

# Fatores em R

Dominando variáveis categóricas para análise de dados eficiente



Eduardo Ogasawara

[eduardo.ogasawara@cefet-rj.br](mailto:eduardo.ogasawara@cefet-rj.br)

<https://eic.cefet-rj.br/~eogasawara>

## O que são Fatores?

### Dados Categóricos

Fatores são variáveis em R especialmente projetadas para representar dados categóricos, como categorias, grupos ou classificações.

### Estrutura Interna

Armazenados internamente como vetores de valores inteiros, com rótulos de caracteres associados que aparecem quando o fator é exibido.

### Transformação de Dados

Variáveis numéricas e de caracteres podem ser facilmente transformadas em fatores, mas os níveis sempre serão valores de caracteres.

### Não Ordenados

Categorias sem hierarquia (ex: cores, tipos)

### Ordenados

Categorias com ordem lógica (ex: baixo, médio, alto)

## Criando Fatores Ordenados

Veja como transformar um vetor numérico em um fator ordenado que representa níveis de dor:

```
pain <- c(0,3,2,2,1)
fpain <- factor(pain, levels=0:3, ordered=TRUE)
fpain
```

### Resultado

```
[1] 0 3 2 2 1
Levels: 0 < 1 < 2 < 3
```

O parâmetro `ordered=TRUE` indica que existe uma hierarquia entre os níveis, onde 0 é menor que 1, que é menor que 2, e assim por diante.

## Atribuindo Rótulos aos Níveis

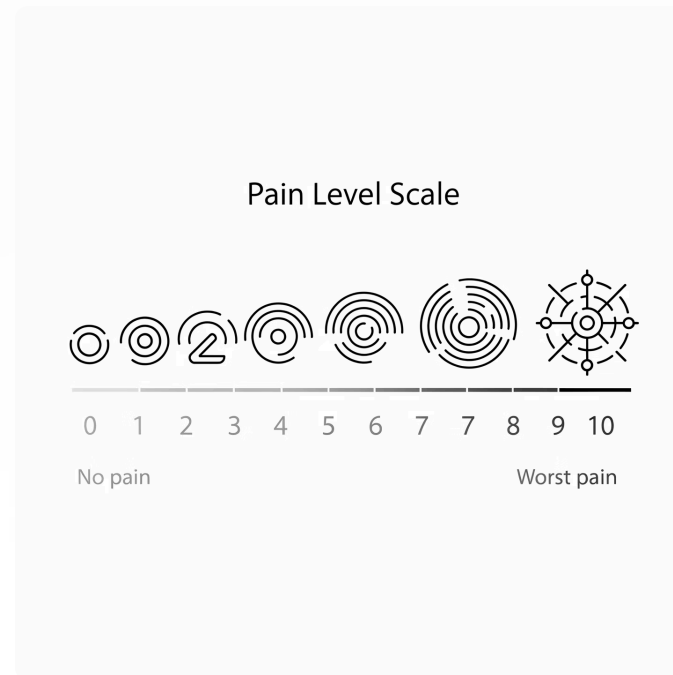
Após criar um fator, você pode substituir os números por rótulos descritivos mais intuitivos usando a função `levels()`:

```
levels(fpain) <- c("sem", "baixa", "média", "alta")  
fpain
```



### Saída

```
[1] sem  alta  média média baixa  
Levels: sem < baixa < média < alta
```



**Exemplo prático:** Fator ordenado indicando níveis de dor para pacientes em um estudo clínico.



## Exemplo Prático: Classificação de Altura

Vamos categorizar pessoas por altura usando fatores ordenados:

01

### Menos de 1,5m

Classificadas como **pequenas**

02

### Entre 1,5m e 1,9m

Classificadas como **médias**

03

### Mais de 1,9m

Classificadas como **altas**

## Dados de Exemplo

```
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")
```

## Conversão Usando Loop

Uma abordagem intuitiva usando estruturas de repetição e condicionais para transformar um vetor numérico em fator:

```
lev <- rep("", length(height))
for (i in 1:length(height)) {
  if (height[i] < 1.7) lev[i] <- "baixa"
  else if (height[i] < 1.9) lev[i] <- "média"
  else lev[i] <- "alta"
}
lev <- factor(lev, levels=c("baixa", "média", "alta"), ordered=TRUE)
lev
```

### Resultado

```
[1] média média baixa alta média alta
Levels: baixa < média < alta
```

Este método oferece controle total sobre as condições de classificação, sendo ideal para lógicas personalizadas complexas.

## Conversão com a Função cut()

A função `cut()` foi especialmente projetada para converter variáveis contínuas em fatores categóricos através de pontos de corte:

```
lev <- cut(height, breaks=c(0, 1.7, 1.9, .Machine$double.xmax),
           ordered=TRUE)
lev
```



```
[1] (1.7,1.9]  (1.7,1.9]  (0,1.7]
(1.7,1.9]  (1.7,1.9]  (1.9,1.8e+308]
Levels: (0,1.7] < (1.7,1.9] < (1.9,1.8e+308]
```

## Personalizando os Rótulos

```
levels(lev) <- c("baixa", "média", "alta")
lev
```



```
[1] média média baixa média média alta
Levels: baixa < média < alta
```

### Vantagem Principal

Solução mais elegante e eficiente, especialmente adequada para conversões baseadas em intervalos numéricos.

### Flexibilidade

Permite definir múltiplos pontos de corte de forma simples e direta.

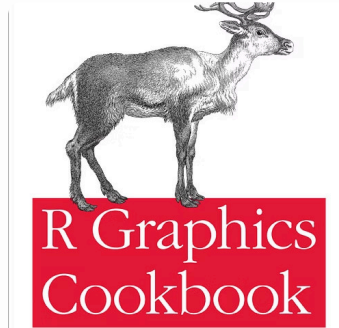
# Referências



## Hands-on Programming

Aprenda R criando suas próprias funções e simulações

<https://rstudio-education.github.io/hopr/index.html>



## R Graphics Cookbook

Domine visualizações de dados em R

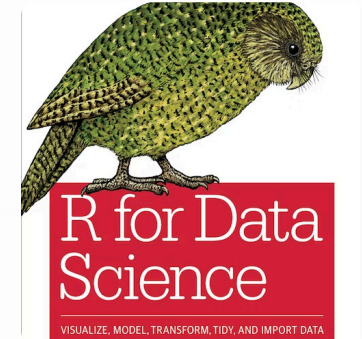
<https://r-graphics.org>



## R Packages

Desenvolva seus próprios pacotes R

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



## R for Data Science

Guia completo para ciência de dados

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