27 Lisboa  M          2600          3250
 7 Gabriela    35 Porto   F          3100          3875
 8 Henrique    45 Coimbra M          2900          3625
 9 Isabela     29 Faro    F          2200          2750
10 João        33 Lisboa  M          3500          4375
```

```r
# 4. Média de salário por cidade
pessoas %>%
  group_by(cidade) %>%
  summarise(media_salario = mean(salario, na.rm = TRUE))
```

```
# A tibble: 4 x 2
  cidade  media_salario
  <chr>           <dbl>
1 Coimbra          3450
2 Faro             2250
3 Lisboa           2725
4 Porto            3150
```

```r
# 5. Gráfico de barras da contagem por cidade
ggplot(pessoas, aes(x = cidade)) + geom_bar()
```



```r
# 6. Boxplot de salários por género
ggplot(pessoas, aes(x = genero, y = salario)) + geom_boxplot()
```