Introdução ao uso do

em Ciências da Saúde

3. Leitura e manejo de dados

Fúlvio Borges Nedel

Departamento de Saúde Pública – SPB
Centro de Ciências da Saúde – CCS
Universidade Federal de Santa Catarina – UFSC

Grups de Recerca d'Amèrica i Àfrica Llatines – GRAAL
http://graal.uab.cat

4 de dezembro de 2017

Fúlvio Nedel
SPB/UFSC

Parte I

O estudo e a leitura dos dados

Roteiro

Introdução
Um estudo transversal

Fúlvio Nedel
SPB/UFSC

- Em 2011 o Serviço de Saúde Comunitária do Grupo Hospitalar Conceição (SSC/GHC), em Porto Alegre, RS, iniciou um estudo sobre o processo saúde-doença-atenção de pessoas com Hipertensão (HAS) ou Dibetes Mellitus (DM) usuárias do Serviço
- No estudo, foi realizado um inquérito sobre uma amostra dos usuários, em que se perguntou o peso e altura do indivíduo.
- Usaremos um extrato dessa base de dados, que pode ser baixado nesse link.[1] Baixe o arquivo de dados e salve-o em um diretório para este curso, no seu computador
- Faremos uma análise exploratória do estado nutricional.

[1] Usaremos 300 registros, selecionados por conveniência. Assim, os resultados aqui encontrados não se aplicam nem à amostra nem à população de onde ela foi coletada. Entretanto, nestes exercícios, nossa base de dados será tratada como uma amostra aleatória da população-alvo. O nº de registro do usuário foi modificado, por questões éticas.

Objetivos da análise

Fúlvio Nedel
SPB/UFSC

1. Descrever o Índice de Massa Corporal (IMC) e analisar fatores associados à sua média.
2. Descrever a frequência de categorias do estado nutricional e analisar fatores possivelmente associados à obesidade:
    1. sexo
    2. idade
    3. condição socioeconômica
    4. participação em grupos de promoção da saúde

As variáveis de análise
Variável dependente

Fúlvio Nedel
SPB/UFSC

- O IMC é calculado como a razão entre o peso em quilos e o quadrado da altura em metros: $IMC = \frac{Kg}{m^2}$
- A obesidade é definida como um $IMC \geqslant 30 Kg/m^2$

```
As perguntas da entrevista
                    peso
"u47. Qual o seu peso?"
                    altura
"u48. Qual a sua altura?"
```

As variáveis de análise
Variáveis independentes

Fúlvio Nedel
SPB/UFSC

Tabela: Variáveis independentes, nome e rótulo.

Nome	Rótulo
sexo	u8. Sexo:
dataentr	u5. Data da entrevista:
datanasc	u7. Qual é a sua data de nascimento?
abepcls	Classificação socioeconômica ABEP modificada
grupohas	u53. Desde <6 MESES ATRÁS> o(a) Sr.(a) participou de algum grupo de hipertensos no <UNIDADE DE SAÚDE DE REFERÊNCIA>?
grupodm	u63. Desde <6 MESES ATRÁS> o(a) Sr.(a) participou de algum grupo de diabéticos no <UNIDADE DE SAÚDE DE REFERÊNCIA>?

Roteiro

Fúlvio Nedel
SPB/UFSC

Fúlvio Nedel
SPB/UFSC

A partir de agora iniciamos a análise de um banco de dados. **Abra o arquivo de sintaxe criado anteriormente.**(exercício do módulo 2)

Crie uma nova linha com um comentário explicando os passos que se seguirão e execute a partir dali os comandos. Algo como:

```
# Leitura e manejo de dados
# -------------------------
...
```

Leitura do arquivo e seleção das variáveis de interesse

- O arquivo de dados foi gerado no $P$$, e está em formato `.sav`
- O pacote `Hmisc` tem funções para facilitar a leitura de arquivos em diferentes formatos, inclusive SAV. Ative o pacote (com `library(Hmisc)`) e leia o arquivo `usuariosCursoR.sav` com a função `spss.get`
- Indique as variáveis de data: `datevars = "..."`
- Lembre-se de destinar a ação a um objeto: `nome do objeto <- spss.get(...)`

```
library(Hmisc)
cursoR <- spss.get( file = "usuariosCursoR.sav",
                    datevars = c("dataentr", "datanasc") )
```

Ignore os avisos. Eles poderiam ser evitados com o argumento `use.value.labels=FALSE`, mas aí... (`?spss.get` para ver o que aconteceria)

Leitura do arquivo e seleção das variáveis de interesse

- O arquivo de dados foi gerado no SPSS, e está em formato `.sav`
- O pacote `Hmisc` tem funções para facilitar a leitura de arquivos em diferentes formatos, inclusive SAV. Ative o pacote (com `library(Hmisc)`) e leia o arquivo `usuariosCursoR.sav` com a função `spss.get`
- Indique as variáveis de data: `datevars = "..."`
- Lembre-se de destinar a ação a um objeto: `nome do objeto <- spss.get(...)`

```
library(Hmisc)
cursoR <- spss.get( file = "usuariosCursoR.sav",
                    datevars = c("dataentr", "datanasc") )
```

Ignore os avisos. Eles poderiam ser evitados com o argumento `use.value.labels=FALSE`, mas aí... (`?spss.get` para ver o que aconteceria)

Leitura do arquivo e seleção das variáveis de interesse

Confirme que o objeto (cursoR) está presente no espaço de trabalho:

```
ls()
[1] "cursoR"
```

Leitura do arquivo e seleção das variáveis de interesse

Confirme que o objeto (`cursoR`) está presente no espaço de trabalho:

```
ls()
[1] "cursoR"
```

Verifique a `classe` do objeto:

```
class(cursoR)
[1] "data.frame"
```

Leitura do arquivo e seleção das variáveis de interesse

Confirme que o objeto (cursoR) está presente no espaço de trabalho:

```
ls()
[1] "cursoR"
```

Verifique a classe do objeto:

```
class(cursoR)
[1] "data.frame"
```

Verifique o nº de registros no banco de dados:

```
nrow(cursoR) # nº de linhas numa matriz ou banco de dados
[1] 300
```

Leitura do arquivo e seleção das variáveis de interesse

Confirme que o objeto (`cursoR`) está presente no espaço de trabalho:

```
ls()
[1] "cursoR"
```

Verifique a `classe` do objeto:

```
class(cursoR)
[1] "data.frame"
```

Verifique o nº de registros no banco de dados:

```
nrow(cursoR) # nº de linhas numa matriz ou banco de dados
[1] 300
```

Verifique o nº de variáveis no banco de dados:

```
ncol(cursoR) # nº de colunas numa matriz ou banco de dados
[1] 169
```

Leitura do arquivo e seleção das variáveis de interesse

Use a função `names()` para listar os nomes das variáveis:

```
names(cursoR)
```

```
  [1] "nquest"     "dataentr"   "u6"         "datanasc"   "sexo"       "u9"
  [7] "u10"        "u10.1"      "u11"        "u12"        "u13"        "u14"
 [13] "u15"        "u16"        "u17"        "u18"        "u19"        "u20"
 [19] "u21"        "u22"        "u23"        "u24.1"      "u24.2"      "u24.3"
 [25] "u24.4"      "u25"        "u26"        "u27"        "u28"        "u29"
 [31] "u30"        "u31"        "u32"        "u33"        "u34.1"      "u34.2"
 [37] "u34.3"      "u34.4"      "u34.5"      "u34.6"      "u35"        "u36"
 [43] "u36.1"      "u37"        "u38"        "u39"        "u40"        "u41"
 [49] "u42"        "u43"        "u44"        "u45"        "u46"        "peso"
 [55] "altura"     "u49"        "u50"        "u51"        "u52"        "grupohas"
 [61] "u54"        "u55.1"      "u55.2"      "u55.3"      "u55.4"      "u55.5"
 [67] "u56.1"      "u56.2"      "u56.3"      "u56.4"      "u56.5"      "u57.1"
 [73] "u57.2"      "u57.3"      "u57.4"      "u57.5"      "u58"        "u59"
 [79] "u60"        "u61"        "u62"        "grupodm"    "u64"        "u65.1"
 [85] "u65.2"      "u65.3"      "u65.4"      "u65.5"      "u66.1"      "u66.2"
 [91] "u66.3"      "u66.4"      "u66.5"      "u67.1"      "u67.2"      "u67.3"
 [97] "u67.4"      "u67.5"      "u68"        "u69"        "u70"        "u71"
[103] "u72.1"      "u72.2"      "u72.3"      "u72.4"      "u72.5"      "u73.1"
[109] "u73.2"      "u73.3"      "u73.4"      "u73.5"      "u74"        "u75"
[115] "u76"        "u77"        "u78"        "u79"        "u80"        "u81"
[121] "u82"        "u83"        "u84"        "u85"        "u86"        "u87"
[127] "u88"        "u89"        "u90.1"      "u90.2"      "u90.3"      "u90.4"
[133] "u90.5"      "u90.6"      "u90.7"      "u90.8"      "u90.9"      "u91"
[139] "u92"        "u93"        "uidade"     "ufxetar"    "tempentr"   "problema"
[145] "remedio"    "taf"        "sedent"     "naf"        "ptcage"     "cage"
[151] "imc"        "dieta3"     "dieta2"     "dietapts"   "ncmhas"     "ncmmhas"
[157] "tpconshas"  "ncmdm"      "ncmmdm"     "tpconsdm"   "ncodt"      "tpconsodt"
[163] "abep"       "abep2"      "abepcls"    "abepX2"     "escola"     "morisky"
```

Leitura do arquivo e seleção das variáveis de interesse

Use a função `names()` para listar os nomes das variáveis:

```
names(cursoR)
  [1] "nquest"     "dataentr"   "u6"         "datanasc"   "sexo"       "u9"
  [7] "u10"        "u10.1"      "u11"        "u12"        "u13"        "u14"
 [13] "u15"        "u16"        "u17"        "u18"        "u19"        "u20"
 [19] "u21"        "u22"        "u23"        "u24.1"      "u24.2"      "u24.3"
 [25] "u24.4"      "u25"        "u26"        "u27"        "u28"        "u29"
 [31] "u30"        "u31"        "u32"        "u33"        "u34.1"      "u34.2"
 [37] "u34.3"      "u34.4"      "u34.5"      "u34.6"      "u35"        "u36"
 [43] "u36.1"      "u37"        "u38"        "u39"        "u40"        "u41"
 [91] "u66.3"      "u66.4"      "u66.5"      "u67.1"      "u67.2"      "u67.3"
 [97] "u67.4"      "u67.5"      "u68"        "u69"        "u70"        "u71"
[103] "u72.1"      "u72.2"      "u72.3"      "u72.4"      "u72.5"      "u73.1"
[109] "u73.2"      "u73.3"      "u73.4"      "u73.5"      "u74"        "u75"
[115] "u76"        "u77"        "u78"        "u79"        "u80"        "u81"
[121] "u82"        "u83"        "u84"        "u85"        "u86"        "u87"
[127] "u88"        "u89"        "u90.1"      "u90.2"      "u90.3"      "u90.4"
[133] "u90.5"      "u90.6"      "u90.7"      "u90.8"      "u90.9"      "u91"
[139] "u92"        "u93"        "uidade"     "ufxetar"    "tempentr"   "problema"
[145] "remedio"    "taf"        "sedent"     "naf"        "ptcage"     "cage"
[151] "imc"        "dieta3"     "dieta2"     "dietapts"   "ncmhas"     "ncmmhas"
[157] "tpconshas"  "ncmdm"      "ncmmdm"     "tpconsdm"   "ncodt"      "tpconsodt"
[163] "abep"       "abep2"      "abepcls"    "abepX2"     "escola"     "morisky"
```

Há muito mais variáveis que as que podem nos interessar.

Vamos manter apenas as necessárias para alcançar os objetivos enunciados.

Redução do banco

Seleção das variáveis de interesse

- **O banco de dados é organizado com cada registro nas *filas* e cada variável nas *colunas*, que são entendidas pelo R como** `data.frame[fila,coluna]`
- As filas e colunas podem ser chamadas pelo seu número ou nome (como vimos)
- As variáveis de interesse eram: `sexo`, `dataentr`, `datanasc`, `peso`, `altura`, `abepcls`, `grupohas`, `grupodm`
- Diremos ao R então que nos faça uma cópia do banco `cursoR` apenas com essas variáveis
- O comando pode ser escrito em um só passo, mas pode ser mais fácil de entender a sintaxe se primeiro criamos um *vetor* com os nomes das variáveis

Redução do banco
Seleção das variáveis de interesse

Fúlvio Nedel
SPB/UFSC

- O banco de dados é organizado com cada registro nas *filas* e cada variável nas *colunas*, que são entendidas pelo R como `data.frame[fila,coluna]`
- **As filas e colunas podem ser chamadas pelo seu número ou nome (como vimos)**
- As variáveis de interesse eram: `sexo`, `dataentr`, `datanasc`, `peso`, `altura`, `abepcls`, `grupohas`, `grupodm`
- Diremos ao R então que nos faça uma cópia do banco `cursoR` apenas com essas variáveis
- O comando pode ser escrito em um só passo, mas pode ser mais fácil de entender a sintaxe se primeiro criamos um *vetor* com os nomes das variáveis

Redução do banco
Seleção das variáveis de interesse

- O banco de dados é organizado com cada registro nas *filas* e cada variável nas *colunas*, que são entendidas pelo R como `data.frame[fila,coluna]`
- As filas e colunas podem ser chamadas pelo seu número ou nome (como vimos)
- **As variáveis de interesse eram:** `sexo`, `dataentr`, `datanasc`, `peso`, `altura`, `abepcls`, `grupohas`, `grupodm`
- Diremos ao R então que nos faça uma cópia do banco `cursoR` apenas com essas variáveis
- O comando pode ser escrito em um só passo, mas pode ser mais fácil de entender a sintaxe se primeiro criamos um *vetor* com os nomes das variáveis

Redução do banco
Seleção das variáveis de interesse

- O banco de dados é organizado com cada registro nas *filas* e cada variável nas *colunas*, que são entendidas pelo R como `data.frame[fila,coluna]`
- As filas e colunas podem ser chamadas pelo seu número ou nome (como vimos)
- As variáveis de interesse eram: `sexo`, `dataentr`, `datanasc`, `peso`, `altura`, `abepcls`, `grupohas`, `grupodm`
- **Diremos ao R então que nos faça uma cópia do banco `cursoR` apenas com essas variáveis**
- O comando pode ser escrito em um só passo, mas pode ser mais fácil de entender a sintaxe se primeiro criamos um *vetor* com os nomes das variáveis

Redução do banco

Seleção das variáveis de interesse

- O banco de dados é organizado com cada registro nas *filas* e cada variável nas *colunas*, que são entendidas pelo R como `data.frame[fila,coluna]`
- As filas e colunas podem ser chamadas pelo seu número ou nome (como vimos)
- As variáveis de interesse eram: `sexo`, `dataentr`, `datanasc`, `peso`, `altura`, `abepcls`, `grupohas`, `grupodm`
- Diremos ao R então que nos faça uma cópia do banco `cursoR` apenas com essas variáveis
- **O comando pode ser escrito em um só passo, mas pode ser mais fácil de entender a sintaxe se primeiro criamos um *vetor* com os nomes das variáveis**

Redução do banco

Seleção das variáveis de interesse

Fúlvio Nedel
SPB/UFSC

Criar um banco com variáveis selecionadas:

```
vars <- c('peso', 'altura', 'sexo', 'dataentr', 'datanasc',
          'abepcls', 'grupohas', 'grupodm')
x <- cursoR[vars]
```

Temos então um `data frame` com todos os registros de "`cursoR`" e apenas as oito variáveis selecionadas:

```
class(x) ; nrow(x) ; ncol(x)
[1] "data.frame"
[1] 300
[1] 8
names(x)
[1] "peso"     "altura"   "sexo"     "dataentr" "datanasc" "abepcls"
[7] "grupohas" "grupodm"
```

Veja também a função `subset`

```
?subset
```

Seleção das variáveis de interesse

Criar um arquivo de dados do R com o novo banco

Tudo funcionou e o objeto 'x' é o banco de dados de interesse!!

Podemos então:

1. chamá-lo 'cursoR', sobreescrevendo o antigo, que não nos interessa mais;
2. salvá-lo no computador como um arquivo de dados do R (extensão '.rdata');
3. remover os outros objetos da sessão de trabalho; e
4. carregar o arquivo de dados criado,

para continuar o trabalho com uma sessão "limpa".

```
cursoR <- x
save(cursoR, file="cursoR.RData")
```

```
ls() # verificar os objetos no espaço de trabalho
rm(list=ls()) # apagar os objetos do espaço de trabalho
```

Apêndice

- Selecionar as variáveis pelo seu nome facilita a leitura humana da sintaxe, mas pode ser mais difícil de digitar e, eventualmente, algum comando necessitará a referência numérica[2]
- Se possível, dê nomes significativos às variáveis, é mais fácil trabalhar com uma variável chamada "sexo" que com uma variável chamada "u8" (por exemplo)
- Em bases com muitas variáveis pode ser difícil encontrar o nº de ordem das variáveis de interesse. Veja abaixo um exemplo de uso da função `%in%` :

Quais as variáveis de `cursoR` estão citadas em `vars`?

```
(nvar <- which(colnames(cursoR) %in% vars))
[1] 1 2 3 4 5 6 7 8
```

E poderíamos então criar o mesmo banco, com as variáveis na ordem do banco original:

```
x2 <- cursoR[nvar]
names(x2)
[1] "peso"     "altura"   "sexo"     "dataentr" "datanasc" "abepcls"
[7] "grupohas" "grupodm"
```

[2] Veja, por exemplo, o comando `names(...) <- ...`

Fúlvio Nedel
SPB/UFSC

Parte II

Limpeza e manejo de dados

Definição de missings

Roteiro

Fúlvio Nedel
SPB/UFSC

Carregar o arquivo de dados do R:

Listar os arquivos .RData no diretório

```
dir(patt='.rdata', ignore.case = TRUE)
```

```
[1] "cursoR.RData"   "usuarios.Rdata"
```

```
# Para tornar a busca "case-insensitive", use o argumento
# ignore.case = T
```

Carregar o arquivo

```
load("cursoR.RData")
```

Verificar sua presença no espaço de trabalho

```
ls()
```

```
[1] "cursoR"
```

As variáveis de análise

1. Verificação de inconsistências
2. Definição de *missings* e análise de sua ocorrência: algumas variáveis são particularmente afetadas?
3. Criação de novas variáveis
   - computação a partir de outras (idade, IMC)
   - recodificação (estado nutricional, faixa etária, ABEP)
4. Criar ou redefinir rótulos de variáveis e categorias

Observar o banco

```
head(cursoR) # observar os primeiros registros
```

```
  peso altura      sexo   dataentr   datanasc abepcls grupohas grupodm
1   64   1.44 Feminino  2011-03-19 1932-07-08      C2      Não    <NA>
2   50    1.5 Feminino  2011-03-19 1951-11-10      C1      Não     Não
3   69    1.6 Feminino  2011-03-20 1947-11-14      C2      Não    <NA>
4   75    1.6 Feminino  2011-03-20 1930-03-09      C2      Não     Não
5   60   1.58 Feminino  2011-03-20 1960-08-13      B1      Não    <NA>
6   73    1.6 Feminino  2011-03-20 1942-04-15       D      Não    <NA>
```

```
tail(cursoR) # observar os últimos registros
```

```
    peso altura      sexo   dataentr   datanasc abepcls grupohas grupodm
295   68   1.72 Masculino 2011-03-13 1936-08-26      C1      Não    <NA>
296   75   1.66 Feminino  2011-03-12 1955-07-21      C1      Não    <NA>
297   70   1.65 Feminino  2011-03-12 1936-06-28      C1      Não    <NA>
298   57   1.48 Feminino  2011-03-14 1934-03-25      B2      Não    <NA>
299   70    1.6 Feminino  2011-03-14 1942-11-12    <NA>      Não    <NA>
300   90   1.62 Feminino  2011-03-14 1958-05-11      C2      Não    <NA>
```

A estrutura do objeto 'cursoR'


```
str(cursoR)
'data.frame': 300 obs. of  8 variables:
 $ peso    : Factor w/ 74 levels "40.5","44.5",..: 21 6 30 37 17 35 28 1
  ..- attr(*, "label")= Named chr "u47. Qual o seu peso?"
  .. ..- attr(*, "names")= chr "peso"
 $ altura  : Factor w/ 46 levels "1.3","1.36","1.4",..: 4 9 19 19 17 19
  ..- attr(*, "label")= Named chr "u48. Qual a sua altura?"
  .. ..- attr(*, "names")= chr "altura"
 $ sexo    : Factor w/ 2 levels "Feminino ","Masculino": 1 1 1 1 1 1 1 1
  ..- attr(*, "label")= Named chr "u8. Sexo:"
  .. ..- attr(*, "names")= chr "sexo"
 $ dataentr: Date, format: "2011-03-19" "2011-03-19" ...
 $ datanasc: Date, format: "1932-07-08" "1951-11-10" ...
 $ abepcls : Factor w/ 7 levels "A2","B1","B2",..: 5 4 5 5 2 6 4 2 4 6 .
  ..- attr(*, "label")= Named chr "Classificação socioeconômica ABEP mod
  .. ..- attr(*, "names")= chr "abepcls"
 $ grupohas: Factor w/ 2 levels "Não","Sim": 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "label")= Named chr "u53. Desde <6 MESES ATRÁS> o(a) Sr.(a
  .. ..- attr(*, "names")= chr "grupohas"
 $ grupodm : Factor w/ 2 levels "Não","Sim": NA 1 NA 1 NA NA 1 NA NA NA
  ..- attr(*, "label")= Named chr "u63. Desde <6 MESES ATRÁS> o(a) Sr.(a
  .. ..- attr(*, "names")= chr "grupodm"
```

A estrutura do objeto 'cursoR'

PROBLEMA

```
str(cursoR)
'data.frame': 300 obs. of  8 variables:
 $ peso     : Factor w/ 74 levels "40.5","44.5",..: 21 6 30 37 17 35 28 1
  ..- attr(*, "label")= Named chr "u47. Qual o seu peso?"
  .. ..- attr(*, "names")= chr "peso"
 $ altura   : Factor w/ 46 levels "1.3","1.36","1.4",..: 4 9 19 19 17 19
  ..- attr(*, "label")= Named chr "u48. Qual a sua altura?"
  .. ..- attr(*, "names")= chr "altura"
 $ sexo     : Factor w/ 2 levels "Feminino ","Masculino": 1 1 1 1 1 1 1 1
  ..- attr(*, "label")= Named chr "u8. Sexo:"
  .. ..- attr(*, "names")= chr "sexo"
 $ dataentr: Date, format: "2011-03-19" "2011-03-19" ...
 $ datanasc: Date, format: "1932-07-08" "1951-11-10" ...
 $ abepcls : Factor w/ 7 levels "A2","B1","B2",..: 5 4 5 5 2 6 4 2 4 6 .
  ..- attr(*, "label")= Named chr "Classificação socioeconômica ABEP mod
  .. ..- attr(*, "names")= chr "abepcls"
 $ grupohas: Factor w/ 2 levels "Não","Sim": 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "label")= Named chr "u53. Desde <6 MESES ATRÁS> o(a) Sr.(a
  .. ..- attr(*, "names")= chr "grupohas"
 $ grupodm : Factor w/ 2 levels "Não","Sim": NA 1 NA 1 NA NA 1 NA NA NA
  ..- attr(*, "label")= Named chr "u63. Desde <6 MESES ATRÁS> o(a) Sr.(a
  .. ..- attr(*, "names")= chr "grupodm"
```

Corrigir a classe de variáveis

Corrigir a classe de `peso` e `altura`

```
str(cursoR[1:2])
```

```
'data.frame': 300 obs. of  2 variables:
 $ peso  : Factor w/ 74 levels "40.5","44.5",..: 21 6 30 37 17 35 28 16
  ..- attr(*, "label")= Named chr "u47. Qual o seu peso?"
  .. ..- attr(*, "names")= chr "peso"
 $ altura: Factor w/ 46 levels "1.3","1.36","1.4",..: 4 9 19 19 17 19 19
  ..- attr(*, "label")= Named chr "u48. Qual a sua altura?"
  .. ..- attr(*, "names")= chr "altura"
```

```
cursoR$peso = as.numeric(as.character(cursoR$peso))
cursoR$altura = as.numeric(as.character(cursoR$altura))
```

```
str(cursoR[1:2])
```

```
'data.frame': 300 obs. of  2 variables:
 $ peso  : num  64 50 69 75 60 73 68 59.5 65 67 ...
 $ altura: num  1.44 1.5 1.6 1.6 1.58 1.6 1.6 1.6 1.72 1.68 ...
```

Note que

- é preciso primeiro converter o `factor` em `character`
- ao fazer a conversão, perde-se o rótulo da variável
- para mantê-lo, guarde-o antes e depois o destine novamente à variável

Fúlvio Nedel
SPB/UFSC

Corrigir a classe de peso e altura

```
load("cursoR.RData")
str(cursoR[1:2])
```

```
'data.frame': 300 obs. of  2 variables:
 $ peso  : Factor w/ 74 levels "40.5","44.5",..: 21 6 30 37 17 35 28 16 23 26 ...
  ..- attr(*, "label")= Named chr "u47. Qual o seu peso?"
  .. ..- attr(*, "names")= chr "peso"
 $ altura: Factor w/ 46 levels "1.3","1.36","1.4",..: 4 9 19 19 17 19 19 19 32 28
  ..- attr(*, "label")= Named chr "u48. Qual a sua altura?"
  .. ..- attr(*, "names")= chr "altura"
```

```
rotulopeso = attributes(cursoR$peso)$label
rotuloaltura = attr(cursoR$altura, "label")

cursoR$peso = as.numeric(as.character(cursoR$peso))
cursoR$altura = as.numeric(as.character(cursoR$altura))

attr(cursoR$peso, "label") <- rotulopeso
attributes(cursoR$altura)$label <- rotuloaltura

str(cursoR[1:2])
```

```
'data.frame': 300 obs. of  2 variables:
 $ peso  : atomic  64 50 69 75 60 73 68 59.5 65 67 ...
  ..- attr(*, "label")= Named chr "u47. Qual o seu peso?"
  .. ..- attr(*, "names")= chr "peso"
 $ altura: atomic  1.44 1.5 1.6 1.6 1.58 1.6 1.6 1.6 1.72 1.68 ...
  ..- attr(*, "label")= Named chr "u48. Qual a sua altura?"
  .. ..- attr(*, "names")= chr "altura"
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
```

```
      peso             altura             sexo        dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20

    datanasc
 Min.   :1916-05-30
 1st Qu.:1936-10-18
 Median :1946-04-09
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08

 grupohas   grupodm
 Não :271   Não : 88
 Sim : 10   Sim :  7
 NA's: 19   NA's:205
```

```
summary(cursoR$abepcls)
```

```
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
      peso            altura             sexo        dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20

    datanasc          grupohas   grupodm
 Min.   :1916-05-30   Não :271   Não : 88
 1st Qu.:1936-10-18   Sim : 10   Sim :  7
 Median :1946-04-09   NA's: 19   NA's:205
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08
```

```
summary(cursoR$abepcls)
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
```

```
      peso            altura             sexo        dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20

    datanasc          grupohas   grupodm
 Min.   :1916-05-30   Não :271   Não : 88
 1st Qu.:1936-10-18   Sim : 10   Sim :  7
 Median :1946-04-09   NA's: 19   NA's:205
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08
```

```
summary(cursoR$abepcls)
```

```
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
      peso            altura             sexo          dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20

    datanasc          grupohas   grupodm
 Min.   :1916-05-30   Não :271   Não : 88
 1st Qu.:1936-10-18   Sim : 10   Sim :  7
 Median :1946-04-09   NA's: 19   NA's:205
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08
```

```
summary(cursoR$abepcls)
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
```

```
      peso            altura             sexo       
 Min.   : 40.50   Min.   :1.30   Feminino :198  
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102  
 Median : 72.00   Median :1.63                  
 Mean   : 76.93   Mean   :1.63                  
 3rd Qu.: 83.00   3rd Qu.:1.70                  
 Max.   :999.00   Max.   :1.92                  
```

```
   dataentr          
 Min.   :2011-03-12  
 1st Qu.:2011-03-15  
 Median :2011-03-18  
 Mean   :2011-03-17  
 3rd Qu.:2011-03-19  
 Max.   :2011-03-20  
```

```
   datanasc          
 Min.   :1916-05-30  
 1st Qu.:1936-10-18  
 Median :1946-04-09  
 Mean   :1946-08-23  
 3rd Qu.:1955-03-25  
 Max.   :1988-01-08  
```

```
 grupohas   grupodm  
 Não :271   Não : 88  
 Sim : 10   Sim :  7  
 NA's: 19   NA's:205  
```

```
summary(cursoR$abepcls)
```

```
  A2   B1   B2   C1   C2   D    E  NA's 
   1   11   61  102   76   30    1   18 
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
```

```
      peso            altura             sexo        dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20
```

```
    datanasc
 Min.   :1916-05-30
 1st Qu.:1936-10-18
 Median :1946-04-09
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08
```

```
 grupohas   grupodm
 Não :271   Não : 88
 Sim : 10   Sim :  7
 NA's: 19   NA's:205
```

```
summary(cursoR$abepcls)
```

```
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
      peso            altura             sexo        dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20

    datanasc          grupohas   grupodm
 Min.   :1916-05-30   Não :271   Não : 88
 1st Qu.:1936-10-18   Sim : 10   Sim :  7
 Median :1946-04-09   NA's: 19   NA's:205
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08
```

```
summary(cursoR$abepcls)
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Um breve sumário do banco

Veja também a função `describe`, do pacote `Hmisc`

```
summary(cursoR)
      peso            altura             sexo        dataentr
 Min.   : 40.50   Min.   :1.30   Feminino :198   Min.   :2011-03-12
 1st Qu.: 64.00   1st Qu.:1.57   Masculino:102   1st Qu.:2011-03-15
 Median : 72.00   Median :1.63                   Median :2011-03-18
 Mean   : 76.93   Mean   :1.63                   Mean   :2011-03-17
 3rd Qu.: 83.00   3rd Qu.:1.70                   3rd Qu.:2011-03-19
 Max.   :999.00   Max.   :1.92                   Max.   :2011-03-20

    datanasc          grupohas    grupodm
 Min.   :1916-05-30   Não :271   Não : 88
 1st Qu.:1936-10-18   Sim : 10   Sim :  7
 Median :1946-04-09   NA's: 19   NA's:205
 Mean   :1946-08-23
 3rd Qu.:1955-03-25
 Max.   :1988-01-08
```

```
summary(cursoR$abepcls)
  A2   B1   B2   C1   C2   D    E  NA's
   1   11   61  102   76   30    1   18
```

Definição de *missings*

Fúlvio Nedel
SPB/UFSC

A variável peso tem um valor máximo de 999 quilos, o que não é um valor válido, foi um código para indicar a falta de informação.
Vamos definir esses valores como missings (NAs):

```
cursoR$peso[cursoR$peso==999] <- NA
summary(cursoR$peso)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  40.50   64.00   72.00   73.85   83.00  126.00       1
```

A função `attach`

Podemos evitar o trabalho de chamar o nome do `data frame` antes de cada variável. Veja as funções `attach` e `detach`. **Use com moderação!**

```
attach(cursoR)
```

Definição de *missings*

GRAAL

Um caso mais complexo

- As variáveis `grupodm` e `grupohas` referem-se à participação em grupos de promoção da saúde para pessoas com, respectivamente, DM ou HAS.
- Se a pessoa não tem DM, a resposta para `grupodm` será um missing e assim para HAS e `grupohas`.
- A definição de missings nesses casos é um pouco mais complexa. Em seguida apresentam-se três formas de realizá-la:

```
As combinações possíveis
  grupohas grupodm
1      Sim     Sim
2      Sim    <NA>
3     <NA>     Sim
4      Não     Não
5      Não    <NA>
6     <NA>     Não
7     <NA>    <NA>
```

Definição de *missings*

Participa em grupos de hipertensos *ou* diabéticos?

```
grupo = rep(NA, 300)
grupo[grupodm == 'Sim' | grupohas == 'Sim'] = "Sim"
grupo[(grupodm == 'Não' & is.na(grupohas))] = "Não"
grupo[(grupohas == 'Não' & is.na(grupodm))] = "Não"
grupo[grupodm == 'Não' & grupohas == 'Não'] = "Não"
table(grupo, useNA = 'ifany')
```

```
grupo
 Não   Sim <NA>
 285    13    2
```

Definição de *missings*

Participa em grupos de hipertensos *ou* diabéticos?

```
grupo = rep(NA, 300)
grupo[grupodm == 'Sim' | grupohas == 'Sim'] = "Sim"
grupo[(grupodm == 'Não' & is.na(grupohas))] = "Não"
grupo[(grupohas == 'Não' & is.na(grupodm))] = "Não"
grupo[grupodm == 'Não' & grupohas == 'Não'] = "Não"
table(grupo, useNA = 'ifany')
```

```
grupo
 Não  Sim <NA>
 285   13    2
```

```
str(grupo)
```

```
 chr [1:300] "Não" "Não" "Não" "Não" "Não" "Não" "Não" "Não" ...
```

```
grupo <- factor(grupo, levels = c("Sim", "Não"),
                labels = c("Sim", "Não"))
str(grupo)
```

```
 Factor w/ 2 levels "Sim","Não": 2 2 2 2 2 2 2 2 2 2 ...
```

```
table(grupo, useNA = 'ifany')
```

```
grupo
 Sim  Não <NA>
  13  285    2
```

Definição de *missings*

Um caso mais complexo

Fúlvio Nedel
SPB/UFSC


```
grupo2 = rep("Não", 300)
grupo2[grupodm == 'Sim' | grupohas == 'Sim'] = "Sim"
grupo2[which(is.na(grupodm) & is.na(grupohas))] = NA
table(grupo2, useNA = 'ifany')
```

```
grupo2
 Não  Sim <NA>
 285   13    2
```

```
#
grupo3 = grupohas
grupo3[grupodm == 'Sim' | grupohas == 'Sim'] = "Sim"
grupo3[is.na(grupohas)] = grupodm[is.na(grupohas)]
table(grupo3, useNA = 'ifany')
```

```
grupo3
 Não  Sim <NA>
 285   13    2
```

Definição de *missings*
Um caso mais complexo

Podemos comprovar que o resultado é o desejado comparando as variáveis em todos os casos possíveis:

```
# Comprovar
x = cbind(cursoR[c('grupohas', 'grupodm')], " " = '-->',
                grupo, grupo2, grupo3)
x[c(77,11,23,2,1,21,25), ]
```

```
   grupohas grupodm      grupo grupo2 grupo3
77      Sim     Sim -->   Sim    Sim    Sim
11      Sim    <NA> -->   Sim    Sim    Sim
23     <NA>     Sim -->   Sim    Sim    Sim
2       Não     Não -->   Não    Não    Não
1       Não    <NA> -->   Não    Não    Não
21     <NA>     Não -->   Não    Não    Não
25     <NA>    <NA> --> <NA>   <NA>   <NA>
```

Definição de *missings*
Um caso mais complexo

Podemos comprovar que o resultado é o desejado comparando as variáveis em todos os casos possíveis:

```
# Comprovar
x = cbind(cursoR[c('grupohas', 'grupodm')], " " = '-->',
                grupo, grupo2, grupo3)
x[c(77,11,23,2,1,21,25), ]
```

```
   grupohas grupodm      grupo grupo2 grupo3
77      Sim     Sim -->    Sim    Sim    Sim
11      Sim    <NA> -->    Sim    Sim    Sim
23     <NA>     Sim -->    Sim    Sim    Sim
2       Não     Não -->    Não    Não    Não
1       Não    <NA> -->    Não    Não    Não
21     <NA>     Não -->    Não    Não    Não
25     <NA>    <NA> --->  <NA>   <NA>   <NA>
```

Para encontrar os casos, usei as funções `which` e `is.na`

```
which(x$grupohas == 'Sim' & x$grupodm == 'Sim')[1]
```

```
[1] 77
```

```
which(x$grupohas == 'Sim' & is.na(x$grupodm))[1]
```

```
[1] 11
```

etc.

Fúlvio Nedel
SPB/UFSC

Parte III

Limpeza e manejo de dados

Criar variáveis

Roteiro

Criação de novas variáveis

Computação a partir de outras

Idade: diferença em dias entre a data da entrevista e a de nascimento, dividida por 365,25

```
idade <- dataentr - datanasc
head(idade)
```

```
Time differences in days
[1] 28743 21679 23137 29596 18481 25176
```

```
idade <- trunc(as.numeric(idade/365.25))
head(idade)
```

```
[1] 78 59 63 81 50 68
```

Veja também

```
?difftime
```

Criação de novas variáveis

Computação a partir de outras

Fúlvio Nedel
SPB/UFSC

Idade: diferença em dias entre a data da entrevista e a de nascimento, dividida por 365,25

```
idade <- dataentr - datanasc
head(idade)
```

```
Time differences in days
[1] 28743 21679 23137 29596 18481 25176
```

```
idade <- trunc(as.numeric(idade/365.25))
head(idade)
```

```
[1] 78 59 63 81 50 68
```

Veja também

```
?difftime
```

IMC: Kg/m^2

```
imc <- peso/altura^2
str(imc)
```

```
 atomic [1:300] 30.9 22.2 27 29.3 24 ...
 - attr(*, "label")= Named chr "u47. Qual o seu peso?"
  ..- attr(*, "names")= chr "peso"
```

Criação de novas variáveis

Computação a partir de outras

Fúlvio Nedel
SPB/UFSC

Idade: diferença em dias entre a data da entrevista e a de nascimento, dividida por 365,25

```
idade <- dataentr - datanasc
head(idade)
```

```
Time differences in days
[1] 28743 21679 23137 29596 18481 25176
```

```
idade <- trunc(as.numeric(idade/365.25))
head(idade)
```

```
[1] 78 59 63 81 50 68
```

Veja também

```
?difftime
```

IMC: Kg/m^2

```
imc <- peso/altura^2
str(imc)
```

Problema: esse é o rótulo de peso

```
 atomic [1:300] 30.9 22.2 27 [illegible]
 - attr(*, "label")= Named chr "[illegible] Qual o seu peso?"
  ..- attr(*, "names")= chr "peso"
```

Criação de novas variáveis

Computação a partir de outras

Idade: diferença em dias entre a data da entrevista e a de nascimento, dividida por 365,25

```
idade <- dataentr - datanasc
head(idade)
```

```
Time differences in days
[1] 28743 21679 23137 29596 18481 25176
```

```
idade <- trunc(as.numeric(idade/365.25))
head(idade)
```

```
[1] 78 59 63 81 50 68
```

Veja também

```
?difftime
```

IMC: Kg/m^2

```
imc <- as.numeric(peso/altura^2)
str(imc)
```

```
 num [1:300] 30.9 22.2 27 29.3 24 ...
```

A função cut

Estado nutricional: é a *categorização* do IMC

```
imccat <- cut(imc, c(min(imc, na.rm=T), 25, 30, max(imc, na.rm=T)),
              include.lowest = T, right = F)
table(imccat)
```

```
imccat
[17.3,25)   [25,30) [30,46.1]
       99       116         84
```

Note o argumento `na.rm = TRUE` (abreviado como 'T') nas funções `min` e `max` ⇒ o peso tem 'missings', portanto o IMC também.

Os pontos de corte são os desejados, vamos rotular as categorias

```
imccat <- factor(imccat, labels=c('normal ou baixo peso',
                                  'sobrepeso',
                                  'obesidade' ) )
```

```
table(imccat)
```

```
imccat
normal ou baixo peso            sobrepeso            obesidade
                  99                  116                   84
```

A função `ifelse`

Obesidade: é a *dicotomização* do IMC

Poderíamos usar `cut`, mas é mais simples com `ifelse`

```
obeso <- factor(ifelse(imc >= 30, 1,2),
                labels = c("sim", "não")
                )
str(obeso)
 Factor w/ 2 levels "sim","não": 1 2 2 2 2 2 2 2 2 2 ...
```

```
addmargins(table(obeso))
obeso
sim não Sum
 84 215 299
summary(obeso)
 sim  não NA's
  84  215    1
```

Criação de novas variáveis
Recodificação

TAREFA

Faixa etária

Categorize a idade em faixas etárias

Agrupar categorias – a função %in%

ABEP

```
# Uma tabela pra conferir os resultados
rbind(freq = table(abepcls), cumfreq = cumsum(table(abepcls)))
```

```
        A2 B1 B2  C1  C2  D   E
freq     1 11 61 102  76  30   1
cumfreq  1 12 73 175 251 281 282
```

```
#
# Criar nova variável agrupando as classes
levels(abepcls)
```

```
[1] "A2" "B1" "B2" "C1" "C2" "D " "E "
```

```
abep2 <- factor(ifelse(abepcls %in% c("A1", "A2", "B1", "B2"), 1,
                  ifelse(abepcls %in% c("C1", "C2"), 2,
                    ifelse(abepcls %in% c("D ", "E "), 3, NA))),
                labels = c("A/B", "C", "D/E") )

# Verificar o resultado
addmargins(table(abep2))
```

```
abep2
A/B   C D/E Sum
 73 178  31 282
```

Rotular e modificar rótulos de variáveis

A função label{Hmisc}

```
# label(cursoR)
# Os rótulos são muito extensos, e 'label' ajusta o texto à direita, o que
# dificulta a leitura -> 'cbind' cria uma matriz com a coluna ajustada à esquerda:
cbind(label(cursoR))
```

```
         [,1]
peso     "u47. Qual o seu peso?"
altura   "u48. Qual a sua altura?"
sexo     "u8. Sexo:"
dataentr "u5. Data da entrevista:"
datanasc "u7. Qual é a sua data de nascimento?"
abepcls  "Classificação socioeconômica ABEP modificada"
grupohas "u53. Desde <6 MESES ATRÁS> o(a) Sr.(a) participou de algum grupo de hipe
grupodm  "u63. Desde <6 MESES ATRÁS> o(a) Sr.(a) participou de algum grupo de diab
```

Vamos arrumar as mais longas e as que criamos:

```
label(grupohas) <- "Participa em grupos de hipertensos"
label(grupodm) <- "Participa em grupos de diabéticos"
label(abep2) <- "Classificação ABEP agrupada"
label(imccat) <- "Estado nutricional"
label(grupo) <- "Participa em grupo de hipertensos ou diabéticos"
```

Rotular e modificar rótulos de categorias

Ao definir um fator, suas categorias são identificadas como *níveis*:

```
levels(imccat)
[1] "normal ou baixo peso" "sobrepeso"
[3] "obesidade"
```

Que podem ser trabalhados como qualquer objeto da classe `character`:

```
class(levels(imccat))
[1] "character"
```

Qual o rótulo da primeira categoria da variável `imccat`?

```
levels(imccat)[1]
[1] "normal ou baixo peso"
```

Como modificá-lo?

```
levels(imccat)[1] <- "normal"
levels(imccat)
[1] "normal"    "sobrepeso" "obesidade"
```

Finalmente!!!

Fúlvio Nedel
SPB/UFSC

Temos todas nossas variáveis

E já poderíamos começar a análise, mas antes vamos novamente "limpar a sujeira" do espaço de trabalho e guardar em arquivo o que nos interessa.

Nesse processo notaremos duas coisas (no mínimo):

- nem todas as mudanças realizadas estão no banco de dados
- não precisa, e mesmo assim podem ser salvas no arquivo de dados .RData

Organizar o espaço de trabalho

Voltemos à função attach

Ela guardou cursoR na memória e criou um novo ambiente de trabalho. As alterações realizadas, quando não destinadas especialmente a cursoR (com cursoR$nome-da-variavel), estão em objetos isolados no espaço de trabalho.

```
search()
 [1] ".GlobalEnv"        "cursoR"            "package:Hmisc"
 [4] "package:ggplot2"   "package:Formula"   "package:survival"
 [7] "package:lattice"   "package:xtable"    "package:knitr"
[10] "package:stats"     "package:graphics"  "package:grDevices"
[13] "package:utils"     "package:datasets"  "package:methods"
[16] "Autoloads"         "package:base"
ls()
 [1] "abep2"         "cursoR"        "grupo"         "grupo2"
 [5] "grupo3"        "grupodm"       "grupohas"      "idade"
 [9] "imc"           "imccat"        "obeso"         "rotuloaltura"
[13] "rotulopeso"    "x"
names(cursoR)
[1] "peso"     "altura"   "sexo"     "dataentr" "datanasc" "abepcls"
[7] "grupohas" "grupodm"
```

ainda a função attach

'grupohas': objeto, da classe `factor`, no espaço de trabalho.

```
label(grupohas)
[1] "Participa em grupos de hipertensos"
```

'cursoR$grupohas': variável de um objeto da classe `data frame` presente no espaço de trabalho.

```
label(cursoR$grupohas)

"u53. Desde <6 MESES ATRÁS> o(a) Sr.(a) participou de algum grupo de hip
```

Organizar o espaço de trabalho

Criar um novo banco de dados com as alterações


```
# passar para 'cursoR' os novos rótulos de grupohas e grupodm
label(cursoR$grupohas) <- label(grupohas)
# ou mandar diretamente a variável toda
cursoR$grupodm <- grupodm

# criar 'cursoR2' como uma cópia de 'cursoR', mas
# apenas com as variáveis de interesse pra análise
names(cursoR)
[1] "peso"     "altura"   "sexo"     "dataentr" "datanasc" "abepcls"
[7] "grupohas" "grupodm"
cursoR2 <- subset(cursoR, select = c(sexo, grupohas:grupodm))

# Incluir as outras variáveis
cursoR2$abep2 <- abep2
cursoR2$imc <- imc
cursoR2$imccat <- imccat
cursoR2$idade <- idade
cursoR2$obeso <- obeso
cursoR2$grupo <- grupo
```

Organizar o espaço de trabalho

Criar um novo banco de dados com as alterações


```
# Variáveis em 'cursoR2'
names(cursoR2)
[1] "sexo"     "grupohas" "grupodm"  "abep2"    "imc"      "imccat"
[7] "idade"    "obeso"    "grupo"
```

Reordenar as variáveis no banco novo

```
cursoR2 <- cursoR2[c(7,1,5:6,8,2:4,9)]
names(cursoR2)
[1] "idade"    "sexo"     "imc"      "imccat"   "obeso"    "grupohas"
[7] "grupodm"  "abep2"    "grupo"
```

Organizar o espaço de trabalho

Criar um novo banco de dados com as alterações


```
# Variáveis em 'cursoR2'
names(cursoR2)
[1] "sexo"     "grupohas" "grupodm"  "abep2"    "imc"      "imccat"
[7] "idade"    "obeso"    "grupo"
```

Reordenar as variáveis no banco novo

```
cursoR2 <- cursoR2[c(7,1,5:6,8,2:4,9)]
names(cursoR2)
[1] "idade"    "sexo"     "imc"      "imccat"   "obeso"    "grupohas"
[7] "grupodm"  "abep2"    "grupo"
```

A propósito...

Uma variável pode ser apagada com

```
banco$variavel <- NULL
```

E uma sequência de variáveis pode ser apagada com

```
banco[c(...)] <- NULL
```

Como em

```
cursoR2[6:7] <- NULL
```

A estrutura de 'cursoR2'

```
str(cursoR2)
'data.frame': 300 obs. of  7 variables:
 $ idade : num  78 59 63 81 50 68 67 76 74 78 ...
 $ sexo  : Factor w/ 2 levels "Feminino ","Masculino": 1 1 1 1 1 1 1 1 2
  ..- attr(*, "label")= Named chr "u8. Sexo:"
  .. ..- attr(*, "names")= chr "sexo"
 $ imc   : num  30.9 22.2 27 29.3 24 ...
 $ imccat: Factor w/ 3 levels "normal","sobrepeso",..: 3 1 2 2 1 2 2 1 1
  ..- attr(*, "label")= chr "Estado nutricional"
 $ obeso : Factor w/ 2 levels "sim","não": 1 2 2 2 2 2 2 2 2 2 ...
 $ abep2 : Factor w/ 3 levels "A/B","C","D/E": 2 2 2 2 1 3 2 1 2 3 ...
  ..- attr(*, "label")= chr "Classificação ABEP agrupada"
 $ grupo : Factor w/ 2 levels "Sim","Não": 2 2 2 2 2 2 2 2 2 2 ...
  ..- attr(*, "label")= chr "Participa em grupo de hipertensos ou diabét
```

Combinar bancos de dados

cbind e rbind

As funções `cbind` e `rbind` permitem com facilidade agregar variáveis (colunas) e registros (linhas) aos bancos de dados.

Podemos criar um novo banco de dados com as variáveis de `cursoR` e `cursoR2` com `cbind`, mas as variáveis que aparecem em ambos bancos se repetem no novo:

```
cursoR3 <- cbind(cursoR, cursoR2)
names(cursoR3)
 [1] "peso"     "altura"   "sexo"     "dataentr" "datanasc"
 [6] "abepcls"  "grupohas" "grupodm"  "idade"    "sexo"
[11] "imc"      "imccat"   "obeso"    "abep2"    "grupo"
table(names(cursoR3))>1

   abep2  abepcls   altura dataentr datanasc    grupo  grupodm
   FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
grupohas    idade      imc   imccat    obeso     peso     sexo
   FALSE    FALSE    FALSE    FALSE    FALSE    FALSE     TRUE
which(table(names(cursoR3))>1)
sexo
  14
```

Combinar bancos de dados

cbind e rbind

Fúlvio Nedel
SPB/UFSC

Temos de excluir essas variáveis em um dos bancos no momento da seleção:

```
# Variáveis que estão em 'cursoR2' mas não em 'cursoR'
(apenas <- setdiff(names(cursoR2), names(cursoR)) )
[1] "idade"  "imc"    "imccat" "obeso"  "abep2"  "grupo"
cursoR3 <- cbind(cursoR, cursoR2[apenas])
names(cursoR3)
 [1] "peso"     "altura"   "sexo"     "dataentr" "datanasc"
 [6] "abepcls"  "grupohas" "grupodm"  "idade"    "imc"
[11] "imccat"   "obeso"    "abep2"    "grupo"
```

A função `merge` amplia e (eventualmente) facilita essas possibilidades.

Vamos antes criar uma **variável de identificação do caso** (que deve haver, mas não a incluímos no início do trabalho) em cada banco.

É com base nessa variável comum que `merge` identificará os registros para a união dos bancos. Como não mudamos a ordem dos registros podemos identificar os casos pelo número da linha no banco de dados.

```
cursoR$id <- rownames(cursoR)
cursoR2$id <- rownames(cursoR2)
cursoR4 <- merge(cursoR, cursoR2[-2])
names(cursoR4)
 [1] "id"       "peso"     "altura"   "sexo"     "dataentr"
 [6] "datanasc" "abepcls"  "grupohas" "grupodm"  "idade"
[11] "imc"      "imccat"   "obeso"    "abep2"    "grupo"
names(cursoR3) %in% names(cursoR4)
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[14] TRUE
names(cursoR4) %in% names(cursoR3)
 [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[11]  TRUE  TRUE  TRUE  TRUE  TRUE
```

Guardar em arquivo

Deu tudo certo. Vamos guardar os dois 'data frames' de interesse no arquivo de dados, 'detachar' o banco colocado na memória e limpar a 'sujeira' do espaço de trabalho.

```
ls()
 [1] "abep2"        "apenas"       "cursoR"       "cursoR2"
 [5] "cursoR3"      "cursoR4"      "grupo"        "grupo2"
 [9] "grupo3"       "grupodm"      "grupohas"     "idade"
[13] "imc"          "imccat"       "obeso"        "rotuloaltura"
[17] "rotulopeso"   "x"
save(cursoR, cursoR2, file='cursoR.RData')
detach(cursoR)
rm(list=ls())
ls()
character(0)
```

Exercício

Fúlvio Nedel
SPB/UFSC

TAREFA

1. Crie um banco de dados com os primeiros e últimos dez registros de peso, altura e IMC (o banco deverá ter, portanto, 20 observações de três variáveis)
2. Verifique a estrutura do banco
3. Descreva um resumo do banco
4. Qual a média e o desvio-padrão do IMC?