



Manipulação de Dados com R

Domine as técnicas fundamentais para criar, combinar e transformar dados usando R e dplyr



Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
<https://eic.cefet-rj.br/~eogasawara>

PASSO 1

Criando uma Tabela Básica

O primeiro passo é criar um data frame com informações básicas. Utilizamos vetores com a função `c()` para definir colunas de peso, altura e identificação dos sujeitos.

A função `data.frame()` combina esses vetores em uma estrutura tabular que podemos manipular facilmente.

```
weight <- c(60, 72, 57, 90, 95, 72)
height <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
subject <- c("A", "B", "C", "D", "E", "F")

d <- data.frame(
  weight=weight,
  height=height,
  subject=subject
)

head(d)
```

PASSO 2

Criando uma Tabela Auxiliar



Tabela Principal

Contém dados físicos (peso e altura) dos sujeitos A-F



Tabela Auxiliar

Adiciona informações geográficas (estado) para cada sujeito

Criamos uma segunda tabela com dados complementares, associando cada sujeito ao seu estado de origem (RJ, SP ou MG).

```
subject <- c("A", "B", "C", "D", "E", "F")
state <- c("RJ", "SP", "MG", "RJ", "SP", "MG")
```

```
ds <- data.frame(
  subject=subject,
  state=state
)
```

PASSO 3

Integração de Dados por Junção

A função `merge()` combina as duas tabelas usando a coluna **subject** como chave de ligação. Isso permite integrar informações de diferentes fontes em uma única tabela consolidada.



Tabela d

Dados físicos



Tabela ds

Dados geográficos



Tabela dsm

Dados completos

```
dsm <- merge(d, ds, by.x="subject", by.y="subject")
head(dsm)
## subject weight height state
## 1 A 60 1.75 RJ
## 2 B 72 1.80 SP
```

O Poder das Transformações

01

Dados de Entrada

Conjunto de dados inicial (dsm)

03

Função de Transformação

Processa os dados (filter, select, arrange)

02

Operador Pipeline

O símbolo |> passa dados implicitamente

04

Resultado

Dados transformados prontos para análise

- ❑ **Importante:** O operador pipeline |> permite encadear múltiplas operações de forma clara e legível, passando o resultado de cada função automaticamente para a próxima.

Filtragem, Projeção e Ordenação

filter()

Seleciona apenas registros onde altura > 1.7m

select()

Escolhe colunas específicas para o resultado

arrange()

Ordena os dados por altura crescente

O pacote `dplyr` oferece funções poderosas para manipular dados de forma intuitiva e eficiente.

```
library(dplyr)

result <- dsm |>
  filter(height > 1.7) |>
  select(subject, weight, height) |>
  arrange(height)

## subject weight height
## 1 E 95 1.74
## 2 A 60 1.75
```

Resumindo Dados por Grupos

A combinação de `group_by()` e `summarize()` permite calcular estatísticas agregadas para cada categoria. Neste exemplo, calculamos a contagem e média de altura por estado.

3

Estados

MG, RJ e SP analisados

2

Sujeitos/Estado

Distribuição equilibrada

1.79m

Altura Média

Média geral dos grupos

```
result <- dsm |>  
group_by(state) |>  
summarize(  
  count = n(),  
  height = mean(height)  
)
```

```
## # A tibble: 3 × 3  
## state count height  
## 1 MG     2     1.78  
## 2 RJ     2     1.82  
## 3 SP     2     1.77
```

Referências



Hands-On
Programming
with R

WRITE YOUR OWN FUNCTIONS AND SIMULATIONS

Hands-on Programming

Aprenda R criando suas
próprias funções e simulações

[https://rstudio-
education.github.io/hopr/index.html](https://rstudio-education.github.io/hopr/index.html)

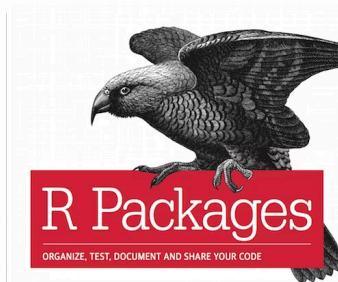


R Graphics
Cookbook

R Graphics Cookbook

Domine visualizações de dados
em R

<https://r-graphics.org>



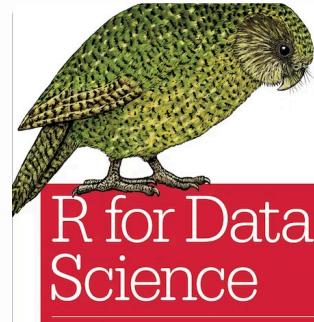
R Packages

ORGANIZE, TEST, DOCUMENT AND SHARE YOUR CODE

R Packages

Desenvolva seus próprios
pacotes R

<https://r-pkgs.org/index.html>



R for Data
Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA

R for Data Science

Guia completo para ciência de
dados

<https://r4ds.had.co.nz>