Exercício Fixação dplyr - resolução

Marcelo Prudente e Rafael Giacomin

20 de março de 2018

Exercício 1: JOIN (merge)

- **Exercício 1**: o banco *dados_sociais* apresenta as Unidades da Federação, mas não especifica os nomes dos Estados ou mesmo as regiões a que pertencem esse estados. De acordo com os comando aprendidos, tente:
  1. Ler arquivo "uf.csv" no **R**. Salvar em um objeto com nome **uf**. Depois disso, encontre:
     - Qual a variável em comum entre os dois arquivos (*dados_sociais* e *uf*)?
     - Tente mesclar os dados de tal forma que todas observações do banco dados sociais contenham os nomes dos estados e suas respectivas regiões. Crie um objeto chamado *dados_sociais_uf*.

RESOLUÇÃO

```
# ler arquivos
dados_sociais <- fread("dados_sociais.csv", dec = ",")
uf<- fread("uf.csv")

# variável em comum: uf
# mesclar bases
ds1 <- inner_join(uf, dados_sociais, by = "uf")
```

Exercício 2: JOIN (merge)

- **Exercício 2**: O banco *dados_sociais* não traz a população Estadual para cada período. Assim, tente:
  1. Ler o arquivo "uf2.csv" no **R**.
     - Identifique as chaves de cruzamento.
     - Tente mesclar os dados de tal forma que todas observações do banco dados sociais contenham os nomes dos estados e suas respectivas populações em cada um dos períodos.

RESOLUÇÃO

```
# chaves: UNIDADE_FEDERACAO e ano
ds2 <- inner_join(uf2, dados_sociais,
                        by = c("UNIDADE_FEDERACAO"  = "uf",
                               "ano" = "ano"))
```

Exercício 3: comandos dplyr com base dados FIES

Baixar base do FIES

1. Selecionar diretório de trabalho: pasta dados
2. Baixar dados do FIES na pasta dados
3. carregar pacote *dplyr*

RESOLUÇÃO

```
# fixar diretório
setwd("C:/meu_diretorio")
```

```r
# baixar arquivos
fies <- fread("fies_sample.csv")

# carregar pacote dplyr
library(tidyverse)
library(dplyr) # carrega apenas o dplyr
```

Observar dicionário de dados da base FIES

Em todas as análises das bases de dados, é importante observar o dicionário de dados disponibizado pela instituição que produziu ou compilou a informação. Em geral, os dicionários identificam as informações de cada coluna como a descrição das variáveis, a classe (numérica, character, entre outras) e o comprimento de cada variável. Entretanto, no caso do FIES, o FNDE apenas fornece a descrição das variáveis.

Os dados originais do FIES para cada ano estão disponíveis no site dados abertos do FNDE, **aqui**. O dicionário de dados pode ser encontrado **neste link**.

Exercícios de fixação dos comando básicos do dplyr:

1. Verifique a classe e a estrutura do objeto fies

RESOLUÇÃO

```r
str(fies)
```

2. Descubra o nome das colunas do banco FIES

RESOLUÇÃO

```r
colnames(fies)
```

3. Selecione variáveis que começam com o nome "DS"

RESOLUÇÃO

```r
fies %>% select(starts_with("DS"))
```

4. Selecione variáveis que contém "CURSO" em seu nome

RESOLUÇÃO

```r
fies %>% select(contains("CURSO"))
```

5. Selecione variáveis que terminam com "O"

RESOLUÇÃO

```r
fies %>% select(ends_with("O"))
```

6. Selecione algumas variáveis de interesse

RESOLUÇÃO

```r
nomes <-  c("SG_UF", "NO_IES", "CO_CONTRATO_FIES", "NU_ANO", "VL_MENSALIDADE", "DT_NASCIMENTO",
            "ST_ENSINO_MEDIO_ESCOLA_PUBLICA", "CO_CIDADE", "DS_SEXO", "CO_AGENTE_FINANCEIRO", "DS_RACA_
            "CO_CURSO","DS_CURSO" )

fies <- fies%>% select(nomes)
```

7. Retire algumas variáveis do banco

RESOLUÇÃO

```
fies <- fies%>% select(-CO_AGENTE_FINANCEIRO)
```

8. Reordenar as variáveis

RESOLUÇÃO

```
fies <- fies %>% select(CO_CONTRATO_FIES:NU_ANO, everything())
```

9. Renomear variável:
   - Sigla da UF como uf;
   - Valor da mensalidade como mens
   - Descrição do curso como curso

RESOLUÇÃO

```
rename(fies, uf = SG_UF,
       mesn = VL_MENSALIDADE,
       curso = DS_TIPO_CURSO)
```

10. Filtrar bolsas do FIES do Distrito Federal:
    - quantas observações permanecem?

RESOLUÇÃO

```
filter(fies, SG_UF =="DF") %>% nrow()
```

11. filtrar bolsas do FIES de Alagoas e Sergipe - utilizar operadores lógicos
    - quantas observações permanecem?

RESOLUÇÃO

```
fies %>% filter(SG_UF =="SE" | SG_UF =="AL") %>% nrow()
```

12. filtrar bolsas do FIES de Alagoas e Sergipe - utilizar operador %in%
    - quantas observações permanecem?

RESOLUÇÃO

```
fies %>% filter(SG_UF %in% c("AL", "SE")) %>% nrow()
```

13. filtrar bolsas do FIES de Alagoas e Sergipe para o curso de medicina
    - quantas observações permanecem?

RESOLUÇÃO

```
fies %>% filter(SG_UF %in% c("AL", "SE") & DS_CURSO == "MEDICINA") %>% nrow()
```

14. filtrar bolsas do FIES de Alagoas e Sergipe para o curso de medicina e direito
    - quantas observações permanecem?

RESOLUÇÃO

```
fies %>% filter(SG_UF %in% c("AL", "SE") & DS_CURSO %in% c("MEDICINA", "DIREITO")) %>% nrow()
```

15. filtrar bolsas do fies cujo valor exceda R$ 1.000,00
    - quantas observações permanecem?

RESOLUÇÃO

```
fies%>%filter(VL_MENSALIDADE > 1000)
```

16. filtrar bolsas do fies que não sejam do Estado de São Paulo ou Minas e com valores de mensalidade menores ou iguais que R$ 600

- quantas observações permanecem?

RESOLUÇÃO

```
fies%>%filter(!SG_UF %in% c("SP", "MG") & VL_MENSALIDADE<=600) %>% nrow()
```

17. filtrar bolsas do fies cujos nomes dos Estados contenham a letra "S"

RESOLUÇÃO

```
# uso do grepl
fies%>%filter(grepl("S", SG_UF))

# conferir os Estados com "S"
fies%>%filter(grepl("S", SG_UF)) %>% distinct(SG_UF)
```

18. ordenar os dados pelo valor da mensalidade (da maior para a menor e da menor para maior)
    - Qual a diferença?

RESOLUÇÃO

```
# ordenar pelo menor valor
fies %>% arrange(VL_MENSALIDADE)

# ordenar pelo maior valor
fies %>% arrange(- VL_MENSALIDADE)
fies %>% arrange(desc(VL_MENSALIDADE)) # idêntico

# ATENÇÃO: se o comando arrange apresentar erro, verificar se a variável foi importada como numérica.
# caso não tenha sido, você pode utilizar o comando gsub:
fies$VL_MENSALIDADE <- (as.numeric(gsub(",", ".", fies$VL_MENSALIDADE)))
```

19. Encontrar a média, o máximo e o mínimo da mensalidade por Estado.

RESOLUÇÃO

```
fies %>%
  group_by(SG_UF) %>%
  summarise(media_mens = mean(VL_MENSALIDADE, na.rm = T),
            max_mens = max(VL_MENSALIDADE, na.rm = T),
            min_mens = min(VL_MENSALIDADE, na.rm = T))
```

20. Encontrar a média, o máximo e o mínimo da mensalidade por região
    - **Dica:** utilizar o comando join() para cruzar o banco fies com o banco uf.

RESOLUÇÃO

```
# você pode fazer um pipe com o inner join
inner_join(fies, uf, by =c("SG_UF" = "sg_uf")) %>%
  group_by(regiao) %>%
  summarise(media_mens = mean(VL_MENSALIDADE, na.rm = T),
            max_mens = max(VL_MENSALIDADE, na.rm = T),
            min_mens = min(VL_MENSALIDADE, na.rm = T))
```

21. Encontrar o valor total pago a título de mensalidade a cada IES **RESOLUÇÃO**

```
fies %>%
  group_by(NO_IES) %>%
  summarise(total_pago = sum(VL_MENSALIDADE, na.rm = T)) %>%
  # bônus! calcular percentual
  mutate( perc_total_pago = (total_pago/sum(total_pago)) * 100,
```

```
          # arredondar valores
          perc_total_pago = round(perc_total_pago, 2)) %>%
  arrange(desc(perc_total_pago))
```

Exercício 4: JOIN (merge)

- **Exercício 4**: Imagine que você está diante de duas bases de dados administrativas. Na primeira, há o número do NIS de um cidadão. Na segunda, há o número do PIS.
    1. Ler os arquivos "nis_exemplo.csv" e "pis_exemplo.csv" no **R**. Certifique-se que todas as variáveis foram importadas corretamente.

RESOLUÇÃO

```
# ler arquivos
nis <- read_csv2("nis_exemplo.csv",
                 locale = locale(encoding = "Latin1"))
pis <- read_csv2("pis_exemplo.csv",
                 locale = locale(encoding = "Latin1"))

# verificar importação
str(nis)
glimpse(nis)
summary(nis)

str(pis)
glimpse(pis)
summary(pis)
```

2. Identifique as chaves de cruzamento.

RESOLUÇÃO

```
# nis favorecido e pis pescador
```

3. Tente mesclar os dados de tal forma que o resultado identifique:
    + todos os caso coincidentes.
    + apenas os casos em que a base *nis* cruza com a base *pis*.
    + todos os casos não coincidentes.
    + todos os casos.

- Salve cada uma das consultas em objetos distintos.

RESOLUÇÃO

```
coincidentes <- inner_join(nis, pis, by = c("Nis Favorecido" = "PIS Pescador"))
casos_nis <- right_join(nis, pis, by = c("Nis Favorecido" = "PIS Pescador"))
distintos <- anti_join(nis, pis, by = c("Nis Favorecido" = "PIS Pescador"))
todos <- full_join(nis, pis, by = c("Nis Favorecido" = "PIS Pescador"))
```

Exercício 5: extrair contratos repetidos

- Com a base do FIES, encontre:

1. há valores duplicados de contratos nesse banco?
    - Identifique o número de observações duplicadas e únicas.
2. remover valores duplicados baseados em no código do contrato

3. remover valores duplicados baseados em múltiplas variáveis: código do contrato e número do semestre
4. encontrar a média, o máximo e o mínimo da mensalidade e da quantidade de semestres financiados por UF e por Região.
5. Obter a média, mediana, máximo, mínimo e desvio padrão de todas as variáveis numéricas do banco fies

RESOLUÇÃO

```
# duplicados
fies %>%
  summarise( unicos = n_distinct(CO_CONTRATO_FIES),
             duplicados = sum(duplicated(CO_CONTRATO_FIES)))

# remover duplicados
fies %>%
  distinct(CO_CONTRATO_FIES, .keep_all = T)

# por contrato e semestre
fies %>%
  distinct(CO_CONTRATO_FIES, NU_SEMESTRE, .keep_all = T)
```

6. transformar todas as variáveis de character em factor.

RESOLUÇÃO

```
# Atenção: nesse caso, atribuir a mudança ao banco "fies"
fies<- fies %>%
  mutate_if(is.character, as.factor )
```

7. obter os níveis de todas as variáveis que são fatores

RESOLUÇÃO

```
# Atenção: nesse caso, atribuir a mudança ao banco "fies"
fies %>%
  summarise_if(is.factor, funs(nlevels(.)))
```