Atividade-da-Aula-2

Bruna Mariz

01/04/2021

# Atividade da Aula 2

Passo a passo que me fizeram chegar até aqui:

1º Criar repositório do projeto no Github (Acessar conta no Github)

2º Criação do Projeto no RStudio (Abro o R; Sandbox - New projet; abre a acaixa de texto; escolho Version Control; Escolho Git; Preencho os capos do repositório com o link que fiz no Github; Aperto em criar)

3º Agora crio meu arquivo no RMarkdown (File; New File; Rmarkdown; Preencho o título)

4º Salvo o arquivo de trabalho no meu computador (clico no disquete e salvo na pasta que eu quero –> salvar arquivo na extensão rmt de Rmarkdown)

## Testando tutorial de introdução ao R da web

Nessa atividade escolhi um tutorial de introdução ao R encontrado no site <http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:soft:rprincip>

Escolhi esse tuturial por ter os comandos bem explicados e com exemplos nos chunks

Quero testar todos os comandos que eles fazem referência

obs. ```{r} sempre abre o chunk (primeira linha) e ’’’ fecha o chunk (última linha) –> se eu coloco algum comando junto as de abertura e fechamento do chunk, a afirmativa não é lida

As linhas de comando vão sempre ficar no chunk As respostas dos chunks vão estar no console (tela abaixo a esta) No console será repetido a operação do chunk (em azul) e a resposta ser logo abaixo em preto Sempre antes do resultado, aparece um [1]–> Isso é um contador (exemplo 1:100)

### Operações Aritméticas

O R faz operações matématicas desde o básico ao avançado. Em operações aritméticas, basta colocar o símbolo como se fosse na calculadora convencional

2+2

## [1] 4

5-1

## [1] 4

2\*3

## [1] 6

4/5

## [1] 0.8

1:100 #demonstração que [1]é um contador --> a cada linha da sequência de 1 a 100 aparece [número em que inicia a linha]

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18  
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36  
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54  
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72  
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  
## [91] 91 92 93 94 95 96 97 98 99 100

### Operações com objetos

No R é possivel criar ou fazer referência a objetos. 1º Escolho um nome 2º Atribuo ao nome valor ou um conjunto de valores ou palavras 3º Se quiser que o objeto tenha mais de um elemento escrevo –> c(“elemento 1”, “elemento 2”)

valor <- 5 #comando correto ... #comando errado se eu der espaço entre < -  
pares <- c(0,2,4,6,8)  
cores <- c("azul", "vermelho", "amarelo")  
#ou posso usar = em vez de <-  
valor = 5  
pares = c(0,2,4,6,8)  
cores = c("azul", "vermelho", "amarelo")

Obs. Quando usamos o símbolo “=” ou “<” seguido de “-” estamos criando objetos com um nome que aparece a esquerda e que contém alguns elementos (o que vem a direita do símbolo). Quando criamos um objeto, ele fica guardado na memória do R até que você feche o programa. Porém, os objetos criados ficam ocultos. Para ver a lista de arquivos ocultos basta dar o seguinte comando:

ls()[pares[-1]] #correto ... errado se eu colocar espaço entre o parentese

## [1] "pares" NA NA NA

############### não consegui obter resultado esperado neste comando --> aparece várias afirmações quando coloco valor entre parenteses e inválido quando coloco pares

### Operações de jogo

Pode-se jogar com os dados no R. Funciona como? a cada vez que você roda uma amostra o R faz um sorteio..

moeda <- c("cara","coroa")  
moeda

## [1] "cara" "coroa"

sample(moeda,1)

## [1] "cara"

sample(moeda,1)

## [1] "coroa"

sample(moeda,1)

## [1] "cara"

sample(moeda,1)

## [1] "coroa"

dado <- c(1:6)  
dado

## [1] 1 2 3 4 5 6

sample(dado,1)

## [1] 6

sample(dado,1)

## [1] 4

sample(dado,1)

## [1] 3

### Matrizes

Transforma um conjunto de dados em uma matriz. Ele dispõe todos os valores de acordo com o tanto de linhas e colunas que você pedir. Por isso é importante que a dimensão da matrix comporte de fato seus dados ou utilize um comando que dê várias matrizes com seus dados.

1:100

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18  
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36  
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54  
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72  
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90  
## [91] 91 92 93 94 95 96 97 98 99 100

matrix(1:100,nrow=20,ncol=5) #nrow define o número de linhas (20) e ncol o de colunas (5)

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 21 41 61 81  
## [2,] 2 22 42 62 82  
## [3,] 3 23 43 63 83  
## [4,] 4 24 44 64 84  
## [5,] 5 25 45 65 85  
## [6,] 6 26 46 66 86  
## [7,] 7 27 47 67 87  
## [8,] 8 28 48 68 88  
## [9,] 9 29 49 69 89  
## [10,] 10 30 50 70 90  
## [11,] 11 31 51 71 91  
## [12,] 12 32 52 72 92  
## [13,] 13 33 53 73 93  
## [14,] 14 34 54 74 94  
## [15,] 15 35 55 75 95  
## [16,] 16 36 56 76 96  
## [17,] 17 37 57 77 97  
## [18,] 18 38 58 78 98  
## [19,] 19 39 59 79 99  
## [20,] 20 40 60 80 100

array(1:100,dim=c(5,5,4)) # o argumento agora é dim... deve conter os valores de cada dimensão c('linha', 'colunas', 'matriz')--> O resultado aparece como fatias de um bolo, onde cada camada é uma matriz

## , , 1  
##   
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 6 11 16 21  
## [2,] 2 7 12 17 22  
## [3,] 3 8 13 18 23  
## [4,] 4 9 14 19 24  
## [5,] 5 10 15 20 25  
##   
## , , 2  
##   
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 26 31 36 41 46  
## [2,] 27 32 37 42 47  
## [3,] 28 33 38 43 48  
## [4,] 29 34 39 44 49  
## [5,] 30 35 40 45 50  
##   
## , , 3  
##   
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 51 56 61 66 71  
## [2,] 52 57 62 67 72  
## [3,] 53 58 63 68 73  
## [4,] 54 59 64 69 74  
## [5,] 55 60 65 70 75  
##   
## , , 4  
##   
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 76 81 86 91 96  
## [2,] 77 82 87 92 97  
## [3,] 78 83 88 93 98  
## [4,] 79 84 89 94 99  
## [5,] 80 85 90 95 100

### Objetos vázios

Nesse comando os objetos são vázios ou não uteis. Usamos NA –> não se aplica, ou seja, aquela posição não está sendo ocupada por nenhum valor

a=rep(NA,10)  
a

## [1] NA NA NA NA NA NA NA NA NA NA

b=matrix(NA,nrow=3,ncol=3)  
b

## [,1] [,2] [,3]  
## [1,] NA NA NA  
## [2,] NA NA NA  
## [3,] NA NA NA

c=array(NA,dim=c(3,3,3))  
c

## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA  
## [2,] NA NA NA  
## [3,] NA NA NA  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA  
## [2,] NA NA NA  
## [3,] NA NA NA  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA  
## [2,] NA NA NA  
## [3,] NA NA NA

Por que fazer isso? Depois preenchemos os lugares vazios com outros valores do nosso interesse

a[1]=2  
a

## [1] 2 NA NA NA NA NA NA NA NA NA

a[1]

## [1] 2

b[3,3]="oi"  
b

## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "oi"

b[2,2]="oi"  
b

## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA "oi" NA   
## [3,] NA NA "oi"

b[1,1]="oi"  
b

## [,1] [,2] [,3]  
## [1,] "oi" NA NA   
## [2,] NA "oi" NA   
## [3,] NA NA "oi"

b[3,3]

## [1] "oi"

c[3,3,1]="oi"  
c

## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "oi"  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA NA   
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA NA

c[3,3,2]="ai"  
c

## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "oi"  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "ai"  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA NA

c[3,3,3]="ui"  
c

## , , 1  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "oi"  
##   
## , , 2  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "ai"  
##   
## , , 3  
##   
## [,1] [,2] [,3]  
## [1,] NA NA NA   
## [2,] NA NA NA   
## [3,] NA NA "ui"

c[3,3,3]

## [1] "ui"

obs. Os comandos acima servem para indexar também.. ou seja, dizem a posição de um objeto. Quando coloco a em seguida a posição do valor da matriz entre colchetes (nome [posição]).

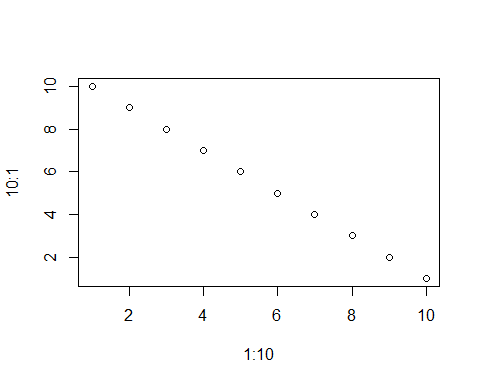
No caso da matriz b (tridimensional) coloco –> nome[número da linha, número da coluna]

Se for a matriz c (tridimensional) coloco –> nome[número da linha, número da coluna, número da fatia do bolo]

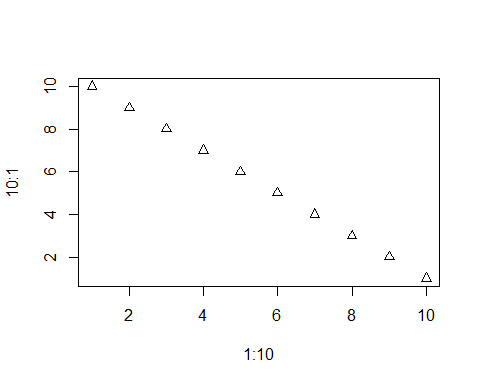
### Gráficos

Segue abaixo vários tipos de gráficos: Linear Histograma Boxplot

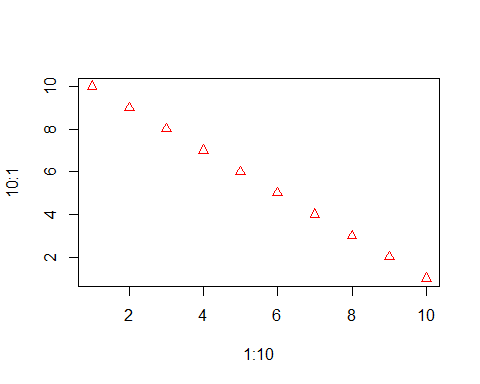
plot(1:10,10:1)



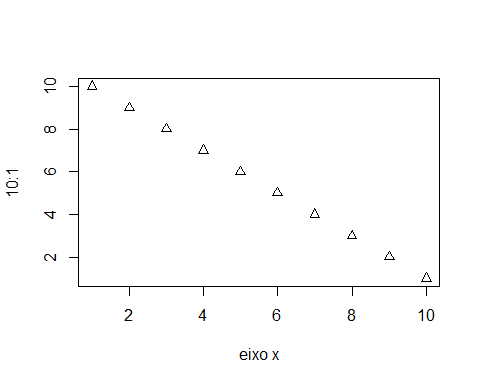
plot(1:10,10:1,pch=2)



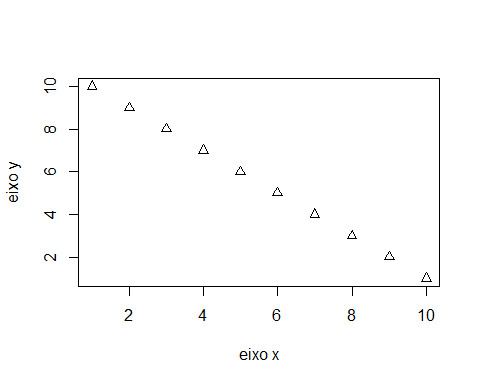
plot(1:10,10:1,pch=2,col=2)



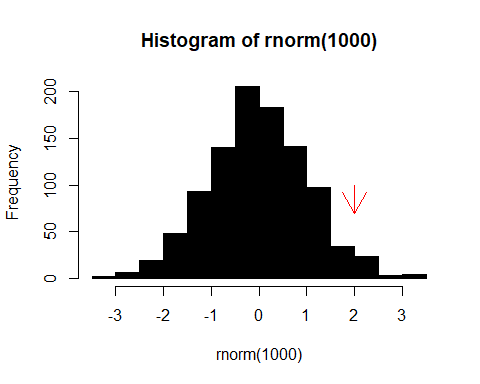
plot(1:10,10:1,pch=2,xlab="eixo x")



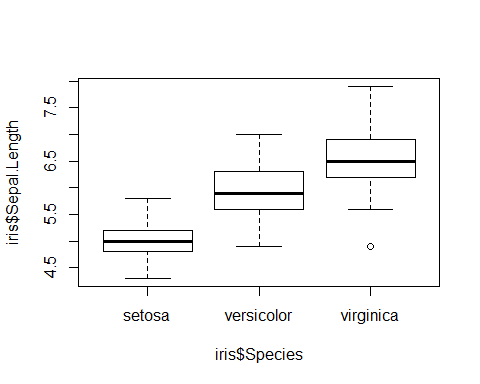
plot(1:10,10:1,pch=2,xlab="eixo x",ylab="eixo y")



hist(rnorm(1000),col=33)  
arrows(2,100,2,70,col=2)



boxplot(iris$Sepal.Length~iris$Species)



### Funções no R

Mais complicado, pois exige mais raciocinio logico na hora de montar a função que se adeque de fato ao seu objetivo.

media=function(objeto){sum(objeto)/length(objeto)} # sum --> soma e Length --> quantidade  
media(c(1,2,3))

## [1] 2

alturas=c(1.4,1.7,2.0,1.6,1.8)  
media(alturas)

## [1] 1.7

#Fazendo soma, número de elementos e média separados  
  
sum(alturas) # soma dos elementos do objeto "alturas"

## [1] 8.5

length(alturas) # número de elementos do objeto "alturas"

## [1] 5

mean(alturas) # média dos elementos do objeto "alturas"

## [1] 1.7

### Carregando Arquivos no R

São comandos que permitem a entrada de dados no R, sem precisar que estes sejam digitados todas as vezes que forem ser utilizados. Recomenda-se que as planilhas a serem importadas, estejam no formato csv.

Para definir o diretório que você está trabalhando:primeiro, verifique em que diretório o R está digitando getwd(). por padrão, no windows o diretório é a sua pasta documentos. para alterar isso, utilize o comando setwd()

getwd()

## [1] "C:/Users/Bruna/Desktop/GIT/FIT678---Atividade-da-Aula-2"

#setwd()  
  
#Exemplo  
#setwd("C:/meu\_drive/meu\_diretorio")   
# O endereço do diretório tem de estar entre aspas, e as barras (“\”) tem de ser trocadas por barras invertidas(“/”). Após usar o comando acima, confira se funcionou usando o getwd(). você também pode alterar o diretório utilizando o menu Arquivo → Mudar dir

###Formatação e leitura de arquivos

Além de salvar em CSV, deve-se separar os números decimais por ponto, uma vez que o R segue padrão americano.

Para pedir ao R que leia alguma arquivo usa-se o coamndo –> read.table() e guardamos o resultado em uma tabela. Exemplo:

#read.table()  
#tabela <- read.table(file="Dados.csv",header=T,sep=",")   
# os dados estão no arquivo “Dados.csv”, que possui uma linha de cabeçalho com o nome das minhas colunas (“header=T”), e que minhas colunas são separadas por vígulas (sep=“,”).   
  
#Se o arquivo não tiver cabeçalho, use header=F, e se suas colunas estiverem separador por outro símbolo, por exemplo ponto-e-vírgula, coloque o símbolo dentro das aspas do parâmetro sep (sep=“;”).