

Treinamento em Programação no Ambiente R

GENt

May 16, 2019

Dia 1

Este relatório foi feito utilizando R Markdown. Ele pode ser exportado em `.html` ou `.pdf`, basta alterar o `output`: no cabeçalho entre `html_document` e `pdf_document`.

Primeiro, um “oi” para o mundo.

```
cat("Hello world!")
```

```
## Hello world!
```

A função `cat` significa “concatenar” e ela vai imprimir o que eu escrevi no console.

Estabelecendo diretório de trabalho

É a pasta no meu computador que o R “conversa”, ou seja, que vai buscar os arquivos de entrada e solta os arquivos de saída. É uma boa prática salvar os scripts, os dados, os gráficos (tudo referente à análise) num mesmo diretório.

```
# Depende do seu computador  
# setwd("~/Documents/CursoR")
```

```
getwd() # Se eu não souber onde estou
```

```
## [1] "/home/cristiane/github/GENt-esalq.github.io/cursor"
```

Operações básicas

O R é uma grande calculadora.

```
1+1.3 #Decimal definido com "."
```

```
## [1] 2.3
```

```
2*3
```

```
## [1] 6
```

```
2^3
```

```
## [1] 8
```

```
4/2
```

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

```
sqrt(4) #raiz quadrada
```

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

```
log(100, base = 10) #logarítimo na base 10
```

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

```
log(100) #logarítimo com base neperiana
```

```
## [1] 4.60517
```

```
# Resolvendo problema
```

```
((13+2+1.5)/3) + log(96, base = 4)
```

```
## [1] 8.792481
```

Lembrando que o que vem antes do parênteses é uma função, e, sendo uma função, existe um manual para ela dentro do R, acesse com:

```
# Pedindo ajuda sobre função do R  
?log
```

Operação com vetores

```
# Diferentes formas de criar um vetor  
c(1,3,2,5,2)
```

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

```
1:10
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
seq(from=0, to=100, by=5)
```

```
## [1] 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80
```

```
## [18] 85 90 95 100
```

```
# ou
```

```
seq(0,100,5) # Se você já souber a ordem dos argumentos da função
```

```
## [1] 0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80
```

```
## [18] 85 90 95 100
```

```
seq(from=4, to=30, by=3)
```

```
## [1] 4 7 10 13 16 19 22 25 28
```

```
rep(3:5, 2)
```

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

```
# Operações
```

```
c(1,4,3,2)*2 # Multiplica todos os elementos por 2
```

```
## [1] 2 8 6 4
```

```
c(4,2,1,5)+c(5,2,6,1) # Soma 4+5, 2+2, 1+6 e assim por diante
```

```
## [1] 9 4 7 6
```

```
c(4,2,1,5)*c(5,2,6,1) # Multiplica 4*5, 2*2, 1*6 e assim por diante
```

```
## [1] 20 4 6 5
```

Criando objetos

```
x = c(30.1,30.4,40,30.2,30.6,40.1)
# ou
x <- c(30.1,30.4,40,30.2,30.6,40.1)

y = c(0.26,0.3,0.36,0.24,0.27,0.35)
```

Operações com os objetos

```
x*2
## [1] 60.2 60.8 80.0 60.4 61.2 80.2
x + y
## [1] 30.36 30.70 40.36 30.44 30.87 40.45
x*y
## [1] 7.826 9.120 14.400 7.248 8.262 14.035
z <- (x+y)/2
z
## [1] 15.180 15.350 20.180 15.220 15.435 20.225
# Aplicando algumas funções
sum(z) # soma dos valores de z
## [1] 101.59
mean(z) # média
## [1] 16.93167
var(z) # variância
## [1] 6.427507
```

Obtendo valores internos dos objetos por indexação

```
z[3] # elemento na terceira posição do vetor
## [1] 20.18
z[2:4]
## [1] 15.35 20.18 15.22
```

Para saber algumas características do objeto

```
str(z)
## num [1:6] 15.2 15.3 20.2 15.2 15.4 ...
```

Vetor de caracteres

```
clone <- c("GRA02", "UR001", "UR003", "GRA02", "GRA01", "UR001")
```

Vetor de fatores (ou variáveis categóricas)

```
clone_fator <- as.factor(clone)
str(clone_fator)
```

```
## Factor w/ 4 levels "GRA01","GRA02",...: 2 3 4 2 1 3
```

```
levels(clone_fator)
```

```
## [1] "GRA01" "GRA02" "UR001" "UR003"
```

```
length(clone_fator)
```

```
## [1] 6
```

Vetor lógico

```
logico <- x > 40
```

```
logico # Os elementos são maiores que 40?
```

```
## [1] FALSE FALSE FALSE FALSE FALSE TRUE
```

```
# Indica a posição dos TRUE
```

```
which(logico) # Obtendo as posições dos elementos TRUE
```

```
## [1] 6
```

```
x[which(logico)] # Obtendo os números maiores que 40 do vetor x pela posição
```

```
## [1] 40.1
```

Para ficar esperto/a

```
(a <- 1:10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
b <- seq(from = 0.1, to = 1, 0.1)
```

```
(b <- b*10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
a==b # Existe um problema computacional de armazenamento
```

```
## [1] TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
```

```
a==round(b) # Evitar que isso aconteça arredondando o resultado
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
?round      # Fiquei com dúvida nessa função

errado <- c(TRUE, "vish", 1) # Não podemos misturar classes num mesmo vetor
errado

## [1] "TRUE" "vish" "1"
```

Dia 2 (manhã)

Matrizes

```
X <- matrix(1:12, nrow = 6, ncol = 2)
X
```

```
##      [,1] [,2]
## [1,]    1    7
## [2,]    2    8
## [3,]    3    9
## [4,]    4   10
## [5,]    5   11
## [6,]    6   12
```

```
W <- matrix(c(x,y), nrow = 6, ncol =2)
W
```

```
##      [,1] [,2]
## [1,] 30.1 0.26
## [2,] 30.4 0.30
## [3,] 40.0 0.36
## [4,] 30.2 0.24
## [5,] 30.6 0.27
## [6,] 40.1 0.35
```

```
X*2
```

```
##      [,1] [,2]
## [1,]    2   14
## [2,]    4   16
## [3,]    6   18
## [4,]    8   20
## [5,]   10   22
## [6,]   12   24
```

```
X*X
```

```
##      [,1] [,2]
## [1,]    1   49
## [2,]    4   64
## [3,]    9   81
## [4,]   16  100
## [5,]   25  121
## [6,]   36  144
```

```
X%*%t(X)      # Multiplicação matricial
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
```

```
## [1,] 50 58 66 74 82 90
## [2,] 58 68 78 88 98 108
## [3,] 66 78 90 102 114 126
## [4,] 74 88 102 116 130 144
## [5,] 82 98 114 130 146 162
## [6,] 90 108 126 144 162 180
```

```
W[4,2] # Número posicionado na linha 4 e coluna 2
```

```
## [1] 0.24
```

```
colnames(W) <- c("altura", "diametro")
rownames(W) <- clone
W
```

```
##      altura diametro
## GRA02  30.1      0.26
## UR001  30.4      0.30
## UR003  40.0      0.36
## GRA02  30.2      0.24
## GRA01  30.6      0.27
## UR001  40.1      0.35
```

Data.frames

Vou escrever isso aqui só para mostrar que podemos criar itens no markdown

- Olha esse
- E esse
- Mais um
- Só para ter certeza

```
campo1 <- data.frame("clone" = clone,      # Antes do sinal de "="
                    "altura" = x,        # estabelecemos os nomes
                    "diametro" = y,     # das colunas
                    "idade" = rep(3:5, 2),
                    "corte" = logico)
campo1
```

```
##   clone altura diametro idade corte
## 1 GRA02  30.1      0.26     3 FALSE
## 2 UR001  30.4      0.30     4 FALSE
## 3 UR003  40.0      0.36     5 FALSE
## 4 GRA02  30.2      0.24     3 FALSE
## 5 GRA01  30.6      0.27     4 FALSE
## 6 UR001  40.1      0.35     5  TRUE
```

```
# Acessando a coluna de idades
campo1$idade
```

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

```
# ou
campo1[,4]
```

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

```

# Especificando linha e coluna
campo1[1,2]

## [1] 30.1
# Diâmetro do UR003
campo1[3,3]

## [1] 0.36
# Volume
volume <- 3.14*((campo1$diametro/2)^2)*campo1$altura
volume

## [1] 1.597287 2.147760 4.069440 1.365523 1.751131 3.856116
# Adicionando volume ao data.frame campo1
campo1 <- cbind(campo1, volume)
str(campo1)

## 'data.frame': 6 obs. of 6 variables:
## $ clone : Factor w/ 4 levels "GRA01","GRA02",...: 2 3 4 2 1 3
## $ altura : num 30.1 30.4 40 30.2 30.6 40.1
## $ diametro: num 0.26 0.3 0.36 0.24 0.27 0.35
## $ idade : int 3 4 5 3 4 5
## $ corte : logi FALSE FALSE FALSE FALSE FALSE TRUE
## $ volume : num 1.6 2.15 4.07 1.37 1.75 ...

```

Listas

```

minha_lista <- list(campo1 = campo1, media_alt = tapply(campo1$altura, campo1$idade, mean), matrix_ex =
str(minha_lista)

## List of 3
## $ campo1 : 'data.frame': 6 obs. of 6 variables:
## ..$ clone : Factor w/ 4 levels "GRA01","GRA02",...: 2 3 4 2 1 3
## ..$ altura : num [1:6] 30.1 30.4 40 30.2 30.6 40.1
## ..$ diametro: num [1:6] 0.26 0.3 0.36 0.24 0.27 0.35
## ..$ idade : int [1:6] 3 4 5 3 4 5
## ..$ corte : logi [1:6] FALSE FALSE FALSE FALSE FALSE TRUE
## ..$ volume : num [1:6] 1.6 2.15 4.07 1.37 1.75 ...
## $ media_alt: num [1:3(1d)] 30.1 30.5 40
## ..- attr(*, "dimnames")=List of 1
## .. ..$ : chr [1:3] "3" "4" "5"
## $ matrix_ex: num [1:6, 1:2] 30.1 30.4 40 30.2 30.6 40.1 0.26 0.3 0.36 0.24 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:6] "GRA02" "UR001" "UR003" "GRA02" ...
## .. ..$ : chr [1:2] "altura" "diametro"

# Acessando conteúdo das listas
minha_lista[[1]]

## clone altura diametro idade corte volume
## 1 GRA02 30.1 0.26 3 FALSE 1.597287
## 2 UR001 30.4 0.30 4 FALSE 2.147760
## 3 UR003 40.0 0.36 5 FALSE 4.069440

```

```
## 4 GRA02 30.2 0.24 3 FALSE 1.365523
## 5 GRA01 30.6 0.27 4 FALSE 1.751131
## 6 UR001 40.1 0.35 5 TRUE 3.856116
```

```
# ou
minha_lista$campo1
```

```
## clone altura diametro idade corte volume
## 1 GRA02 30.1 0.26 3 FALSE 1.597287
## 2 UR001 30.4 0.30 4 FALSE 2.147760
## 3 UR003 40.0 0.36 5 FALSE 4.069440
## 4 GRA02 30.2 0.24 3 FALSE 1.365523
## 5 GRA01 30.6 0.27 4 FALSE 1.751131
## 6 UR001 40.1 0.35 5 TRUE 3.856116
```

```
# Arrays
(meu_array <- array(1:24, dim = c(2,3,4)))
```

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]   1   3   5
## [2,]   2   4   6
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]   7   9  11
## [2,]   8  10  12
##
## , , 3
##
##      [,1] [,2] [,3]
## [1,]  13  15  17
## [2,]  14  16  18
##
## , , 4
##
##      [,1] [,2] [,3]
## [1,]  19  21  23
## [2,]  20  22  24
```

Exportando e importando dados

```
# Salvo meu objeto campo1
save(campo1, file = "campo1.RData")

# Removo o objeto
rm(campo1) # Certifique-se que salvou o objeto antes de removê-lo

# Chamo ele de novo
load("campo1.RData")

#save.image() #salva um .RData no meu diretório de trabalho
```



```
## $ Para.que.você.utiliza.ou.utilizará.o.R. : chr "programa
## $ Qual.a.sua.motivação.para.fazer.este.curso...texto.de.30.a.100.palavras. : chr "material
# também
dim(dados)

## [1] 35 12
```

Alterando nome das colunas

```
colnames(dados) <- c("Data_pesq", "Idade", "Niver", "Genero", "Cidade",
                    "Altura", "Peso", "Area", "ConhecimentoR", "Outras_linguagens",
                    "Utilizacao", "Motivacao")
str(dados)

## 'data.frame':   35 obs. of  12 variables:
## $ Data_pesq      : chr  "5/16/2019 11:19:37" "5/16/2019 11:30:40" "5/16/2019 14:21:20" "5/17/2019
## $ Idade          : int   38 27 26 24 25 34 45 22 31 21 ...
## $ Niver          : chr   "02/01" "28/05" "21/04" "07/02" ...
## $ Genero         : chr   "Masculino" "Feminino" "Feminino" "Feminino" ...
## $ Cidade         : chr   "Piracicaba-SP" "Guaxupé - MG" "São José dos Campos" "Alta Floresta - MT"
## $ Altura         : num   1.82 1.5 1.56 1.64 1.7 1.64 1.88 1.81 1.73 1.63 ...
## $ Peso           : int   91 58 56 58 54 56 93 85 75 58 ...
## $ Area           : chr   "Biológicas" "Biológicas" "Interdisciplinar" "Interdisciplinar" ...
## $ ConhecimentoR  : int    3 2 1 2 4 2 0 2 3 1 ...
## $ Outras_linguagens: chr   "Não" "Não" "Não" "Não" ...
## $ Utilizacao     : chr   "programa livre" "Tese de Doutorado" "Análise de dados de pesquisa" "Anál.
## $ Motivacao      : chr   "material mais robusto para análise de dados" "Importância de aprendizado
```

Paradoxo do aniversário

```
table(dados$Niver)

##
##      02/01      02/09      06/04      06/12      07/02      08/04
##          1          1          1          1          1          1
##      08/06      08/10      09/04      09/05      10/10      10/11
##          1          1          1          1          1          1
##      11/04      12/73 16/10/1966      17/01      17/06      17/08
##          1          1          1          1          1          1
##      18/12      19/06      20/05      21/04      22/08      23/05
##          1          1          1          1          1          1
##      23/11/      25/08      25/09 25/09/1993 26/05/1989      27/03
##          1          1          1          1          1          1
##      28/05      30/05      30/11      31/08      31/12
##          1          1          1          1          1
```

Estruturas condicionais

```
## If e else
```

```
if(2 >3){
    print("dois é maior que três")
} else {
    print("dois não é maior que três")
}
```

```
## [1] "dois não é maior que três"
```

```
if(dados[3,9] == 0){
    print("Nunca é tarde para começar!")
} else {
    print("Já pegou o embalo, agora é só continuar!")
}
```

```
## [1] "Já pegou o embalo, agora é só continuar!"
```

```
if(dados[7,9] == 0){
    print("Nunca é tarde para começar!")
} else if (dados[3,9] > 0 && dados[3,9] < 5){
    print("Já pegou o embalo, agora é só continuar!")
} else {
    print("Nos avise se estivermos falando algo errado...hehe")
}
```

```
## [1] "Nunca é tarde para começar!"
```

```
## Switch
```

```
switch(dados[5,8],
      Exatas = print("Será que aprendeu alguma linhagem de programação na graduação?"),
      Interdisciplinar = print("Em que foi a graduação?"),
      print("Ta aqui colocando o pezinho na exatas")
)
```

```
## [1] "Ta aqui colocando o pezinho na exatas"
```

Estruturas de repetição

```
## For
```

```
for(i in 1:10){
    print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```

test <- vector()
for(i in 1:10){
  test[i] <- i+4
}
test

```

```
## [1] 5 6 7 8 9 10 11 12 13 14
```

```

for(i in 1:nrow(dados)){
  if(dados[i,9] == 0){
    print("Nunca é tarde para começar!")
  } else if (dados[i,9] > 0 && dados[i,9] < 5){
    print("Já pegou o embalo, agora é só continuar!")
  } else {
    print("Nos avise se estivermos falando algo errado...hehe")
  }
}

```

```

## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Nunca é tarde para começar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Nos avise se estivermos falando algo errado...hehe"
## [1] "Nunca é tarde para começar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Nos avise se estivermos falando algo errado...hehe"
## [1] "Nunca é tarde para começar!"
## [1] "Nos avise se estivermos falando algo errado...hehe"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Nunca é tarde para começar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Nunca é tarde para começar!"
## [1] "Nunca é tarde para começar!"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Nos avise se estivermos falando algo errado...hehe"
## [1] "Já pegou o embalo, agora é só continuar!"
## [1] "Já pegou o embalo, agora é só continuar!"

```

```
# Exemplo do uso da função grepl  
grepl("-", dados[1,5]) # A primeira linha contem o caracter "-"
```

```
## [1] TRUE  
for(i in 1:nrow(dados)){  
  if(grepl("-", dados[i,5])){  
    cat("Esse/a seguiu o exemplo direitinho. Parabéns!\n")  
  } else {  
    cat("Precisamos adicionar mais informações na linha", i, "\n")  
  }  
}
```

```
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 3  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 7  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 10  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 13  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 15  
## Precisamos adicionar mais informações na linha 16  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 20  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 24  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 27  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Precisamos adicionar mais informações na linha 30  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!  
## Esse/a seguiu o exemplo direitinho. Parabéns!
```

```
corrigir <- vector()  
for(i in 1:nrow(dados)){  
  if(grepl("-", dados[i,5])){  
    cat("Esse/a seguiu o exemplo direitinho. Parabéns!\n")  
  } else {  
    cat("Precisamos adicionar mais informações na linha", i, "\n")  
  }  
}
```

```
    corrigir <- c(corrigir, i)
  }
}
```

```
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 3
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 7
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 10
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 13
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 15
## Precisamos adicionar mais informações na linha 16
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 20
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 24
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 27
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Precisamos adicionar mais informações na linha 30
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
## Esse/a seguiu o exemplo direitinho. Parabéns!
```