Introdução ao uso do

em Ciências da Saúde

3. Leitura, limpeza e manejo de dados
c. Criar variáveis

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

19 de dezembro de 2017

Roteiro

Fúlvio Nedel
SPB/UFSC

1. Criar variáveis
   - Computar a partir de outras
   - Recodificar uma variável
2. Definir rótulos
3. Criar um banco para a análise de interesse
   - Combinar bancos de dados

Carregar o arquivos de dados do R:

Fúlvio Nedel
SPB/UFSC

Carregar o arquivo e "attachar" o banco

- Inicie com uma sessão vazia e carregue o arquivo de dados
- Verá que ele agora tem dois objetos,
  - o banco de dados com as variáveis selecionadas antes da transformação e
  - o vetor – de classe "factor" – criado anteriormente

```
rm(list=ls())
load('cursoR.RData')
ls()
[1] "cursoR" "grupo"
class(cursoR)
[1] "data.frame"
class(grupo)
[1] "factor"
```

```
attach(cursoR)
```

Roteiro

Fúlvio Nedel
SPB/UFSC

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
```

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 28.3 21 ...
 - attr(*, "label")= Named chr "047. 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 <- as.numeric(peso/altura^2)
str(imc)
```

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

Criação de novas variáveis
Recodificação

Fúlvio Nedel
SPB/UFSC

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`
$\Rightarrow$ 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
```

Criação de novas variáveis
Recodificação

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

Fúlvio Nedel
SPB/UFSC

TAREFA

Faixa etária

Categorize a idade em faixas etárias

Fúlvio Nedel
SPB/UFSC

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
```

Roteiro

Fúlvio Nedel
SPB/UFSC

1. Criar variáveis
   - Computar a partir de outras
   - Recodificar uma variável
2. Definir rótulos
3. Criar um banco para a análise de interesse
   - Combinar bancos de dados

Rotular e modificar rótulos de variáveis

Fúlvio Nedel
SPB/UFSC

A função label{Hmisc}

```
library(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 hip
grupodm  "u63. Desde <6 MESES ATRÁS> o(a) Sr.(a) participou de algum grupo de dia
```

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"            "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

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

Organizar o espaço de trabalho

Fúlvio Nedel
SPB/UFSC

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 hi
```

Roteiro

Fúlvio Nedel
SPB/UFSC

1. Criar variáveis
   - Computar a partir de outras
   - Recodificar uma variável
2. Definir rótulos
3. Criar um banco para a análise de interesse
   - Combinar bancos de dados

```
# 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
```

Fúlvio Nedel
SPB/UFSC

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
  ..- 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
  ..- 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 diabé
```

Combinar bancos de dados

Com as funções `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" "abepcls"
 [7] "grupohas" "grupodm"  "idade"    "sexo"     "imc"      "imccat"
[13] "obeso"    "abep2"    "grupo"
```

```
table(names(cursoR3))>1
```

```
   abep2  abepcls   altura dataentr datanasc    grupo  grupodm grupohas
   FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE
   idade      imc   imccat    obeso     peso     sexo
   FALSE    FALSE    FALSE    FALSE    FALSE     TRUE
```

```
which(table(names(cursoR3))>1)
```

```
sexo
  14
```

Combinar bancos de dados

Com as funções `cbind` e `rbind`

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" "abepcls"
 [7] "grupohas" "grupodm"  "idade"    "imc"      "imccat"   "obeso"
[13] "abep2"    "grupo"
```

Combinar bancos de dados

Com a função `merge`

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" "datanasc"
 [7] "abepcls"  "grupohas" "grupodm"  "idade"    "imc"      "imccat"
[13] "obeso"    "abep2"    "grupo"
names(cursoR3) %in% names(cursoR4)
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
names(cursoR4) %in% names(cursoR3)
 [1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[13]  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"  "cursoR3"
 [6] "cursoR4"  "grupo"    "grupodm"  "grupohas" "idade"
[11] "imc"      "imccat"   "obeso"
```

```
save(cursoR, cursoR2, file='cursoR.RData')
detach(cursoR)
rm(list=ls())
ls()
```

```
character(0)
```

Exercício

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?