Aula 3 - Pacote do R: dplyr

Patricia Kuyven

06/09/2018

## Pacote para manipulação de dados: dplyr

Este é um dos principais pacotes encarregados da tarefa de estruturar os dados. Instale e carregue os pacotes utilizando:

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

#### Vamos trabalhar aqui com a base mtcars que já vem no R, que já utilizamos na aula passada. E também com a base de evasão escolar utilizada na disciplina de estatística.

mtcars

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

library(readxl)  
dados\_evasao <- read\_excel("~/GitHub/GeneralRepositoriesUnisinos/PosUnisinosIntroducaoPythonR/base\_dados\_evasao\_escolar.xlsx")  
dados\_evasao

## # A tibble: 370 x 13  
## Curso Disciplina `Grau de exigênc~ `Modalidade da ~ `Média de notas ~  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Admin~ Estratégia~ 1 2 4.5  
## 2 Gestã~ Matemática~ 2 1 4.6  
## 3 Gestã~ Métodos Es~ 3 2 4.8  
## 4 Gestã~ Matemática~ 2 2 4.9  
## 5 Gestã~ Métodos Es~ 3 1 5.3  
## 6 Gestã~ Matemática~ 2 2 5.4  
## 7 Gestã~ Métodos Es~ 3 1 5.4  
## 8 Engen~ Métodos Es~ 3 1 5.5  
## 9 Admin~ Cálculo A 3 1 5.5  
## 10 Admin~ Matemática~ 2 1 5.6  
## # ... with 360 more rows, and 8 more variables: `número de disciplinas  
## # evadidas em outros semestres` <dbl>, `Número de ocorrências de  
## # mensalidades pagas com atraso` <dbl>, Sexo <chr>, Idade <dbl>,  
## # `Distância endereço res. do aluno até campus` <dbl>, `Semestre em que  
## # ocorreu a disciplina (2015\_1 é 1; 2015\_2 é 2; 2016\_1 é 3...)` <dbl>,  
## # `Número de alunos na turma` <dbl>, `Situação final da  
## # disciplina` <dbl>

### Função select

A função select() seleciona colunas (variáveis). É possível utilizar nomes, índices, intervalos de variáveis ou utilizar as funções starts\_with(x), contains(x), matches(x), one\_of(x) para selecionar as variáveis.

dados\_evasao %>%  
 select(Disciplina, 'Modalidade da disciplina', 'Situação final da disciplina')

## # A tibble: 370 x 3  
## Disciplina `Modalidade da disciplina` `Situação final da~  
## <chr> <dbl> <dbl>  
## 1 Estratégias de marketing 2 1  
## 2 Matemática financeira 1 1  
## 3 Métodos Estatísticos 2 1  
## 4 Matemática financeira 2 1  
## 5 Métodos Estatísticos 1 NA  
## 6 Matemática financeira 2 1  
## 7 Métodos Estatísticos 1 1  
## 8 Métodos Estatísticos 1 1  
## 9 Cálculo A 1 0  
## 10 Matemática financeira 1 1  
## # ... with 360 more rows

dados\_evasao %>%  
 select(Sexo:'Distância endereço res. do aluno até campus','Situação final da disciplina')

## # A tibble: 370 x 4  
## Sexo Idade `Distância endereço res. do aluno~ `Situação final da disc~  
## <chr> <dbl> <dbl> <dbl>  
## 1 F 24 1 1  
## 2 F 25 1 1  
## 3 M 22 3 1  
## 4 M 26 2 1  
## 5 F 18 3 NA  
## 6 M 20 1 1  
## 7 F 32 3 1  
## 8 F 21 2 1  
## 9 F 28 1 0  
## 10 M 25 1 1  
## # ... with 360 more rows

### Função filter

A função filter() filtra/seleciona linhas.

dados\_evasao %>%  
 filter(Disciplina=='Matemática financeira')

## # A tibble: 53 x 13  
## Curso Disciplina `Grau de exigênc~ `Modalidade da ~ `Média de notas ~  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Gestã~ Matemática~ 2 1 4.6  
## 2 Gestã~ Matemática~ 2 2 4.9  
## 3 Gestã~ Matemática~ 2 2 5.4  
## 4 Admin~ Matemática~ 2 1 5.6  
## 5 Admin~ Matemática~ 2 1 5.8  
## 6 Gestã~ Matemática~ 2 2 5.8  
## 7 Gestã~ Matemática~ 2 1 5.9  
## 8 Admin~ Matemática~ 2 1 6.3  
## 9 Admin~ Matemática~ 2 1 6.4  
## 10 Gestã~ Matemática~ 2 2 6.4  
## # ... with 43 more rows, and 8 more variables: `número de disciplinas  
## # evadidas em outros semestres` <dbl>, `Número de ocorrências de  
## # mensalidades pagas com atraso` <dbl>, Sexo <chr>, Idade <dbl>,  
## # `Distância endereço res. do aluno até campus` <dbl>, `Semestre em que  
## # ocorreu a disciplina (2015\_1 é 1; 2015\_2 é 2; 2016\_1 é 3...)` <dbl>,  
## # `Número de alunos na turma` <dbl>, `Situação final da  
## # disciplina` <dbl>

Se quiser aplicar as duas funções anteriores, pode ser usado o operador lógico &.

dados\_evasao %>%  
 select(Disciplina, Sexo, `Situação final da disciplina`) %>%  
 filter(Disciplina=='Matemática financeira')

## # A tibble: 53 x 3  
## Disciplina Sexo `Situação final da disciplina`  
## <chr> <chr> <dbl>  
## 1 Matemática financeira F 1  
## 2 Matemática financeira M 1  
## 3 Matemática financeira M 1  
## 4 Matemática financeira M 1  
## 5 Matemática financeira F 0  
## 6 Matemática financeira F 1  
## 7 Matemática financeira M 1  
## 8 Matemática financeira M 1  
## 9 Matemática financeira M 0  
## 10 Matemática financeira M 0  
## # ... with 43 more rows

### Função mutate

A função mutate() cria ou modifica colunas. Ela é equivalente à função transform(), mas aceita várias novas colunas iterativamente. Novas variáveis devem ter o mesmo número de linhas da base original (ou terem comprimento 1).

mtcars\_modif <- mtcars %>%  
 mutate(nova\_variavel = mpg \* hp)  
mtcars\_modif

## mpg cyl disp hp drat wt qsec vs am gear carb nova\_variavel  
## 1 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 2310.0  
## 2 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 2310.0  
## 3 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 2120.4  
## 4 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 2354.0  
## 5 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 3272.5  
## 6 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 1900.5  
## 7 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 3503.5  
## 8 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 1512.8  
## 9 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 2166.0  
## 10 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 2361.6  
## 11 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 2189.4  
## 12 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 2952.0  
## 13 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 3114.0  
## 14 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 2736.0  
## 15 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 2132.0  
## 16 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 2236.0  
## 17 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 3381.0  
## 18 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 2138.4  
## 19 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 1580.8  
## 20 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 2203.5  
## 21 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 2085.5  
## 22 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 2325.0  
## 23 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 2280.0  
## 24 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 3258.5  
## 25 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 3360.0  
## 26 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 1801.8  
## 27 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 2366.0  
## 28 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 3435.2  
## 29 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 4171.2  
## 30 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 3447.5  
## 31 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 5025.0  
## 32 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 2332.6

### Função arrange

Esta função ordena a base. Geralmente utilizada com outras funções. Pode ser usado o argumento desc= para colocar em ordem decrescente.

dados\_evasao %>%  
 select(Disciplina, Idade) %>%  
 filter(Disciplina == 'Matemática financeira') %>%   
 arrange(desc(Idade))

## # A tibble: 53 x 2  
## Disciplina Idade  
## <chr> <dbl>  
## 1 Matemática financeira 32  
## 2 Matemática financeira 30  
## 3 Matemática financeira 30  
## 4 Matemática financeira 29  
## 5 Matemática financeira 29  
## 6 Matemática financeira 29  
## 7 Matemática financeira 28  
## 8 Matemática financeira 27  
## 9 Matemática financeira 27  
## 10 Matemática financeira 27  
## # ... with 43 more rows