Primeiros passos no R

ensinaR

Programação

- R como calculadora
- Atribuições
- Classes
- Operadores Relacionais
- Estruturas de dados
- Importação de dados
- Pacotes
- Gráficos
- Ajuda do R

R

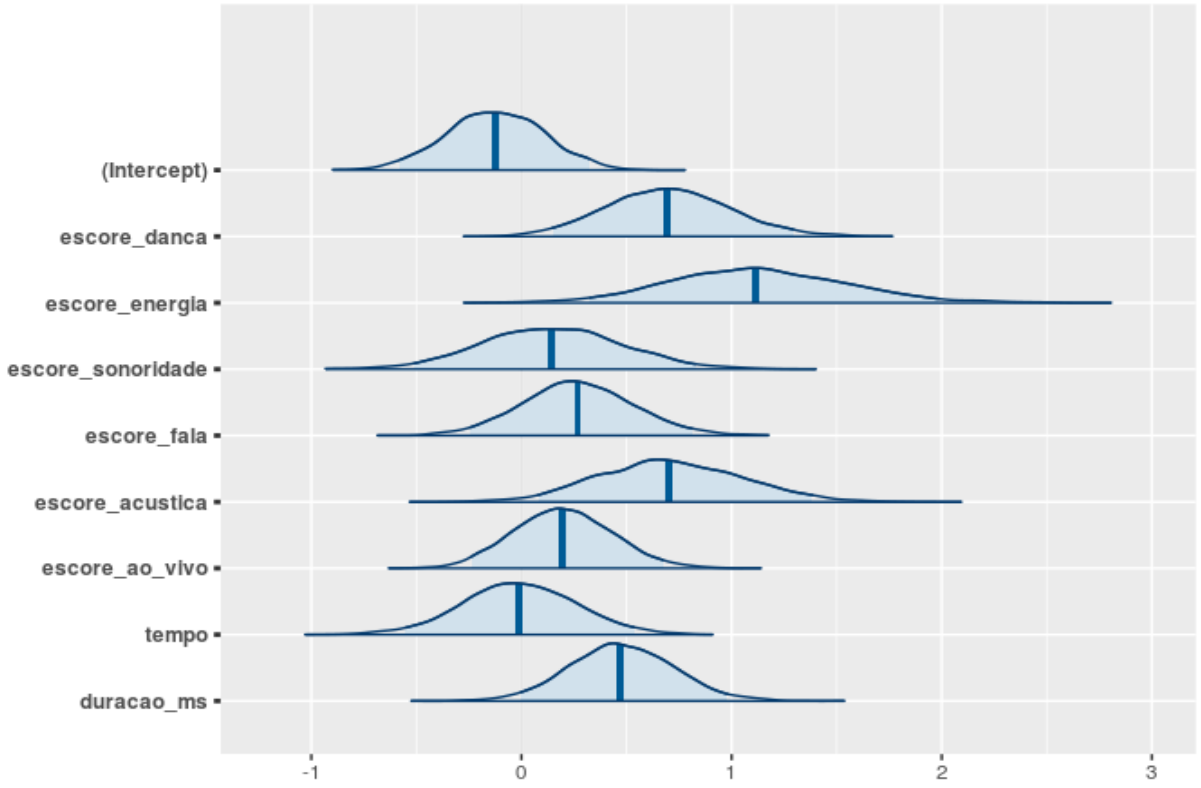
- O R é uma linguagem de programação, além de um ambiente de software gratuito.
- Oferece muitas funcionalidades acessíveis via instalações de bibliotecas.
- O R possui uma comunidade ativa, engajada no aprimoramento da ferramenta e desenvolvimento de novas bibliotecas.

RStudio

- Optar por programar em R também implica na escolha de uma IDE (Integrated Development Enviroment) que, na maioria dos casos, será o Rstudio.
- O Rstudio é um conjunto de ferramentas integradas projetadas uma (IDE - Integrated Development Enviroment) da linguagem R para editar e executar os códigos em R.

Funcionalidades do R

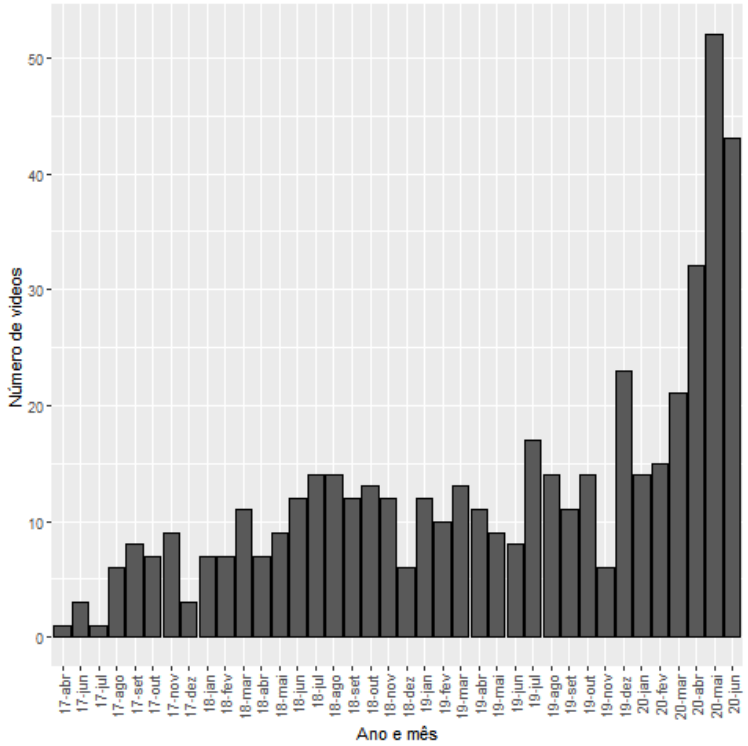
Modelagem estatística

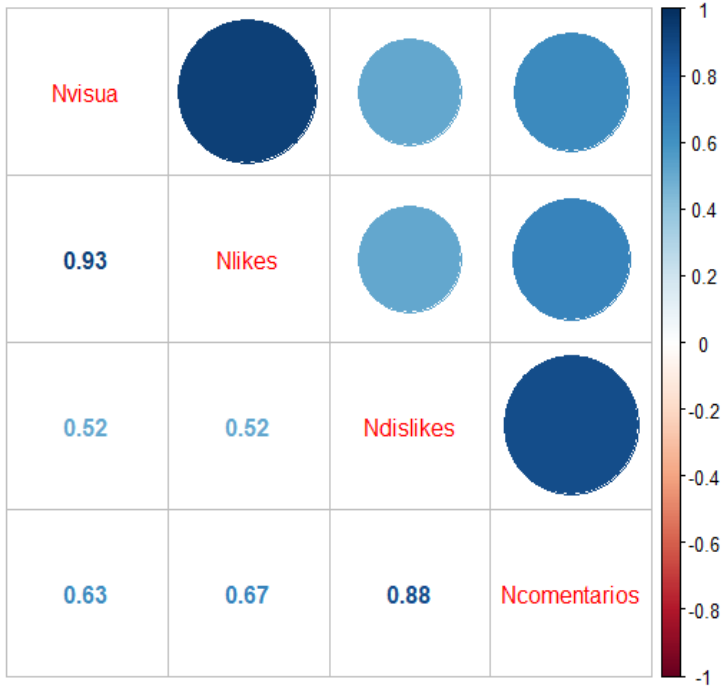


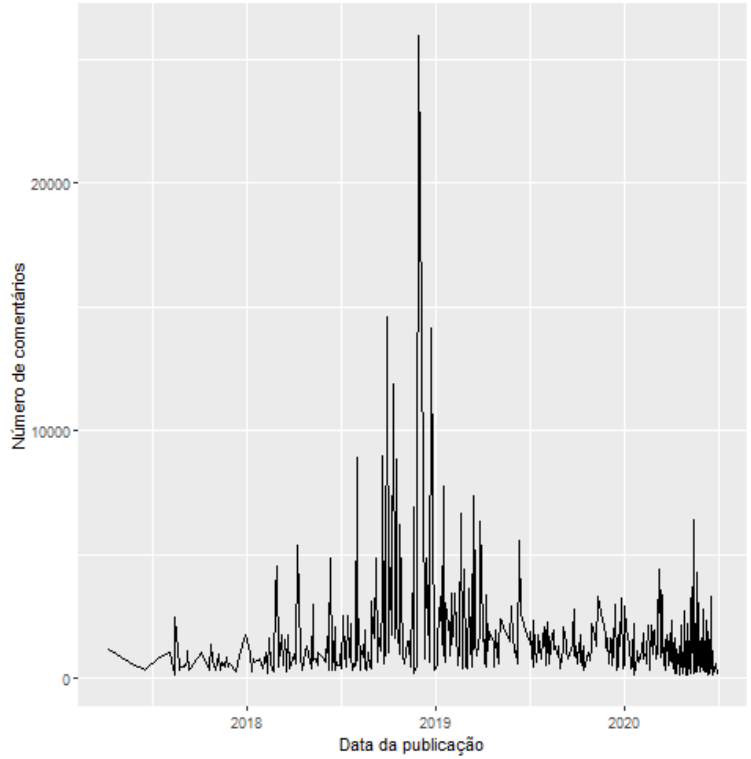
Fonte: DasLab – UFES: Construindo playlists no Spotify a partir de modelos probabilísticos.

Mineração de texto

Visualização dos dados







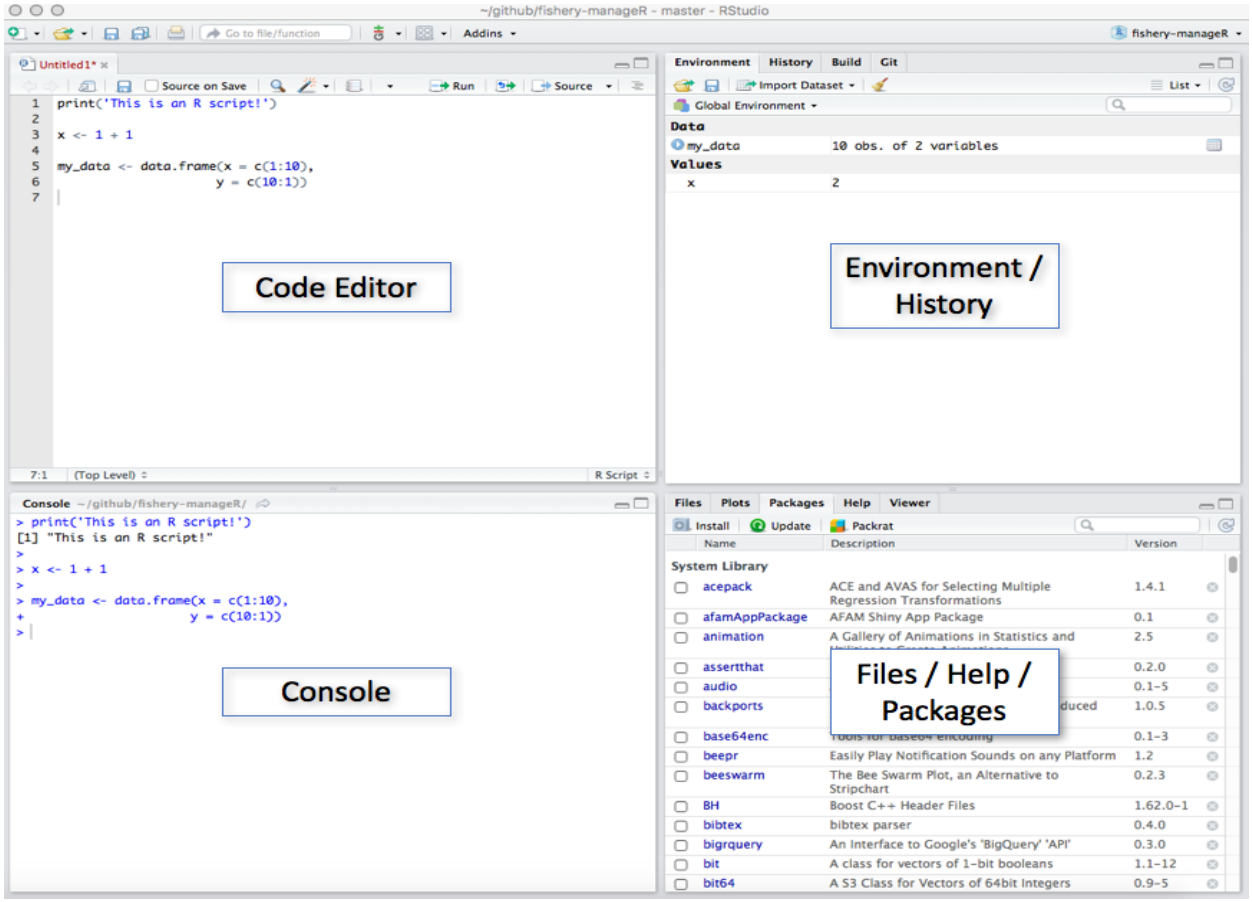
Fonte: DasLab – UFES: Analisando o melhor canal do YouTube.

Instalação do R e do RStudio

Link para material

https://daslab-ufes.github.io/materiais

RStudio


~/github/fishery-manageR - master - RStudio
Addins
fishery-manageR
Untitled1*
Source on Save
Run
Source
1 print('This is an R script!')
2
3 x <- 1 + 1
4
5 my_data <- data.frame(x = c(1:10),
6 y = c(10:1))
7
7:1
(Top Level)
R Script
Code Editor
Environment
History
Build
Git
Import Dataset
List
Global Environment
Data
my_data
10 obs. of 2 variables
Values
x
2
Environment / History
Console ~/github/fishery-manageR/
> print('This is an R script!')
[1] "This is an R script!"
>
> x <- 1 + 1
>
> my_data <- data.frame(x = c(1:10),
+ y = c(10:1))
>
Console
Files
Plots
Packages
Help
Viewer
Install
Update
Packrat
Name
Description
Version
System Library
acepack
ACE and AVAS for Selecting Multiple Regression Transformations
1.4.1
afamAppPackage
AFAM Shiny App Package
0.1
animation
A Gallery of Animations in Statistics and
2.5
assertthat
0.2.0
audio
0.1-5
backports
1.0.5
base64enc
0.1-3
beepr
Easily Play Notification Sounds on any Platform
1.2
beeswarm
The Bee Swarm Plot, an Alternative to Stripchart
0.2.3
BH
Boost C++ Header Files
1.62.0-1
bibtex
bibtex parser
0.4.0
bigrquery
An Interface to Google's 'BigQuery' 'API'
0.3.0
bit
A class for vectors of 1-bit booleans
1.1-12
bit64
A S3 Class for Vectors of 64bit Integers
0.9-5
Files / Help / Packages

RStudio

Editor/Scripts: É onde se escreve os códigos. Arquivos do tipo .R.

Console: Executa os comandos e ver os resultados.

Enviroment: Painel com todos os objetos criados.

History: Histórico dos comandos já criados.

Files: Navegar em pastas e arquivos.

Plots: Onde os gráficos serão apresentados.

Packages: Pacotes instalados.

Help: Retorna o tutorial de ajuda do comando solicitado com help ou ?comando.

Projetos e diretórios

Para criar um projeto precisamos seguir estes passos:

Clique na opção “File” do menu, e então em “New Project”.

Clique em “New Directory”.

Clique em “New Project”.

Escreva o nome do diretório(pasta) onde deseja manter seu projeto.

Clique no botão “Create Project”.

Para criar um novo script para escrever os códigos, vá em File -> New File -> R Script.

R como calculadora

#Soma

- 15 + 17

#Subtração

- 30 - 15

#Divisão

- 30/15

#Multiplicação

- 30 * 30

#Potência

- 2^2 ou 2**2

#Raiz quadrada

- sqrt(4)

Atribuição

Para atribuir a um objeto o sinal é = ou <-. Exemplos:

- x <- 9
- x

```
[1] 9
```

- y <- “Palavra”
- Y

```
[1] "Palavra"
```

O R é case sensitive, faz diferenciação entre letras maiúsculas e minúsculas.

Classes

Existem 5 classes básicas no R:

- Character: “Olá”.
- Numeric: 10.2 (números reais).
- Integer: 10 (inteiros).
- Complex: 3 + 2i (números complexos).
- Logical: 3 == 4 (True/False).

Para saber a classe de um objeto use a função class().

Operadores Relacionais

Retornam valores TRUE ou FALSE

#Maior que: >

- 3 < 4

#Menor que: <

- 3 > 4

#Igual a: ==

- 3 == 4

#Diferente de: !=

- 3 != 4

Estrutura de dados

- Vetor
- Matriz
- Data frame
- Lista
- Arrays

Vetor

- x <- c(3, 2, 1, 0, -1, -2)
- x

```
[1]  3  2  1  0 -1 -2
```

- y <- c("Azul", "Branco", "Cinza", "Amarelo", "Preto", "Vermelho")
- y

```
[1] "Azul"     "Branco"   "Cinza"    "Amarelo"  "Preto"    "Vermelho"
```

Sequências

- x <- seq(1, 100)
- x <- seq(1, 100, by = 2)

Operações Vetoriais

- x <- c(3, 2, 1, 0, -1, -2)
- w <- c(1, 2, 3, 4, 5, 6)
- x – 1

```
[1]  2  1  0 -1 -2 -3
```

- x * 2

```
[1]  6  4  2  0 -2 -4
```

- x * w

```
[1]   3   4   3   0  -5 -12
```

Matriz

- x <- matrix(seq(1, 20), nrow = 5, ncol = 4)
- x

```
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
```

#Como acessar os elementos da matriz?

- x[2, 3] #Retorna o elemento na 2º linha e terceira coluna da matriz
- x[2,] #Seleciona a 2º linha
- x[, 2] #Seleciona a 2º coluna
- x[1,] <- c(13, 15, 19, 30) #substitui a primeira linha por (13, 15, 19, 30)

Concatenar linhas em uma matriz

- vet <- c(10, 20, 30, 40)
- x2 <- rbind(x, vet)

```
    [,1] [,2] [,3] [,4]
       1    6   11   16
       2    7   12   17
       3    8   13   18
       4    9   14   19
       5   10   15   20
vet   10   20   30   40
```

Concatenar colunas em uma matriz

- vet2 <- c(60, 50, 70, 80, 90)
- x3 <- cbind(x, vet2)

```
                vet2
[1,] 1  6 11 16   60
[2,] 2  7 12 17   50
[3,] 3  8 13 18   70
[4,] 4  9 14 19   80
[5,] 5 10 15 20   90
```

Data frame

Trata – se de uma “tabela de dados” onde as colunas são as variáveis e as linhas são os registros.

Posso criar um data frame no R com os vetores:

- Nome <- c("Ana", "Bianca", "Carlos")
- Idade <- c(22, 30, 43)
- Sexo <- c("F", "F", "M")
- Peso <- c(62, 70, 52)
- Altura <- c(1.70, 1.82, 1.75)
- Pessoas <- data.frame(Nome, Idade, Sexo, Peso, Altura)
- Pessoas

```
    Nome Idade Sexo Peso Altura
1    Ana    22    F   62   1.70
2 Bianca    30    F   70   1.82
3 Carlos    43    M   52   1.75
```

Algumas funções úteis:

- str() – Estrutura do data frame. Mostra, entre outras coisas, as classes de cada coluna.
- mean(Pessoas$Altura)
- sd(Pessoas$Altura)
- summary(Pessoas$Altura)

Fator

- Sexo <- c(“M”, “H”, “M”, “H”)
- Sex <- as.factor(Sexo)
- Sex

```
[1] M H M H
Levels: H M
```

- levels(Sex)

```
[1] "H" "M"
```

Arrays

- dim3 <- array(rnorm(18), dim = c(3, 3, 2))
- dim3

```
, , 1

           [,1]        [,2]       [,3]
[1,] 1.17393449 -0.96231606  0.6837920
[2,] 0.01039629  0.23044347  0.7699587
[3,] 0.45421694 -0.07930519 -0.3563882

, , 2

           [,1]       [,2]       [,3]
[1,] -1.2691880  0.2347704  1.1823409
[2,] -1.1764635 -0.7478970 -0.4754011
[3,] -0.8333145  0.1021144  0.6653134
```

List

- ls <- list(ls1 = “a”, ls2 = c(1, 2, 3))
- ls

```
$ls1
[1] "a"

$ls2
[1] 1 2 3
```

Funções

- imc <- function(Peso, Altura){

  Peso/(Altura^2)}

- imc(Peso, Altura)

- O nome: imc
- Os argumentos: Peso e Altura
- O corpo: Peso/(Altura^2)
- O que retorna: imc(Peso, Altura)

If e else

```
a <- 224

a <- 225

If(a == b) {
    print(“igual”)
 } else {
   print(“diferente”)
}
a
```

For

- y <- seq(10, 100, 10)
- y

Quero saber a metade de cada um dos valores em y:

- for(i in y[1:10]) {

```
        print((i)/2)
}
```

Importar dados

Dados em formato csv:

dados <- read.csv(file = “local-do-arquivo", sep = ";").

Arquivos de outros softwares

Library(haven)

Dados_stata <- read.stata(“dados.dta”)

Dados_spss <- read.spss(“dados.sav”)

Dados_sas <- read.sas(“dados.sas7bdat”)

Pacotes

Instalação

Via CRAN: install.packages(“nome-do-pacote”)

Carregar pacotes

library(nome-do-pacote)

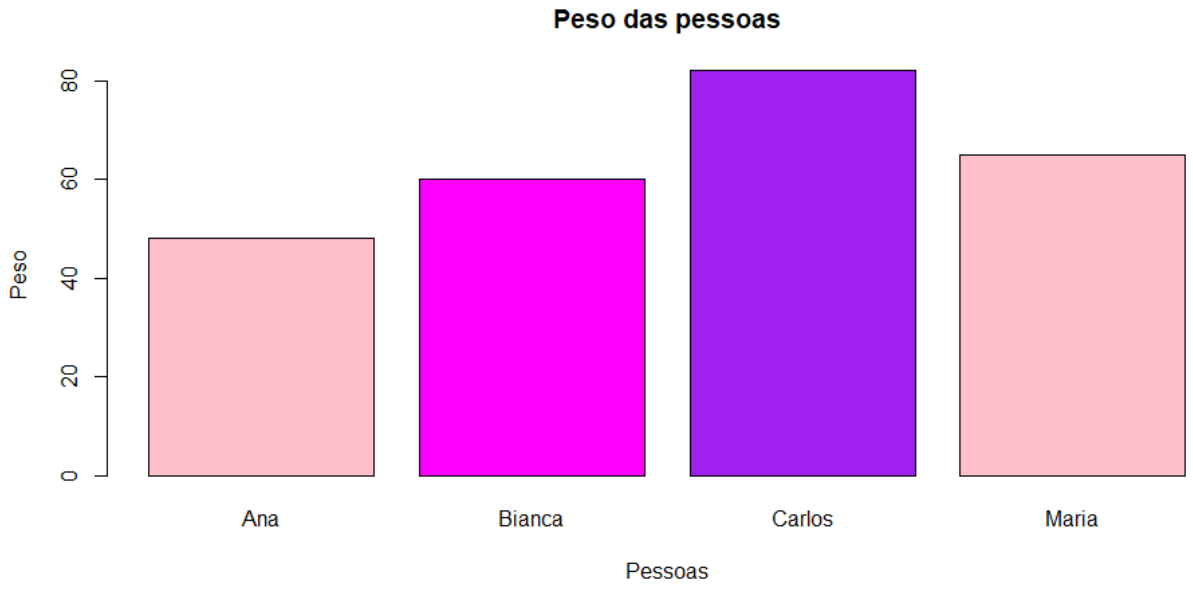
Funções úteis

- View(dados)
- names(dados)
- dim(dados)
- head(dados)
- filter(dados, Sexo == “F”)
- dados$Sexo <- factor(dados$Sexo, levels = c("F","M","I"), labels = c("Feminino", "Masculino", "Ignorados"))
- select(dados)

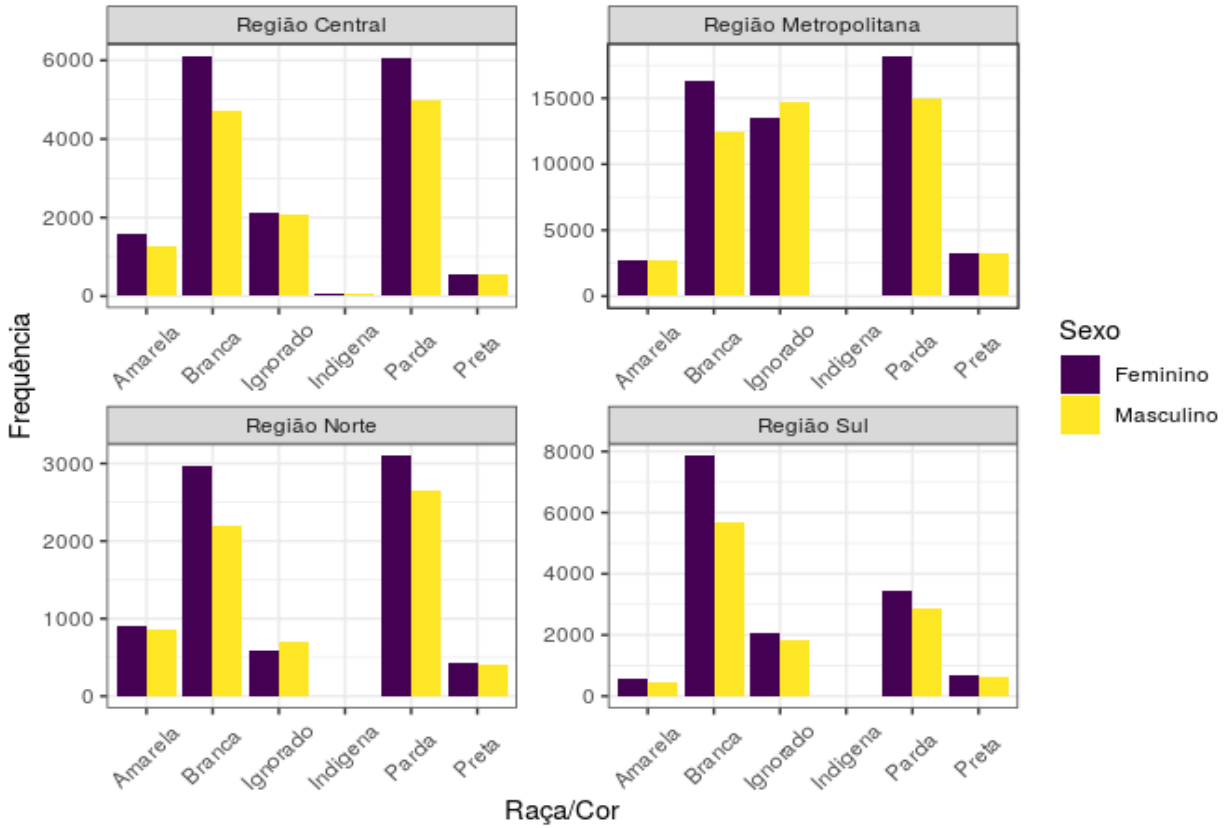
Gráficos

barplot(peso, names.arg = nome, xlab = "Pessoas", ylab = "Peso", main = "Peso das pessoas",

col = c("pink", "magenta", "purple"))



Ggplot



Fonte: DasLab – UFES: Como usar o R para analisar dados da Covid-19 no Espírito Santo.

Ajuda no R

- Help(mean)
- ?mean

Comunidade:

- Stack Overflow.