



1



2

Fatores

- Usados para representar dados categóricos
 - Vetores de inteiros com rótulos associados
 - Depois de criado só pode ter a faixa de valores definidos em levels
- Pode ser criado com a função `factor()`

```
> fumante <- factor(c("sim", "não", "sim", "sim", "não", "não", "não"))
> fumante
[1] sim não sim sim não não não
Levels: não sim
```

- Pode ser convertido de um vetor de strings

```
> fumante <- c("sim", "não", "sim", "sim", "não", "não", "não")
> fator <- as.factor(fumante)
> fator
[1] sim não sim sim não não não
Levels: não sim
```

Prof. Dr. Razer A N R Montão

SEPT/UFR

3

3

Fatores

- Levels são os valores dos rótulos do fator

```
> levels(fumante)
[1] "não" "sim"
> nlevels(fumante)
[1] 2
```

- Podem ser convertidos para numérico

```
> as.numeric(fumante)
[1] 2 1 2 2 1 1 1
```

Prof. Dr. Razer A N R Montão

SEPT/UFR

4

4

Fatores.

- Um dos usos
 - Carga de bases de dados (CSV) para memória
 - Ex.: função `read.csv()`
 - Quando carrega colunas *string*, por default transforma em *factor* (categórico)
 - Se for uma coluna que não é categórico, como um nome, ou uma descrição, usa-se
 - `stringAsFactors=FALSE`
 - Ex.:

```
arq <- read.csv(file = 'dados.csv', stringsAsFactors = FALSE)
```

Fórmulas

- São objetos do tipo $y \sim x$
 - Lê-se *y depende de x*
 - Representa uma expressão não avaliada
 - y é a variável dependente
 - x é a variável independente
- São associações entre objetos, como em um modelo de regressão

```
> formula <- y ~ x1 + x2
```

```
> class(formula)
```

```
[1] "formula"
```

- Significa que y é em função de $x1 + x2$
- Serão usadas em exemplos posteriores para **regressão**

Fórmulas.

- Fórmulas indicam dependência entre variáveis
 - Símbolos * e ^ possuem significado diferente
 - Se quiser expressar operações, use I ()
- Exemplo

```
> formula <- y ~ x1 + I(2*x2) + I(x3^3)
```