Exercícios aula 1

Beatriz Milz

10/03/2020

Exercícios da aula do dia 06/03/2020 - [Link](https://jonnyphillips.github.io/Ciencia_de_Dados/introducao.html).

## Exercício 1

1. Comece um novo script de Rmarkdown e accesse o banco de dados gapminder no pacote dslabs. Estes dados descrevem as estatísticas resumidas para países do mundo desde 1960.

# Carregar a biblioteca gapminder  
library(gapminder)

1. Produza um mini-relatório com algumas frases curtas em HTML que descreva o banco de dados gapminder:

* Número de observações
* Número e nomes das variáveis
* O tipo de cada variável
* O que representa cada observação no banco de dados?

O banco de dados gapminder possui 1704 observações e 6 variáveis. As variáveis que estão disponíveis neste dataframe são: country, continent, year, lifeExp, pop, gdpPercap.

* A variável country é do tipo factor e representa o país de qual aquela observação se refere.
* A variável continent é do tipo factor e representa o continente onde está o país de qual aquela observação se refere.
* A variável year é do tipo integer e representa o ano de qual aquela observação se refere.
* A variável lifeExp é do tipo numeric e representa a expectativa de vida no país a qual aquela observação se refere.
* A variável pop é do tipo integer e representa a população no país a qual aquela observação se refere.
* A variável gdpPercap é do tipo numeric e representa o PIB per capita no país a qual aquela observação se refere.

1. Verifique que as respostas 2(a) e 2(b) são calculadas automaticamente no relatório usando funções do R e in-line code para inserir as respostas numa frase.

Sim, resposta no exercício 2.

1. Calcule o valor do número de observações multiplicado pelo número de colunas.

# Multiplica número de colunas pelo número de linhas  
ncol(gapminder) \* nrow(gapminder)

## [1] 10224

1. Inclua uma tabela do banco de dados gapminder no relatório.

# Incluir uma tabela Do dataframe gapminder  
  
gapminder

## # A tibble: 1,704 x 6  
## country continent year lifeExp pop gdpPercap  
## <fct> <fct> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.8 8425333 779.  
## 2 Afghanistan Asia 1957 30.3 9240934 821.  
## 3 Afghanistan Asia 1962 32.0 10267083 853.  
## 4 Afghanistan Asia 1967 34.0 11537966 836.  
## 5 Afghanistan Asia 1972 36.1 13079460 740.  
## 6 Afghanistan Asia 1977 38.4 14880372 786.  
## 7 Afghanistan Asia 1982 39.9 12881816 978.  
## 8 Afghanistan Asia 1987 40.8 13867957 852.  
## 9 Afghanistan Asia 1992 41.7 16317921 649.  
## 10 Afghanistan Asia 1997 41.8 22227415 635.  
## # ... with 1,694 more rows

1. Verifique que o seu relatório tem título, autor e data corretos, e comentários suficientes para explicar o que faça cada linha de código.

OK!

1. Tente compilar o seu script final para um documento de Word.

OK!

## 

## Exercício 3: Encontre o erro em todos os códigos abaixo:

library("nycflights13")

## Warning: package 'nycflights13' was built under R version 3.6.3

library("tidyverse")

## -- Attaching packages -------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.4  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ----------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

Q1)

dim**[flights]**

dim(flights)

## [1] 336776 19

Q2)

di**n**(flights)

dim(flights)

## [1] 336776 19

Q3)

v1 <- (“pato”, “cachorro”, “minhoca”, “lagarto”)

v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
v1

## [1] "pato" "cachorro" "minhoca" "lagarto"

Q4)

v**2** <- c(“1”, “2”, “3”, “4”)

v1 + 42

v2 <- c(1, 2, 3, 4)  
v2 + 42

## [1] 43 44 45 46

Q5)

v1 <- c(“pato”, “cachorro”, “minhoca”, “lagarto”

v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
v1

## [1] "pato" "cachorro" "minhoca" "lagarto"

Q6)

v3 <- c(33 31 40 25 27 40)

v3 <- c(33, 31, 40, 25, 27, 40)  
v3

## [1] 33 31 40 25 27 40

Q7)

v1 <- c(pato, cachorro, minhoca, lagarto)

v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
v1

## [1] "pato" "cachorro" "minhoca" "lagarto"

Q8)

v1 <- c(“pato” “cachorro” “minhoca” “lagarto”)

v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
v1

## [1] "pato" "cachorro" "minhoca" "lagarto"

Q9)

v3 <- **C**(33, 31, 40, 25, 27, 40)

v3 <- c(33, 31, 40, 25, 27, 40)  
v3

## [1] 33 31 40 25 27 40

Q10)

v1 <- c(“pato”, “cachorro”**"**, “minhoca”, “lagarto”)

v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
  
v1

## [1] "pato" "cachorro" "minhoca" "lagarto"

Q11)

v1 <- c(“pato”, “cachorro”, “minhoca”, “lagarto”)

v4 <- c(33, 31, 40, 25, 27, 40)

myData <- tibble(v1, v4)

# v1 e v4 precisam ter o mesmo comprimento  
  
v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
v4 <- c(33, 31, 40, 25)  
myData <- tibble(v1, v4)  
myData

## # A tibble: 4 x 2  
## v1 v4  
## <chr> <dbl>  
## 1 pato 33  
## 2 cachorro 31  
## 3 minhoca 40  
## 4 lagarto 25

Q12)

v1 <- c(“pato”, “cachorro”, “minhoca”, “lagarto”)

v4 <- c(33, 31, 40, 25)

myData <- tibble(v1 = animal, v4 = idade)

v1 <- c("pato", "cachorro", "minhoca", "lagarto")  
v4 <- c(33, 31, 40, 25)  
myData <- tibble(animal = v1, idade = v4)  
  
myData

## # A tibble: 4 x 2  
## animal idade  
## <chr> <dbl>  
## 1 pato 33  
## 2 cachorro 31  
## 3 minhoca 40  
## 4 lagarto 25