Métodos Quantitativos

Aula 04. Estatísticas descritivas no R

Pedro H. G. Ferreira de Souza

pedro.ferreira@ipea.gov.br

Mestrado Profissional em Políticas Públicas e Desenvolvimento

Instituto de Pesquisa Econômica Aplicada (Ipea)

03 out. 2022

Recapitulação

Introdução

Mais funções no R

Estatísticas descritivas

Recapitulação

Introdução

Mais funções no R

Estatísticas descritivas

Aula passada

- Boas práticas para organização de projetos
- Scripts, pacotes e funções no R
- Classes mais importantes de objetos atômicos
  - Character, logical, integer e numeric
- Classes mais comuns de coleções de objetos
  - Vetores, matrizes e data frames (ou tibbles)
- Manipulação de dados com o `dplyr`
  - filter, select, arrange, mutate
  - (**Atividade #3**) summarise e group_by

Recapitulação

Introdução

Mais funções no R

Estatísticas descritivas

Introdução

Objetivos de hoje

- Consolidar lições da aula passada
- Apresentar (brevemente) mais funções úteis do R
- Estatísticas descritivas uni- e bivariadas no R

Preliminares

- No **RStudio**, criar novo projeto em uma pasta vazia
- No **Github**, baixar o zip com o material de apoio da aula 04 e descompatar o arquivo na pasta do projeto

Pacotes que usaremos hoje

Continuaremos usando os pacotes da última aula e da atividade #3: `here`, `tidyverse`, `summarytools` e `gapminder`. Também usaremos um pacote novo, o `nycflights13`.

Exercício: carregar os quatro pacotes no script de vocês.

Pacotes que usaremos hoje

Continuaremos usando os pacotes da última aula e da atividade #3: `here`, `tidyverse`, `summarytools` e `gapminder`. Também usaremos um pacote novo, o `nycflights13`.

Exercício: carregar os quatro pacotes no script de vocês.

```
library(here)
library(tidyverse)
library(summarytools)
library(gapminder)
library(nycflights13)
```

Se der algum erro, lembrem-se que os pacotes precisam ser instalados antes do primeiro uso!

Data frames que usaremos hoje

Usaremos vários bancos de dados diferentes. Dois deles nós já vimos:

- `Oxfam/Datafolha`, que vimos na aula passada
- `Gapminder`, que vimos na atividade #3

Além disso, usaremos quatro data frames do pacote `nycflights13`:

- `flights`
- `airlines`
- `airports`
- `weather`

Os dicionários de dados dessas bases podem ser acessados do console com `?NOME` ou `help(NOME)`: por exemplo, `?flights` ou `help(flights)`.

Data frames que usaremos hoje

Exercício: carregar logo os data frames no workspace para poupar trabalho depois.

Data frames que usaremos hoje

Exercício: carregar logo os data frames no workspace para poupar trabalho depois.

```
oxfam.df <- readr::read_csv(...)
gapminder.df <- ...
voos.df <- ...
aeroportos.df <- ...
cias.df <- ...
clima.df <- ...
```

Data frames que usaremos hoje

Exercício: carregar logo os data frames no workspace para poupar trabalho depois.

```
oxfam.df <- read_csv(here("dados", "brutos",
                          "datafolha_oxfam.csv"))
gapminder.df <- gapminder
voos.df <- flights
aeroportos.df <- airports
cias.df <- airlines
clima.df <- weather
```

No código acima, o nome dos pacotes é opcional porque eles já foram carregados.

Data frames que usaremos hoje

Exercício: quantas linhas e quantas colunas cada data frame tem?

Data frames que usaremos hoje

Exercício: quantas linhas e quantas colunas cada data frame tem?

```
dim(aeroportos.df)
dim(cias.df)
dim(clima.df)
dim(voos.df)
dim(oxfam.df)
dim(gapminder.df)
```

```
## [1] 1458    8
## [1] 16  2
## [1] 26115    15
## [1] 336776     19
## [1] 2086   23
## [1] 1704    6
```

Recapitulação

Introdução

Mais funções no R

Estatísticas descritivas

Mais sobre o `dplyr`

Na aula passada, exploramos quatro das principais funções do pacote `dplyr`, incluído no pacote `tidyverse`:

filter() para selecionar linhas

select() para selecionar colunas

arrange() para reordenar os dados de acordo com uma ou mais colunas

mutate() para criar ou modificar colunas

Também vimos como encadear funções com o operador **pipe** %>%.

No exercício #3, vocês aprenderam mais dois comandos úteis do `dplyr`:

group_by() para agrupar linhas por categorias de uma coluna

summarise() para calcular estatísticas de colunas selecionadas

O `group_by` no `dplyr`

Exemplo: **filter**, **group_by** e **mutate** para calcular a a renda média, mínima e máxima por continente e a renda relativa dos países em 2007

O group_by no dplyr

Exemplo: **filter**, **group_by** e **mutate** para calcular a a renda média, mínima e máxima por continente e a renda relativa dos países em 2007

```
renda.relativa.df <-
  gapminder.df %>%
    filter(...) %>%
      group_by(...) %>%
        mutate(media_continente = ...,
               minimo_continente = ...,
               maximo_continente = ...,
               relativa = ...) %>%
          arrange(...)
```

O group_by no dplyr

Exemplo: **filter**, **group_by** e **mutate** para calcular a a renda média, mínima e máxima por continente e a renda relativa dos países em 2007

```
renda.relativa.df <-
  gapminder.df %>%
    filter(year == 2007) %>%
      group_by(continent) %>%
        mutate(media_continente = mean(gdpPercap),
               minima_continente = min(gdpPercap),
               maxima_continente = max(gdpPercap),
               relativa = gdpPercap /
                          media_continente) %>%
          arrange(continent, desc(relativa))
```

O `group_by` no `dplyr`

Exemplo: **select**, **group_by** e **slice_max** para mostrar os países mais ricos de cada continente

O group_by no dplyr

Exemplo: **select**, **group_by** e **slice_max** para mostrar os países mais ricos de cada continente

```
renda.relativa.df %>%
  select(continent, country, gdpPercap, relativa) %>%
    group_by(continent) %>%  slice_max(relativa, n = 1)
```

O group_by no dplyr

Exemplo: **select**, **group_by** e **slice_max** para mostrar os países mais ricos de cada continente

```
renda.relativa.df %>%
  select(continent, country, gdpPercap, relativa) %>%
    group_by(continent) %>%  slice_max(relativa, n = 1)
```

```
## # A tibble: 5 x 4
## # Groups:   continent [5]
##   continent country       gdpPercap relativa
##   <fct>     <fct>             <dbl>    <dbl>
## 1 Africa    Gabon            13206.     4.28
## 2 Americas  United States    42952.     3.90
## 3 Asia      Kuwait           47307.     3.79
## 4 Europe    Norway           49357.     1.97
## 5 Oceania   Australia        34435.     1.16
```

O `group_by` no `dplyr`

Exemplo: **select**, **group_by**, **slice_min** e **arrange** para mostrar os **dois** países mais pobres de cada continente ordenados pela renda relativa

O group_by no dplyr

Exemplo: **select**, **group_by**, **slice_min** e **arrange** para mostrar os **dois** países mais pobres de cada continente ordenados pela renda relativa

```
renda.relativa.df %>%
  select(continent, country, gdpPercap, relativa) %>%
    group_by(continent) %>%
      slice_min(relativa, n = 2) %>% arrange(relativa)
```

O group_by no dplyr

Exemplo: **select**, **group_by**, **slice_min** e **arrange** para mostrar os **dois** países mais pobres de cada continente ordenados pela renda relativa

```
renda.relativa.df %>%
  select(continent, country, gdpPercap, relativa) %>%
    group_by(continent) %>%
      slice_min(relativa, n = 2) %>% arrange(relativa)
```

```
## # A tibble: 10 x 4
## # Groups:   continent [5]
##    continent country                 gdpPercap relativa
##    <fct>     <fct>                       <dbl>    <dbl>
##  1 Asia      Myanmar                      944    0.0757
##  2 Asia      Afghanistan                  975.   0.0781
##  3 Africa    Congo, Dem. Rep.             278.   0.0899
##  4 Americas  Haiti                       1202.   0.109
##  5 Africa    Liberia                      415.   0.134
##  6 Europe    Albania                     5937.   0.237
##  7 Americas  Nicaragua                   2749.   0.250
##  8 Europe    Bosnia and Herzegovina      7446.   0.297
##  9 Oceania   New Zealand                25185.   0.845
## 10 Oceania   Australia                  34435.   1.16
```

O `summarise` no `dplyr`

Exemplo: use **filter** e **summarise** para obter o número de países, a renda média e os percentis 25, 50 e 75 em 2007

O summarise no dplyr

Exemplo: use **filter** e **summarise** para obter o número de países, a renda média e os percentis 25, 50 e 75 em 2007

```
summarise.exemplo <-
  gapminder.df %>%
    filter(year == 2007) %>%
      summarise(n_paises = n(),
                renda.media = mean(gdpPercap),
                renda.p25 = quantile(gdpPercap,
                                     probs = 0.25),
                renda.50 = median(gdpPercap),
                renda.p75 = quantile(gdpPercap,
                                     probs = 0.75))
```

O summarise no dplyr

```
print(summarise.exemplo)
```

```
## # A tibble: 1 x 5
##   n_paises renda.media renda.p25 renda.50 renda.p75
##      <int>       <dbl>     <dbl>    <dbl>     <dbl>
## 1      142      11680.     1625.    6124.    18009.
```

O `summarise` no `dplyr`

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

O summarise no dplyr

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

```
expvida <- gapminder.df %>%
             filter(year == 2007) %>%
                group_by(continent) %>%
                  summarise(n = n(),
                            media = mean(lifeExp),
                            min = min(lifeExp),
                            max = max(lifeExp))
```

O `summarise` no `dplyr`

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

O summarise no dplyr

Exemplo: use **filter**, **group_by** e **summarise** para obter o número de países, a expectativa de vida média e os valores máximo e mínimo por continente em 2007

```
print(expvida)
```

```
## # A tibble: 5 x 5
##   continent     n media   min   max
##   <fct>     <int> <dbl> <dbl> <dbl>
## 1 Africa       52  54.8  39.6  76.4
## 2 Americas     25  73.6  60.9  80.7
## 3 Asia         33  70.7  43.8  82.6
## 4 Europe       30  77.6  71.8  81.8
## 5 Oceania       2  80.7  80.2  81.2
```

Mutating joins no dplyr

Exemplo: junte as informações sobre o clima com o data frame de vôos

```
voos.join.df <- clima.df %>%
                  select(time_hour, origin,
                         visib, precip, temp) %>%
                  right_join(y = voos.df,
                             by = c('time_hour',
                                    'origin'))
```

Compare `voos.join.df` com `voos.df`: o número de linhas mudou? E se substituirmos o `right_join` por `inner_join`, `full_join` e `left_join`?

Outras funções úteis do `dplyr`

recode()

if_else()

case_when()

rename()

Mais informações em https://dplyr.tidyverse.org/reference/

Recapitulação

Introdução

Mais funções no R

Estatísticas descritivas