Desafio Escas - Limpeza e Integração de Dados

RSTUDIO - GITHUB - ORGANIZAÇÃO DE DADOS

Prof. Dr. Fernando Lima

Carregar Pacotes

```
rm(list = ls(all = TRUE))
library(here)
library(dplyr)
library(stringr)
library(tidyr)
```

CARREGAR ARQUIVOS E CORREÇÃO GERAL

Vamos carregar o arquivo que com a formatação que discutimos.

```
planilha01 <-  readr::read_csv2(
  here("01_dadosDeEntrada", "desafioTabela01.csv"),
  show_col_types = FALSE)
```

Como lidar com dados agrupados em uma só coluna?

Como vimos na preparação dos dados, as estimativas de área de vida estão com os valores do invervalo de confiança na mesma célula.

Essa informação não pode ser usada desta maneira.

São três variáveis, então vamos separar cada uma em uma coluna.

ÁREA DE VIDA Primeiro vamos extrair a primeira sequência de dígitos, correspondente a área de vida das onças.

Note que é necessário converter para numérico usando `as.numeric` .

```
planilha01 <- planilha01 %>%
  dplyr::mutate(
    HRC = as.numeric(
      str_extract(HR, "\\d+")
      )
  )
```

INTERVALO DE CONFIANÇA (IC-) Agora precisamos extrair o valor mínimo do intervalo de confiança, o primeiro valor entre parênteses.

Existem diversas formas de fazer isso.

Nesse caso vamos usar o `(` e o `-` como referência.

Ou seja, vamos dizer ao *R* para pegar o valores que estão entre o `(` e o `-`. que correspondem aos valores mínimos de intervalo de confiança.

A coluna será criada automaticamente.

```
planilha01 <- planilha01 %>%
  dplyr::mutate(
    ICL = as.numeric(
      sub("\\-.*", "", sub(".*\\(", "", HR))
      )
  )
```

INTERVALO DE CONFIANÇA (IC+) Agora precisamos dos valores máximos de intervalo de confiança.

Vamos usar a mesma estratégia que usamos para os intervalos mínimos.

Mas dessa vez vamos inverter as referências.

Vamos dizer ao *R* para pegar os valores que estão entre o `-` e o `)` .

```
planilha01 <- planilha01 %>%
  dplyr::mutate(
    ICH = as.numeric(
      sub("\\).*", "", sub(".*\\-", "", HR))
      )
  )
```

REVISÃO DA PLANILHA Um detalhe ficou para trás.

O caluna com os números de localizações não está sendo interpretada como valores númericos.

O símbolo `-` causou isso, precisamos removê-los.

```
planilha01 <- planilha01 %>%
  dplyr::mutate(
    `No. loc.` = as.numeric(
      na_if(`No. loc.`,"-")
    )
  )
```

Não precisamos mais da coluna original.

Podemos removê-la sem problemas.

```
planilha01 <- planilha01 %>%
  dplyr::select(
    -HR
  )
```

```
readr::write_csv2(planilha01, here("03_dadosDeSaida", "desafioEscasPlanilha01.txt"))
```

EXPORTAR ARQUIVO `*.txt`

COMENTÁRIOS FINAIS Lembre-se que nenhum desses ajustes e modificações alterou o arquivo original.

A vantagem do script é que você consegue rastrear todas as mudanças feitas, mantendo um histórico de tudo que fez.

Por isso é extremamente importante que os códigos sejam bem comentados.

Nós fizemos as alterações passo a passo, para explicar melhor cada etapa.

Mas é possível fazer tudo de uma vez com usando o operador pipe `%>%`, presente em todos os pacotes no `tidyverse`. Ele permite o encadeamento de ações, é como se dissesse `em seguida faça` no script.

Para testar é necessário carregar o arquivo de novo, já que excluímos a coluna com as informações.

```
planilha01 <-  readr::read_csv2(
  here("01_dadosDeEntrada", "desafioTabela01.csv"),
  show_col_types = FALSE)
```

```
planilha01 <- planilha01 %>%
  # Extraia a primeira sequência de dígitos para mim e converta para numérico.
  dplyr::mutate(
    HRC = as.numeric(
      str_extract(HR, "\\d+")
      )
  ) %>%
  # Extraia os valores entre o "(" e "-" e converta para numérico
  dplyr::mutate(
    ICL = as.numeric(
      sub("\\-.*", "", sub(".*\\(", "", HR))
      )
  ) %>%
  # Extraia os valores entre o "-" e ")" e converta para numérico
  dplyr::mutate(
    ICH = as.numeric(
      sub("\\).*", "", sub(".*\\-", "", HR))
      )
  ) %>%
  # Converta para numérico
  dplyr::mutate(
    `No. loc.` = as.numeric(
      na_if(`No. loc.`,"-")
    )
  ) %>%
  # Exclua a coluna original.
  dplyr::select(
    -HR
  )
```