Universidade Federal do Ceará (UFC/Sobral)

Aula 10 - Métodos Computacionais Aplicados

Prof. Weligton Gomes

16/10/2023

```
library(ISwR)

data(energy)
data(thuesen)
```

Funções que facilitam a manipulação de dados no R

```
thue1 <- subset(thuesen,blood.glucose<7)
thue1
```

```
##    blood.glucose short.velocity
## 6            5.3           1.49
## 11           6.7           1.25
## 12           5.2           1.19
## 15           6.7           1.52
## 17           4.2           1.12
## 22           4.9           1.03
```

```
thue2 <- transform(thuesen,log.gluc=log(blood.glucose))
thue2
```

```
##    blood.glucose short.velocity log.gluc
## 1           15.3           1.76 2.727853
## 2           10.8           1.34 2.379546
## 3            8.1           1.27 2.091864
## 4           19.5           1.47 2.970414
## 5            7.2           1.27 1.974081
## 6            5.3           1.49 1.667707
## 7            9.3           1.31 2.230014
## 8           11.1           1.09 2.406945
## 9            7.5           1.18 2.014903
## 10          12.2           1.22 2.501436
## 11           6.7           1.25 1.902108
## 12           5.2           1.19 1.648659
## 13          19.0           1.95 2.944439
## 14          15.1           1.28 2.714695
## 15           6.7           1.52 1.902108
## 16           8.6             NA 2.151762
## 17           4.2           1.12 1.435085
## 18          10.3           1.37 2.332144
## 19          12.5           1.19 2.525729
```

```
## 20          16.1           1.05 2.778819
## 21          13.3           1.32 2.587764
## 22           4.9           1.03 1.589235
## 23           8.8           1.12 2.174752
## 24           9.5           1.70 2.251292
```

```
thue3 <- within(thuesen,{
  log.gluc <- log(blood.glucose)
  m <- mean(log.gluc)
  centered.log.gluc <- log.gluc - m
  rm(m)
})
thue3
```

```
##    blood.glucose short.velocity centered.log.gluc log.gluc
## 1           15.3           1.76       0.481879807 2.727853
## 2           10.8           1.34       0.133573113 2.379546
## 3            8.1           1.27      -0.154108960 2.091864
## 4           19.5           1.47       0.724441444 2.970414
## 5            7.2           1.27      -0.271891996 1.974081
## 6            5.3           1.49      -0.578266201 1.667707
## 7            9.3           1.31      -0.015958621 2.230014
## 8           11.1           1.09       0.160972087 2.406945
## 9            7.5           1.18      -0.231070001 2.014903
## 10          12.2           1.22       0.255462930 2.501436
## 11           6.7           1.25      -0.343865495 1.902108
## 12           5.2           1.19      -0.597314396 1.648659
## 13          19.0           1.95       0.698465958 2.944439
## 14          15.1           1.28       0.468721722 2.714695
## 15           6.7           1.52      -0.343865495 1.902108
## 16           8.6             NA      -0.094210818 2.151762
## 17           4.2           1.12      -0.810888496 1.435085
## 18          10.3           1.37       0.086170874 2.332144
## 19          12.5           1.19       0.279755623 2.525729
## 20          16.1           1.05       0.532846250 2.778819
## 21          13.3           1.32       0.341791014 2.587764
## 22           4.9           1.03      -0.656737817 1.589235
## 23           8.8           1.12      -0.071221300 2.174752
## 24           9.5           1.70       0.005318777 2.251292
```

```
summary(thue3)
```

```
##  blood.glucose    short.velocity  centered.log.gluc     log.gluc
##  Min.   : 4.200   Min.   :1.030   Min.   :-0.81089   Min.   :1.435
##  1st Qu.: 7.075   1st Qu.:1.185   1st Qu.:-0.28989   1st Qu.:1.956
##  Median : 9.400   Median :1.270   Median :-0.00532   Median :2.241
##  Mean   :10.300   Mean   :1.326   Mean   : 0.00000   Mean   :2.246
##  3rd Qu.:12.700   3rd Qu.:1.420   3rd Qu.: 0.29526   3rd Qu.:2.541
##  Max.   :19.500   Max.   :1.950   Max.   : 0.72444   Max.   :2.970
##                   NA's   :1
```

o R é uma linguagem que permite criar novas funções. Na verdade, muitas das funções em R são atualmente funções de funções. A estrutura de uma função é dada abaixo:

myfunction<-function(arg1, arg2,...){ statements ou afirmações return(object) }

O R é uma verdadeira linguagem de programação que permite a execução condicional e também construções

de loop. Por exemplo:

```
for(i in 1:5){
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

```
# criando vetor de exemplo
x <- 10:20

# divide cada elemento por 10
for(i in seq_along(x))
  x[i] <- x[i]/10

# resultado
x
```

```
##  [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
```

```
y <- 12345
x <- y/2
while (abs(x*x-y) > 1e-10) x <- (x + y/x)/2
x
```

```
## [1] 111.1081
```

```
x^2
```

```
## [1] 12345
```

Observe a construção da expressão while (condição), que diz que a expressão deve ser avaliada enquanto a condição for TRUE.

O teste ocorre na parte superior do loop

Uma variação do mesmo algoritmo com o teste na parte inferior do loop pode ser escrita com uma construção repetida:

```
y<-12345
x <- y/2
repeat{
  x <- (x + y/x)/2
  if (abs(x*x-y) < 1e-10) break
}
x
```

```
## [1] 111.1081
```

```
x^2
```

```
## [1] 12345
```

Progressão Aritmética (PA) e Progressão Geométrica (PG)

Criando funções para calcular os n-ésimos termos e a soma de uma PA:

```
nesimotermo<-function(a1, n, r){
  a1+(n-1)*r
}

nesimotermo(107,101,6)
```

```
## [1] 707
```

```
termonpa<-function(a1, an, r){
  ((an-a1)/r)+1
}
termonpa(2,100,2)
```

```
## [1] 50
```

```
somapa<-function(a1, n, r){
  an=a1+(n-1)*r
  ((a1+an)*n)/2
}

somapa(2,5,2)
```

```
## [1] 30
```

Progressão Geométrica (PG)

Criando funções para calcular os n-ésimos termos e a soma de uma PG finita e infinita:

PG finita

```
nesimotermopg<-function(a1,q,n){
  a1*q^(n-1)
}

nesimotermopg(3,2,4)
```

```
## [1] 24
```

```
somapg<-function(a1, q, n){
  (a1*(q^n -1))/(q-1)
}

somapg(1,10,5)
```

```
## [1] 11111
```

PG infinita

```
sompginf<-function(a1, q){
  a1/(1-q)
}

sompginf(1, 0.5)
```

```
## [1] 2
```

Exercícios de P.A e P.G

Questão 01: Qual é o centésimo primeiro termo de uma P.A. cujo primeiro termo é 107 e a razão é 6?

```
nesimotermo<-function(a1, n, r){
  a1+(n-1)*r
}

nesimotermo(107,101,6)
```

```
## [1] 707
```

Questão 02: Sabendo que o primeiro termo é 10, o último é 109 e a razão é 3, basta usar a fórmula do termo geral para encontrar a posição do termo 109.

```
termonpa<-function(a1, an, r){
  ((an-a1)/r)+1
}
termonpa(10,109,3)
```

```
## [1] 34
```

Questão 03: Determine o décimo quinto termo da progressão geométrica a seguir: (1, 3, 9, 27, . . .).

```
nesimotermopg<-function(a1,q,n){
  a1*q^(n-1)
}

nesimotermopg(1,3,15)
```

```
## [1] 4782969
```

Transformar a temperatura em grau Fahrenheit para grau Celsius.

Lembre-se que em grau Fahrenheit a escala de temperatura varia de $32°F$ $(0°C)$ até $212°F$ $(100°C)$.

```
fahrenheit_to_celsius <- function(temp_F) {
  temp_C <- (temp_F - 32) * 5 / 9
  return(temp_C)
}

fahrenheit_to_celsius(55)
```

```
## [1] 12.77778
```

```
fahrenheit_to_celsius(212)
```

```
## [1] 100
```

Entrada de Dados no R.

```
require(foreign)
```

```
## Loading required package: foreign
```

```
A<-read.table("/Users/weligtongomes/data.txt",header = F,sep = ",",col.names = c("ano","x","y"))
```

```
## Warning in read.table("/Users/weligtongomes/data.txt", header = F, sep = ",", :
## incomplete final line found by readTableHeader on
```

```
## '/Users/weligtongomes/data.txt'
```

```
A$y[1]
```

```
## [1] 4
```

```
summary(A)
```

```
##       ano             x               y
##  Min.   :1997   Min.   :1.100   Min.   : 2.0
##  1st Qu.:1998   1st Qu.:1.550   1st Qu.: 3.5
##  Median :1998   Median :2.400   Median : 8.5
##  Mean   :1998   Mean   :3.275   Mean   : 9.5
##  3rd Qu.:1999   3rd Qu.:4.125   3rd Qu.:14.5
##  Max.   :2000   Max.   :7.200   Max.   :19.0
```

```
# Importação sem o nome das colunas e insersão posterior
B<-read.table("/Users/weligtongomes/data.txt",header = F,sep = ",")
```

```
## Warning in read.table("/Users/weligtongomes/data.txt", header = F, sep = ","):
## incomplete final line found by readTableHeader on
## '/Users/weligtongomes/data.txt'
```

```
colnames(B)<-c("ano","x","y")
B
```

```
##    ano   x  y
## 1 1997 3.1  4
## 2 1998 7.2 19
## 3 1999 1.7  2
## 4 2000 1.1 13
```

Observação: No Windows o endereço do arquivo a \ deve ser substituida por /.

Inserindo dados com o Data Entry

Arquivo no formato .txt (Bloco de Notas)

```
x401k <- read.delim("~/Base de Dados_MCA/401k.txt")
```

Arquivo no formato .csv (Separador de Vírgula)

```
nerlove <- read.csv2("~/Base de Dados_MCA/nerlove.csv")
```

Arquivo no formato .xlsx (Excel)

```
library(xlsx)
```

```
rental1 <- read.xlsx("~/Base de Dados_MCA/rental1.xlsx", 1)
```

Arquivo no formato .dta (Stata)

```
library(haven)
```

```
# mydata <- read.dta("c:/mydata.dta")

campus <- read_dta("~/Base de Dados_MCA/campus.dta")
```

Manipulação de Bases de Dados

```
library(readxl)
```

```
rental1 <- read_excel("~/Base de Dados_MCA/rental1.xlsx")

rental2 <- read_excel("~/Base de Dados_MCA/rental2.xlsx")

rental3 <- read_excel("~/Base de Dados_MCA/rental3.xlsx")

rental4 <- read_excel("~/Base de Dados_MCA/rental4.xlsx")
```

A Função Merge

```
# mydata<-merge(mydata1, mydata2, by=c("id_1","id_2"))

rental12<-merge(rental1, rental2, by=c("city","year"))
```

A Função Append

```
rental34<-rbind(rental3, rental4)
```

O pacote dplyr

O dplyr é o pacote mais útil para realizar transformação de dados, aliando simplicidade e eficiência de uma forma elegante. Os scripts em R que fazem uso inteligente dos verbos dplyr e as facilidades do operador pipe (%>%)(atalho Ctrl + Shift + M) tendem a ficar mais legíveis e organizados, sem perder velocidade de execução. (https://livro.curso-r.com/7-2-dplyr.html)

Principais funções do dplyr são:

select() - seleciona colunas; arrange() - ordena a base; filter() - filtra linhas; mutate() - cria/modifica colunas; group_by() - agrupa a base; summarise() - sumariza a base.

```
#install.packages("dplyr")
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
```

```
library(readr)
```

```
imdb <- readRDS("~/imdb.rds")
View(imdb)
```

Selecionando colunas

Para selecionar colunas, utilizamos a função select().

```
select(imdb, titulo)
```

```
## # A tibble: 11,340 x 1
##    titulo
##    <chr>
##  1 Broadcast News
##  2 Murder, He Says
##  3 Me, Myself & Irene
##  4 Never Give a Sucker an Even Break
##  5 Adam & Steve
##  6 Henry Gamble's Birthday Party
##  7 No Way Out
##  8 Welcome Home, Roscoe Jenkins
##  9 Some Kind of Wonderful
## 10 The Family That Preys
## # i 11,330 more rows
```

Também podemos selecionar várias colunas.

```
select(imdb, titulo, ano, orcamento)
```

```
## # A tibble: 11,340 x 3
##    titulo                              ano orcamento
##    <chr>                             <dbl>     <dbl>
##  1 Broadcast News                     1987  20000000
##  2 Murder, He Says                    1945        NA
##  3 Me, Myself & Irene                 2000  51000000
##  4 Never Give a Sucker an Even Break  1941        NA
##  5 Adam & Steve                       2005        NA
##  6 Henry Gamble's Birthday Party      2015        NA
##  7 No Way Out                         1987  15000000
##  8 Welcome Home, Roscoe Jenkins       2008  35000000
##  9 Some Kind of Wonderful             1987        NA
## 10 The Family That Preys              2008        NA
## # i 11,330 more rows
```

```
select(imdb, titulo:generos)
```

```
## # A tibble: 11,340 x 4
##    titulo                              ano data_lancamento generos
##    <chr>                             <dbl> <chr>           <chr>
##  1 Broadcast News                     1987 1988-04-01      Comedy, Drama, Roman~
##  2 Murder, He Says                    1945 1945-06-23      Comedy, Crime, Myste~
##  3 Me, Myself & Irene                 2000 2000-09-08      Comedy
##  4 Never Give a Sucker an Even Break  1941 1947-05-02      Comedy, Musical
##  5 Adam & Steve                       2005 2007-05-17      Comedy, Drama, Music
##  6 Henry Gamble's Birthday Party      2015 2016-01-08      Drama
##  7 No Way Out                         1987 1987-12-11      Action, Crime, Drama
```

```
##  8 Welcome Home, Roscoe Jenkins       2008 2008-02-08      Comedy, Romance
##  9 Some Kind of Wonderful             1987 1988-01-13      Drama, Romance
## 10 The Family That Preys              2008 2008-09-12      Drama
## # i 11,330 more rows
```

```
select(imdb, starts_with("num"))
```

```
## # A tibble: 11,340 x 3
##    num_avaliacoes num_criticas_publico num_criticas_critica
##             <dbl>                <dbl>                <dbl>
##  1          26257                  142                   62
##  2           1639                   35                   10
##  3         219069                  502                  161
##  4           2108                   35                   18
##  5           2953                   48                   15
##  6           2364                   26                   14
##  7          34513                  125                   72
##  8          13315                   45                   74
##  9          27065                  145                   55
## 10           6703                   52                   29
## # i 11,330 more rows
```

```
select(imdb, -ano, -direcao)
```

```
## # A tibble: 11,340 x 18
##    id_filme  titulo       data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>        <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt0092699 Broadcast N~ 1988-04-01      Comedy~     133 USA   Engli~  20000000
##  2 tt0037931 Murder, He ~ 1945-06-23      Comedy~      91 USA   Engli~        NA
##  3 tt0183505 Me, Myself ~ 2000-09-08      Comedy      116 USA   Engli~  51000000
##  4 tt0033945 Never Give ~ 1947-05-02      Comedy~      71 USA   Engli~        NA
##  5 tt0372122 Adam & Steve 2007-05-17      Comedy~      99 USA   Engli~        NA
##  6 tt3703836 Henry Gambl~ 2016-01-08      Drama        87 USA   Engli~        NA
##  7 tt0093640 No Way Out   1987-12-11      Action~     114 USA   Engli~  15000000
##  8 tt0494652 Welcome Hom~ 2008-02-08      Comedy~     104 USA   Engli~  35000000
##  9 tt0094006 Some Kind o~ 1988-01-13      Drama,~      95 USA   Engli~        NA
## 10 tt1142798 The Family ~ 2008-09-12      Drama       111 USA   Engli~        NA
## # i 11,330 more rows
## # i 10 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, roteiro <chr>, producao <chr>, elenco <chr>,
## #   descricao <chr>, num_criticas_publico <dbl>, num_criticas_critica <dbl>
```

```
select(imdb, -starts_with("num"))
```

```
## # A tibble: 11,340 x 17
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt0092699 Broad~  1987 1988-04-01      Comedy~     133 USA   Engli~  20000000
##  2 tt0037931 Murde~  1945 1945-06-23      Comedy~      91 USA   Engli~        NA
##  3 tt0183505 Me, M~  2000 2000-09-08      Comedy      116 USA   Engli~  51000000
##  4 tt0033945 Never~  1941 1947-05-02      Comedy~      71 USA   Engli~        NA
##  5 tt0372122 Adam ~  2005 2007-05-17      Comedy~      99 USA   Engli~        NA
##  6 tt3703836 Henry~  2015 2016-01-08      Drama        87 USA   Engli~        NA
##  7 tt0093640 No Wa~  1987 1987-12-11      Action~     114 USA   Engli~  15000000
##  8 tt0494652 Welco~  2008 2008-02-08      Comedy~     104 USA   Engli~  35000000
##  9 tt0094006 Some ~  1987 1988-01-13      Drama,~      95 USA   Engli~        NA
```

```
## 10 tt1142798 The F~  2008 2008-09-12      Drama       111 USA   Engli~        NA
## # i 11,330 more rows
## # i 8 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   direcao <chr>, roteiro <chr>, producao <chr>, elenco <chr>, descricao <chr>
```

Ordenando a base

```
arrange(imdb, orcamento)
```

```
## # A tibble: 11,340 x 20
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt5345298 Patie~  2016 2016-10-11      Horror      116 USA   Icela~         0
##  2 tt7692822 Driven  2019 2019-02-09      Comedy~      90 USA   Engli~         0
##  3 tt3748918 To Yo~  2019 2020-03-17      Animat~      91 USA   Engli~         1
##  4 tt8196068 Twist~  2018 2018-10-03      Drama,~      89 USA   Engli~         3
##  5 tt0772152 Amate~  2006 2006-07-30      Crime,~      71 USA   Engli~        45
##  6 tt1260680 Pathf~  2011 2011-01-11      Action~     100 USA   Engli~        50
##  7 tt1701224 My Na~  2012 2012-10-19      Crime,~      90 USA   Frenc~       300
##  8 tt0054880 Flami~  1963 1963-04-29      Comedy~      45 USA   Engli~       300
##  9 tt1980185 Memor~  2012 2014-03-10      Crime,~      70 USA   Engli~       300
## 10 tt5009236 King ~  2015 2015-03-27      Biogra~      46 USA   Engli~       500
## # i 11,330 more rows
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

```
arrange(imdb, desc(orcamento)) #desc ordena de forma decrescente.
```

```
## # A tibble: 11,340 x 20
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt4154796 Aveng~  2019 2019-04-24      Action~     181 USA   Engli~ 356000000
##  2 tt4154756 Aveng~  2018 2018-04-25      Action~     149 USA   Engli~ 321000000
##  3 tt2527336 Star ~  2017 2017-12-13      Action~     152 USA   Engli~ 317000000
##  4 tt0449088 Pirat~  2007 2007-05-23      Action~     169 USA   Engli~ 300000000
##  5 tt2527338 Star ~  2019 2019-12-18      Action~     141 USA   Engli~ 275000000
##  6 tt3778644 Solo:~  2018 2018-05-23      Action~     135 USA   Engli~ 275000000
##  7 tt0348150 Super~  2006 2006-09-01      Action~     154 USA   Engli~ 270000000
##  8 tt0398286 Tangl~  2010 2010-11-26      Animat~     100 USA   Engli~ 260000000
##  9 tt0413300 Spide~  2007 2007-05-01      Action~     139 USA   Engli~ 258000000
## 10 tt2975590 Batma~  2016 2016-03-23      Action~     152 USA   Engli~ 250000000
## # i 11,330 more rows
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

```
arrange(imdb, desc(ano), desc(orcamento))  #ordena segundo duas ou mais colunas
```

```
## # A tibble: 11,340 x 20
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt6587640 Troll~  2020 2020-04-10      Animat~      90 USA   Engli~  90000000
```

```
##  2 tt7713068 Birds~  2020 2020-02-06      Action~     109 USA   Engli~  84500000
##  3 tt5774060 Under~  2020 2020-01-30      Action~      95 USA   Engli~  80000000
##  4 tt6820324 Timmy~  2020 2020-03-24      Advent~      99 USA   Engli~  45000000
##  5 tt1634106 Blood~  2020 2020-03-27      Action~     109 USA   Engli~  45000000
##  6 tt100595~ Unhin~  2020 2020-09-24      Action~      90 USA   Engli~  33000000
##  7 tt8461224 The T~  2020 2020-08-07      Action~      95 USA   Engli~  30000000
##  8 tt103089~ Force~  2020 2020-06-30      Action~      91 USA   Engli~  23000000
##  9 tt4411584 The S~  2020 2020-07-31      Drama,~     107 USA   Engli~  21000000
## 10 tt8244784 The H~  2020 2020-03-24      Action~      90 USA   Engli~  14000000
## # i 11,330 more rows
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

O pipe em ação

Podemos aplicar mais de uma função de manipulação em uma base para obtermos a tabela que desejamos. Poderíamos, por exemplo, querer uma tabela apenas com o título e ano dos filmes, ordenada de forma crescente de lançamento. Para fazer isso, poderíamos aninhar as funções.

```
arrange(select(imdb, titulo, ano), ano)
```

```
## # A tibble: 11,340 x 2
##    titulo                                          ano
##    <chr>                                         <dbl>
##  1 Tillie's Punctured Romance                     1914
##  2 Judith of Bethulia                             1914
##  3 The Avenging Conscience: or 'Thou Shalt Not Kill'  1914
##  4 The Regeneration                               1915
##  5 The Cheat                                      1915
##  6 The Birth of a Nation                          1915
##  7 Intolerance: Love's Struggle Throughout the Ages   1916
##  8 20,000 Leagues Under the Sea                   1916
##  9 The Poor Little Rich Girl                      1917
## 10 Shoulder Arms                                  1918
## # i 11,330 more rows
```

Alguns códigos códigos funcionam e levam ao mesmo resultado, mas não são muito bons ou intuitivos. Para isso, a solução para aplicar diversas operações de manipulação em uma base de dados seria por meio da aplicação do operador pipe: %>%.

```
imdb %>%
  select(titulo, ano) %>%
  arrange(ano)
```

```
## # A tibble: 11,340 x 2
##    titulo                                          ano
##    <chr>                                         <dbl>
##  1 Tillie's Punctured Romance                     1914
##  2 Judith of Bethulia                             1914
##  3 The Avenging Conscience: or 'Thou Shalt Not Kill'  1914
##  4 The Regeneration                               1915
##  5 The Cheat                                      1915
##  6 The Birth of a Nation                          1915
##  7 Intolerance: Love's Struggle Throughout the Ages   1916
```

```
##  8 20,000 Leagues Under the Sea                        1916
##  9 The Poor Little Rich Girl                           1917
## 10 Shoulder Arms                                       1918
## # i 11,330 more rows
```

Filtrando linhas

Para filtrar valores de uma coluna da base, utilizamos a função filter().

```
imdb %>%
  filter(nota_imdb > 9)
```

```
## # A tibble: 5 x 20
##   id_filme   titulo   ano data_lancamento generos duracao pais  idioma orcamento
##   <chr>      <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
## 1 tt10218912 As I ~  2019 2019-12-06      Drama,~      62 USA   Engli~     10000
## 2 tt6735740  Love ~  2019 2019-06-23      Comedy      100 USA   Engli~   3000000
## 3 tt0111161  The S~  1994 1995-02-10      Drama       142 USA   Engli~  25000000
## 4 tt0068646  The G~  1972 1972-09-21      Crime,~     175 USA   Engli~   6000000
## 5 tt5980638  The T~  2018 2020-06-19      Music,~      96 USA   Engli~     90000
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

Podemos selecionar apenas as colunas título e nota para visualizarmos as notas:

```
imdb %>%
  filter(nota_imdb > 9) %>%
  select(titulo, nota_imdb)
```

```
## # A tibble: 5 x 2
##   titulo                   nota_imdb
##   <chr>                        <dbl>
## 1 As I Am                        9.3
## 2 Love in Kilnerry               9.3
## 3 The Shawshank Redemption       9.3
## 4 The Godfather                  9.2
## 5 The Transcendents              9.2
```

Podemos estender o filtro para duas ou mais colunas. Para isso, separamos cada operação por uma vírgula.

```
imdb %>%
  filter(ano > 2010, nota_imdb > 8.5)
```

```
## # A tibble: 8 x 20
##   id_filme   titulo   ano data_lancamento generos duracao pais  idioma orcamento
##   <chr>      <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
## 1 tt10218912 As I ~  2019 2019-12-06      Drama,~      62 USA   Engli~     10000
## 2 tt8503618  Hamil~  2020 2020-07-03      Biogra~     160 USA   Engli~        NA
## 3 tt6735740  Love ~  2019 2019-06-23      Comedy      100 USA   Engli~   3000000
## 4 tt10765852 Metal~  2019 2019-10-18      Music       150 USA   Engli~        NA
## 5 tt6019206  Kill ~  2011 2011-03-27      Action~     247 USA   <NA>          NA
## 6 tt8241876  5th B~  2020 2020-06-03      Crime        95 USA   Engli~        NA
## 7 tt2170667  Wheels  2014 2017-02-01      Drama       115 USA   Engli~        NA
## 8 tt5980638  The T~  2018 2020-06-19      Music,~      96 USA   Engli~     90000
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
```

```
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

Também podemos fazer operações com as colunas da base dentro da função filter. O código abaixo devolve uma tabela apenas com os filmes que lucraram.

```
imdb %>%
  filter(receita - orcamento > 0)
```

```
## # A tibble: 2,541 x 20
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt0092699 Broad~  1987 1988-04-01      Comedy~     133 USA   Engli~  20000000
##  2 tt0183505 Me, M~  2000 2000-09-08      Comedy      116 USA   Engli~  51000000
##  3 tt0093640 No Wa~  1987 1987-12-11      Action~     114 USA   Engli~  15000000
##  4 tt0494652 Welco~  2008 2008-02-08      Comedy~     104 USA   Engli~  35000000
##  5 tt1488555 The C~  2011 2011-12-09      Comedy~     112 USA   Engli~  52000000
##  6 tt0090022 Silve~  1985 1986-01-16      Action~     133 USA   Engli~  26000000
##  7 tt0120434 Vegas~  1997 1997-02-14      Comedy       93 USA   Engli~  25000000
##  8 tt1086772 Blend~  2014 2014-07-02      Comedy~     117 USA   Engli~  40000000
##  9 tt0064115 Butch~  1969 1969-09-26      Biogra~     110 USA   Engli~   6000000
## 10 tt0441796 Stay ~  2006 2006-03-24      Fantas~      85 USA   Engli~   7000000
## # i 2,531 more rows
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

Também podemos filtrar colunas categóricas. O exemplo abaixo retorna uma tabela apenas com os filmes dirigidos por Quentin Tarantino ou Steven Spielberg.

```
imdb %>%
  filter(direcao %in% c("Quentin Tarantino", "Steven Spielberg"))
```

```
## # A tibble: 30 x 20
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt0102057 Hook    1991 1992-03-27      Advent~     142 USA   Engli~  70000000
##  2 tt0118607 Amist~  1997 1998-03-13      Biogra~     155 USA   Engli~  36000000
##  3 tt0096794 Always  1989 1989-12-22      Drama,~     122 USA   Engli~  31000000
##  4 tt0110912 Pulp ~  1994 1994-10-28      Crime,~     154 USA   Engli~   8000000
##  5 tt0407304 War o~  2005 2005-06-29      Advent~     116 USA   Engli~ 132000000
##  6 tt0075860 Close~  1977 1978-03-03      Drama,~     138 USA   Engli~  20000000
##  7 tt0082971 Raide~  1981 1981-06-12      Action~     115 USA   Engli~  18000000
##  8 tt0097576 India~  1989 1989-10-06      Action~     127 USA   Engli~  48000000
##  9 tt0362227 The T~  2004 2004-09-03      Comedy~     128 USA   Engli~  60000000
## 10 tt0120815 Savin~  1998 1998-10-30      Drama,~     169 USA   Engli~  70000000
## # i 20 more rows
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

Modificando e criando novas colunas

Para modificar uma coluna existente ou criar uma nova coluna, utilizamos a função mutate(). O código abaixo divide os valores da coluna duração por 60, mudando a unidade de medida dessa variável de minutos para horas.

```
imdb %>%
  mutate(duracao = duracao/60)
```

```
## # A tibble: 11,340 x 20
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt0092699 Broad~  1987 1988-04-01      Comedy~    2.22 USA   Engli~  20000000
##  2 tt0037931 Murde~  1945 1945-06-23      Comedy~    1.52 USA   Engli~        NA
##  3 tt0183505 Me, M~  2000 2000-09-08      Comedy     1.93 USA   Engli~  51000000
##  4 tt0033945 Never~  1941 1947-05-02      Comedy~    1.18 USA   Engli~        NA
##  5 tt0372122 Adam ~  2005 2007-05-17      Comedy~    1.65 USA   Engli~        NA
##  6 tt3703836 Henry~  2015 2016-01-08      Drama      1.45 USA   Engli~        NA
##  7 tt0093640 No Wa~  1987 1987-12-11      Action~    1.9  USA   Engli~  15000000
##  8 tt0494652 Welco~  2008 2008-02-08      Comedy~    1.73 USA   Engli~  35000000
##  9 tt0094006 Some ~  1987 1988-01-13      Drama,~    1.58 USA   Engli~        NA
## 10 tt1142798 The F~  2008 2008-09-12      Drama      1.85 USA   Engli~        NA
## # i 11,330 more rows
## # i 11 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>
```

Também poderíamos ter criado essa variável em uma nova coluna. Repare que a nova coluna duracao_horas é colocada no final da tabela.

```
imdb %>%
  mutate(duracao_horas = duracao/60)
```

```
## # A tibble: 11,340 x 21
##    id_filme  titulo   ano data_lancamento generos duracao pais  idioma orcamento
##    <chr>     <chr>  <dbl> <chr>           <chr>     <dbl> <chr> <chr>      <dbl>
##  1 tt0092699 Broad~  1987 1988-04-01      Comedy~     133 USA   Engli~  20000000
##  2 tt0037931 Murde~  1945 1945-06-23      Comedy~      91 USA   Engli~        NA
##  3 tt0183505 Me, M~  2000 2000-09-08      Comedy      116 USA   Engli~  51000000
##  4 tt0033945 Never~  1941 1947-05-02      Comedy~      71 USA   Engli~        NA
##  5 tt0372122 Adam ~  2005 2007-05-17      Comedy~      99 USA   Engli~        NA
##  6 tt3703836 Henry~  2015 2016-01-08      Drama        87 USA   Engli~        NA
##  7 tt0093640 No Wa~  1987 1987-12-11      Action~     114 USA   Engli~  15000000
##  8 tt0494652 Welco~  2008 2008-02-08      Comedy~     104 USA   Engli~  35000000
##  9 tt0094006 Some ~  1987 1988-01-13      Drama,~      95 USA   Engli~        NA
## 10 tt1142798 The F~  2008 2008-09-12      Drama       111 USA   Engli~        NA
## # i 11,330 more rows
## # i 12 more variables: receita <dbl>, receita_eua <dbl>, nota_imdb <dbl>,
## #   num_avaliacoes <dbl>, direcao <chr>, roteiro <chr>, producao <chr>,
## #   elenco <chr>, descricao <chr>, num_criticas_publico <dbl>,
## #   num_criticas_critica <dbl>, duracao_horas <dbl>
```

Podemos fazer qualquer operação com uma ou mais colunas. A única regra é que o resultado da operação retorne um vetor com comprimento igual ao número de linhas da base (ou com comprimento 1 para distribuir um mesmo valor em todas as linhas). Você também pode criar/modificar quantas colunas quiser dentro de um mesmo mutate.

```
imdb %>%
  mutate(lucro = receita - orcamento, pais = "Estados Unidos") %>%
  select(titulo, lucro, pais)
```

```
## # A tibble: 11,340 x 3
##    titulo                              lucro pais
##    <chr>                               <dbl> <chr>
##  1 Broadcast News                   47331309 Estados Unidos
##  2 Murder, He Says                        NA Estados Unidos
##  3 Me, Myself & Irene               98270999 Estados Unidos
##  4 Never Give a Sucker an Even Break      NA Estados Unidos
##  5 Adam & Steve                           NA Estados Unidos
##  6 Henry Gamble's Birthday Party          NA Estados Unidos
##  7 No Way Out                       20509515 Estados Unidos
##  8 Welcome Home, Roscoe Jenkins      8655418 Estados Unidos
##  9 Some Kind of Wonderful                 NA Estados Unidos
## 10 The Family That Preys                  NA Estados Unidos
## # i 11,330 more rows
```

Sumarizando a base

Sumarização é a técnica de se resumir um conjunto de dados utilizando alguma métrica de interesse. A média, a mediana, a variância, a frequência, a proporção, por exemplo, são tipos de sumarização que trazem diferentes informações sobre uma variável.

```
imdb %>%
  summarize(media_orcamento = mean(orcamento, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   media_orcamento
##             <dbl>
## 1       19030515.
```

Podemos calcular diversas sumarizações diferentes em um mesmo summarize. Cada sumarização será uma coluna da nova base.

```
imdb %>% summarise(
  media_orcamento = mean(orcamento, na.rm = TRUE),
  mediana_orcamento = median(orcamento, na.rm = TRUE),
  variancia_orcamento = var(orcamento, na.rm = TRUE)
)
```

```
## # A tibble: 1 x 3
##   media_orcamento mediana_orcamento variancia_orcamento
##             <dbl>             <dbl>               <dbl>
## 1       19030515.           6500000             1.05e15
```

E também sumarizar diversas colunas.

```
imdb %>% summarize(
  media_orcamento = mean(orcamento, na.rm = TRUE),
  media_receita = mean(receita, na.rm = TRUE),
  media_lucro = mean(receita - orcamento, na.rm = TRUE)
)
```

```
## # A tibble: 1 x 3
##   media_orcamento media_receita media_lucro
```

```
##             <dbl>         <dbl>       <dbl>
## 1       19030515.     54682645.   50182761.
```

Muitas vezes queremos sumarizar uma coluna agrupada pelas categorias de uma segunda coluna. Para isso, além do summarize, utilizamos também a função group_by().

O código a seguir calcula a receita média dos filmes para cada categoria da coluna "cor".

```
imdb %>%
  filter(!is.na(producao), !is.na(receita))  %>%
  group_by(producao) %>%
  summarise(receita_media = mean(receita, na.rm = TRUE))
```

```
## # A tibble: 2,299 x 2
##    producao                          receita_media
##    <chr>                                     <dbl>
##  1 .406 Production                           10580
##  2 10 West Studios                          814906
##  3 101st Street Films                       181233
##  4 10th Hole Productions                    191019
##  5 120dB Films                              557263.
##  6 1492 Pictures                          68581364.
##  7 1818 Productions                       12232628
##  8 1821 Pictures                           1537640.
##  9 19 Entertainment                        4928883
## 10 1984 Private Defense Contractors       29430198.
## # i 2,289 more rows
```