

# Treinamento em Programação no Ambiente R

*GENT*

*July 24, 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/fernando/Data/Diversos/GENT/GENT-esalq.github.io/XIII_GMP"
```

### 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) #logarítmo na base 10
```

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

```
log(100) #logarítmo 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

### 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

```
# Criando objetos
```

```
Repeticao <- rep(1:4,3)
```

```
Tratamento <- c(rep("L.A",4), rep("L.B",4), rep("hibridoAB", 4) )
```

```
numero_folhas <- c(5,4,5,NA,13,9,10,11,6,9,9,9)
```

```
altura <- c(95,55,70,NA,61,77,80,97,140,155,175,170)
```

```
comprimento <- c(86,30,69,NA,80,77,88,87,95,75,95,100)
```

```
# Criando data frame
```

```
campo1 <- data.frame("Repeticao" = Repeticao,      # Antes do sinal de "="
                    "Tratamento" = Tratamento, # estabelecemos os nomes
                    "numero_folhas" = numero_folhas, # das colunas
                    "altura" = altura,
                    "comprimento" = comprimento)
```

```
campo1
```

```
##      Repeticao Tratamento numero_folhas altura comprimento
## 1           1          L.A             5     95             86
## 2           2          L.A             4     55             30
## 3           3          L.A             5     70             69
## 4           4          L.A            NA     NA             NA
```

```
## 5      1      L.B      13      61      80
## 6      2      L.B      9       77      77
## 7      3      L.B     10      80      88
## 8      4      L.B     11      97      87
## 9      1  hibridoAB     6     140     95
## 10     2  hibridoAB     9     155     75
## 11     3  hibridoAB     9     175     95
## 12     4  hibridoAB     9     170    100
```

```
#Acessar coluna
campo1$altura
```

```
## [1] 95 55 70 NA 61 77 80 97 140 155 175 170
```

```
campo1[,4]
```

```
## [1] 95 55 70 NA 61 77 80 97 140 155 175 170
```

```
#Acessar elemento
campo1[1,2]
```

```
## [1] L.A
## Levels: hibridoAB L.A L.B
```

```
campo1[3,4]
```

```
## [1] 70
```

```
#Adicionar nova coluna
altura_m <- campo1$altura/100
altura_m
```

```
## [1] 0.95 0.55 0.70 NA 0.61 0.77 0.80 0.97 1.40 1.55 1.75 1.70
```

```
campo1 <- cbind(campo1, altura_m)
str(campo1)
```

```
## 'data.frame': 12 obs. of 6 variables:
## $ Repeticao : int 1 2 3 4 1 2 3 4 1 2 ...
## $ Tratamento : Factor w/ 3 levels "hibridoAB","L.A",...: 2 2 2 2 3 3 3 3 1 1 ...
## $ numero_folhas: num 5 4 5 NA 13 9 10 11 6 9 ...
## $ altura : num 95 55 70 NA 61 77 80 97 140 155 ...
## $ comprimento : num 86 30 69 NA 80 77 88 87 95 75 ...
## $ altura_m : num 0.95 0.55 0.7 NA 0.61 0.77 0.8 0.97 1.4 1.55 ...
```

## Exportando e importando dados

```
# Exportando
write.table(campo1, file = "campo1.txt", sep = ";", dec = ".", row.names = FALSE)
write.csv(campo1, file = "campo1.csv", row.names = TRUE)
```

## Importando tabelas

```
campo1_txt <- read.table(file = "campo1.txt", sep=";", dec=".", header = TRUE)
campo1_csv <- read.csv(file = "campo1.csv")
head(campo1_txt)
```

```
##   Repeticao Tratamento numero_folhas altura comprimento altura_m
## 1      1      1      L.A           5     95           86     0.95
## 2      2      2      L.A           4     55           30     0.55
## 3      3      3      L.A           5     70           69     0.70
## 4      4      4      L.A           NA    NA           NA      NA
## 5      1      1      L.B          13     61           80     0.61
## 6      2      2      L.B           9     77           77     0.77
```

```
head(campo1_csv)
```

```
##   X Repeticao Tratamento numero_folhas altura comprimento altura_m
## 1 1      1      1      L.A           5     95           86     0.95
## 2 2      2      2      L.A           4     55           30     0.55
## 3 3      3      3      L.A           5     70           69     0.70
## 4 4      4      4      L.A           NA    NA           NA      NA
## 5 5      1      1      L.B          13     61           80     0.61
## 6 6      2      2      L.B           9     77           77     0.77
```

## Leitura dos dados

```
dados <- read.csv(file = "dados.csv", stringsAsFactors = FALSE, na.strings="-")
```

## If e else

```
if(2 > 3){ #o resultado de 2 > 3 é TRUE
  print("dois é maior que três")
} else {
  print("dois não é maior que três")
}
```

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

```
if(dados[3,5] < 1.5){
  print("Híbrido pequeno demais")
} else {
  print("Tamanho provavelmente adequado")
}
```

```
## [1] "Híbrido pequeno demais"
```

```
if(dados[8,5] < 1.5){
  print("Híbrido pequeno demais")
} else if (dados[8,5] > 1.5 && dados[8,5] < 2.0){
  print("Tamanho OK")
} else {
  print("Cresceu demais")
}
```

```
## [1] "Cresceu demais"
```

```
# Cansativo:
maior_altura = max(dados[,5])
```



```

if(dados[1,5] == maior_altura){           # Testando com a primeira linha
  print("A planta 1 é a maior planta")
} else if (dados[2,5] == maior_altura) { # Testando com a segunda linha
  print("A planta 2 é a maior planta")
} else if (dados[3,5] == maior_altura) { # Testando com a terceira linha
  print("A planta 3 é a maior planta")
} else if (dados[4,5] == maior_altura) { # Testando com a quarta linha
  print("A planta 4 é a maior planta")
} else if (dados[5,5] == maior_altura) { # Testando com a quinta linha
  print("A planta 5 é a maior planta")
} else {                                  # Cansei
  mensagem = paste("Vou ter que escrever", nrow(dados), "linhas?")
  print(mensagem)
}

```

```
## [1] "Vou ter que escrever 200 linhas?"
```

## 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 <- rep(0, times = 10)
for(i in 1:10){
  test[i] <- i*10
}
test

```

```
## [1] 10 20 30 40 50 60 70 80 90 100
```

```

for(i in 1:nrow(dados)){
  if(dados[i,5] < 1.5){
    print("Híbrido pequeno demais")
  } else if (dados[i,5] > 1.5 && dados[i,5] < 2.0){
    print("Tamanho OK")
  } else {
    print("Cresceu demais")
  }
}

```

```
## [1] "Híbrido pequeno demais"
```

```
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
```

```
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
```

```
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Cresceu demais"
## [1] "Tamanho OK"
```

```

## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Tamanho OK"
## [1] "Cresceu demais"
## [1] "Tamanho OK"
## [1] "Híbrido pequeno demais"
## [1] "Cresceu demais"
## [1] "Híbrido pequeno demais"
## [1] "Híbrido pequeno demais"

```

```
maior_altura = max(dados[,5])
```

```

# Usando laço de repetição
for(i in 1:nrow(dados)){
  if(dados[i,5] == maior_altura){
    print(paste("A planta", i, "é a maior planta"))
  }
}

```

```
## [1] "A planta 8 é a maior planta"
```

```

# Usando a função which
indice = which(dados[,5] == maior_altura)
print(paste("A planta", indice, "é a maior planta"))

```

```
## [1] "A planta 8 é a maior planta"
```

## Loop dentro de loop

```
# Criando uma matrix vazia
ex_mat <- matrix(nrow=10, ncol=10)

# cada número dentro da matrix será o produto no índice da coluna pelo índice da linha
for(i in 1:dim(ex_mat)[1]) {
  for(j in 1:dim(ex_mat)[2]) {
    ex_mat[i,j] = i*j
  }
}
```

ex\_mat

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]  1   2   3   4   5   6   7   8   9   10
## [2,]  2   4   6   8  10  12  14  16  18  20
## [3,]  3   6   9  12  15  18  21  24  27  30
## [4,]  4   8  12  16  20  24  28  32  36  40
## [5,]  5  10  15  20  25  30  35  40  45  50
## [6,]  6  12  18  24  30  36  42  48  54  60
## [7,]  7  14  21  28  35  42  49  56  63  70
## [8,]  8  16  24  32  40  48  56  64  72  80
## [9,]  9  18  27  36  45  54  63  72  81  90
## [10,] 10  20  30  40  50  60  70  80  90  100
```