



Semana temática da Biologia  
IB - USP

# Comandos básicos do R (Introdução à Bioestatística)

Bárbara Maria de Andrade Costa

Murillo Fernandes

São Paulo

4 a 6 de outubro, 2016

# O que é R?

---

R é uma linguagem de programação e um ambiente de desenvolvimento integrado, para cálculos estatísticos e gráficos.

# Interfaces para o R

---

Para trabalhar mais eficientemente com o R você vai precisar de uma interface, ou de um ambiente de desenvolvimento (IDE - Integrated Development Environment).

São ambientes de programação que combinam ferramentas para escrever e executar seu código de maneira mais fácil e eficiente.

# Interface sugerida: **RStudio**

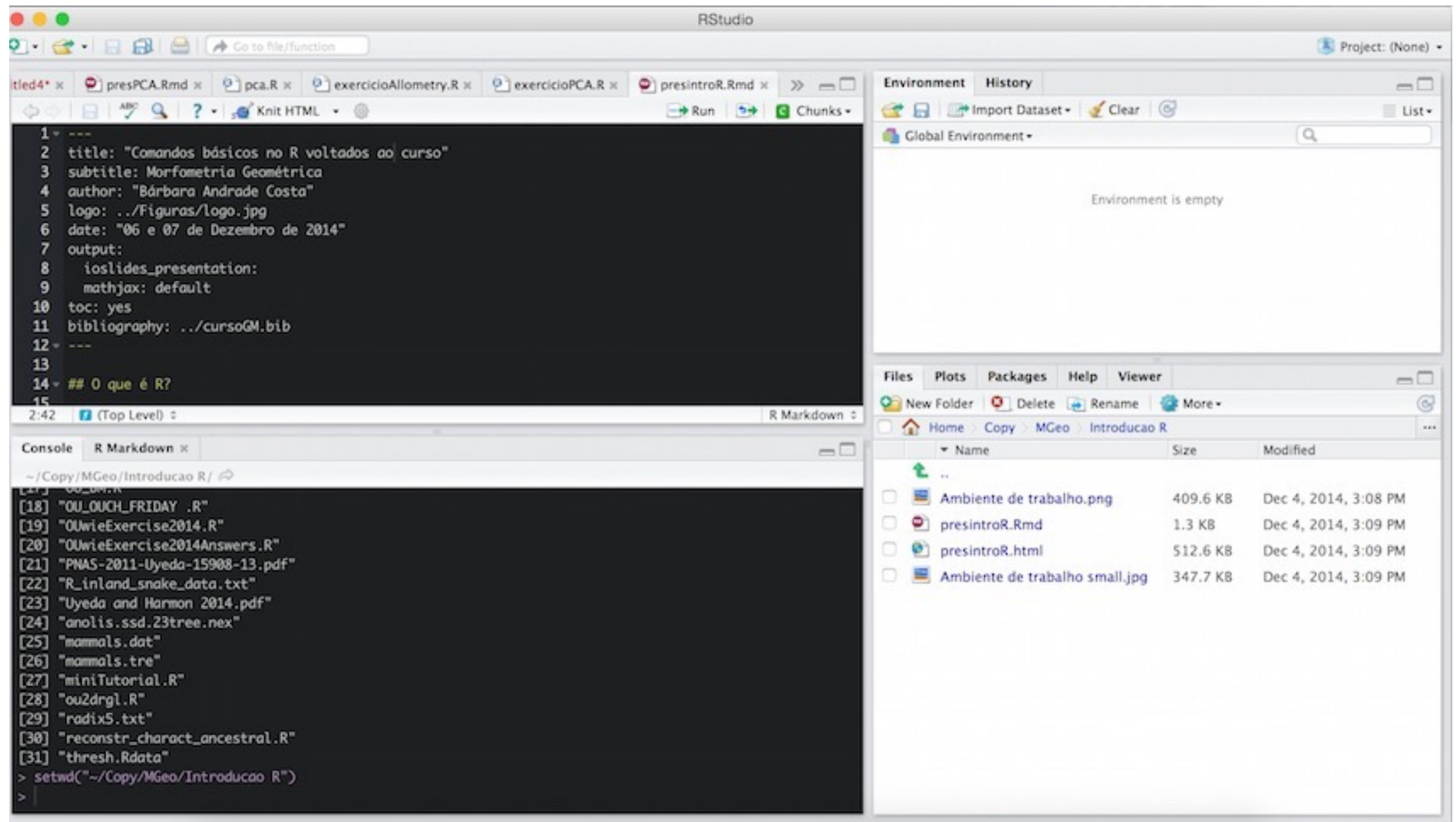
---

Primeiro IDE criado especificamente para R. Seu ponto forte é a organização das janelas sem um arranjo fixo e de fácil acesso.

Simple e intuitivo, é uma das melhores opções para começar:  
<http://www.rstudio.org>

O ambiente de trabalho

# Script, Console, Global Environment, Files/Plots/Help/Viewer



Use o # para comentar códigos no  
seu script!

# Use # e comente seus códigos no script

---

**data (plethodon)** # carrega os dados

**plethodon** # olhe o conteudo do objeto plethodon

**names (plethodon)** # olhe os nomes dentro do objeto  
plethodon

**class (plethodon)** # mostra a classe a que pertence o  
objeto plethodon



# Como trabalhar com o R no curso de Introdução à Bioestatística?

---

1. Vamos criar uma pasta para o Curso e armazená-la no diretório que você preferir (Desktop, Meus Documentos...);
2. Sempre que iniciar o R você precisa checar em qual diretório de trabalho está;
3. Para isso vamos usar: `getwd()` e `setwd()`.

# getwd() e setwd()

---

Qual o diretório de trabalho atual?

```
getwd()
```

```
[1] "/Users/tafinha/Dropbox/MGeo"
```

Vamos mudar o diretório atual de trabalho?

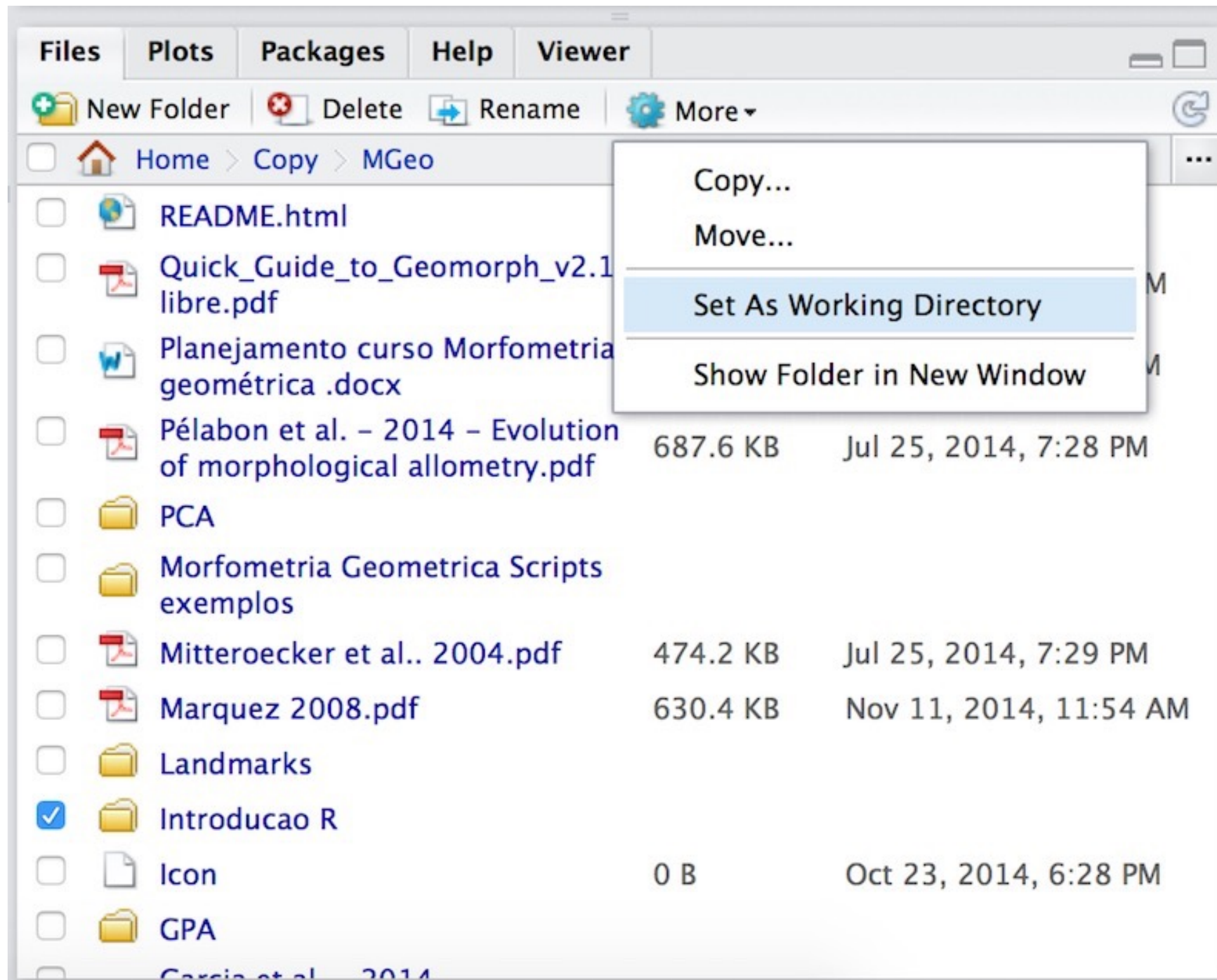
```
setwd("~/Desktop/bioestat")
```

Verificando...

```
getwd()
```

```
[1] "/Users/tafinha/Desktop/bioestat"
```

# Outra opção: vá na janela do canto inferior direito do RStudio...



# Qual os arquivos presentes no meu diretório ou pasta de trabalho?

---

`getwd()` # onde estou?

[1] "/Users/tafinha/Desktop/bioestat"

`dir()` # o que há dentro de Introducao R?

[1] "aulas" "Figuras" "scripts"

[4] "pasta"

`ls()` # listar os objetos ...

# Instalando e carregando pacotes no R

Instalar pacotes

```
install.packages()
```

# install.packages()

---

`install.packages ("car")`

tentando a URL 'https://cran.rstudio.com/bin/macosx/mavericks/contrib/  
3.3/car\_2.1-3.tgz'

Content type 'application/x-gzip' length 1448242 bytes (1.4 MB)

=====

downloaded 1.4 MB

The downloaded binary packages are in

    /var/folders/d2/g84sbdbx42sgm\_zqpx\_bhtvm0000gn/T//Rtmp97pkzS/  
downloaded\_packages

# install.packages()

---

`install.packages ("ggplot2")`

tentando a URL 'https://cran.rstudio.com/bin/macosx/mavericks/  
contrib/3.3/ggplot2\_2.1.0.tgz'

Content type 'application/x-gzip' length 2009223 bytes (1.9 MB)

=====

===

downloaded 1.9 MB

The downloaded binary packages are in

`/var/folders/d2/g84sbdbx42sgm_zqpx_bhtvm0000gn/T//`

`RtmpZSOdOR/downloaded_packages`



Carregar pacotes

**library()**

# install.packages()

---

## library (car)

Loading required package: car

Loading required package: dependencia

## library (ggplot2)

Loading required package: ggplot2

Loading required package: dependencia

Operadores <- , =

# Operadores e objetos

---

objeto <- expressão

`aves.area <- c(50, 67, 26, 22, 47, 16, 21)`

objeto = expressão

`aves.area = c(50, 67, 26, 22, 47, 16, 21)`

objeto <- expressão

`aves.riqueza <- c(10, 15, 6, 18, 5, 11, 20)`

objeto = expressão

`aves.riqueza = c(10, 15, 6, 18, 5, 11, 20)`

# Operadores e objetos

---

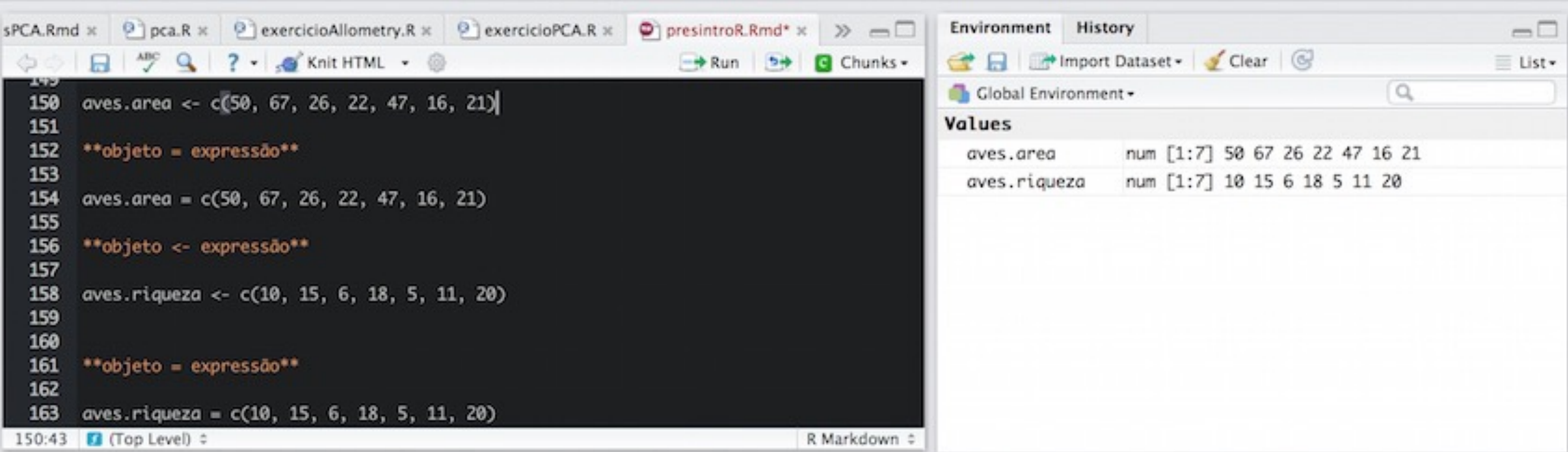
aves.area

```
[1] 50 67 26 22 47 16 21
```

aves.riqueza

```
[1] 10 15 6 18 5 11 20
```

Você acabou de criar dois objetos (veja no Global Environment)



The screenshot displays the RStudio interface. The top pane shows the source editor with R code. The bottom-right pane shows the 'Environment' tab, which lists the objects created in the Global Environment.

**Source Editor Code:**

```
149  
150 aves.area <- c(50, 67, 26, 22, 47, 16, 21)  
151  
152 **objeto = expressão**  
153  
154 aves.area = c(50, 67, 26, 22, 47, 16, 21)  
155  
156 **objeto <- expressão**  
157  
158 aves.riqueza <- c(10, 15, 6, 18, 5, 11, 20)  
159  
160  
161 **objeto = expressão**  
162  
163 aves.riqueza = c(10, 15, 6, 18, 5, 11, 20)
```

**Environment Pane:**

Values	
aves.area	num [1:7] 50 67 26 22 47 16 21
aves.riqueza	num [1:7] 10 15 6 18 5 11 20

Linguagem orientada a objetos

# Linguagem orientada a objetos

---

```
aves.area <- c(50, 67, 26, 22, 47, 16, 21)
```

```
aves.area
```

```
[1] 50 67 26 22 47 16 21
```

```
summary (aves.area) # mostra resultados resumidos
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
16.00	21.50	26.00	35.57	48.50	67.00



# ls()

Lista os objetos

## ls()

```
[1] "aves.area" "aves.riqueza" "modelo1"
```

# Sintaxe básica das funções

---

função (argumento1 = valor, argumento2=valor)

?rnorm # essa função gera uma distribuição normal aleatória

args(rnorm) # vamos olhar os argumentos da função

function (n, mean = 0, sd = 1)

USE O HELP

# Sintaxe básica das funções

---

`help(summary)`

`?summary`

Ou na janela inferior direita do RStudio digitando no help o  
que procura

# O help fornece

- O help fornece
- Descrição da função
- Como usar a função
- Os argumentos da função
- Detalhes
- Referências
- Exemplos

# Tipos de dados no R

("Atomic data types")

# Tipos de dados

---

## 1.Numeric

```
value <- 605
```

```
value
```

```
[1] 605
```

# Tipos de dados

---

## 2. Character

```
string <- "Hello World"
```

```
string
```

```
[1] "Hello World"
```



# Tipos de dados

---

## 3. Logical

$2 < 4$

[1] TRUE

# Tipos de dados

---

## 4. Complex number

```
cn <- 2+3i
```

```
cn
```

```
[1] 2+3i
```

# Tipos de objetos no R

# Tipos de objetos

---

## 1. Vetor

Contém várias cópias de um mesmo tipo de objeto. Um vetor pode conter somente objetos de mesma classe.

**atenção para a função `c()` (catenate)**

```
meu.vetor <- c(10.5,11.3,12.4,5.7)
```

```
meu.vetor
```

```
[1] 10.5 11.3 12.4 5.7
```

# Tipos de objetos

---

## 2. Matriz

um array de duas dimensões com número arbitrário de linhas e colunas

```
minha.matriz <- matrix(data=1:12,nrow=3,ncol=4)
```

```
minha.matriz
```

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

# Tipos de objetos

---

## 3. Array

como uma matriz, mas com dimensões arbitrárias (isto é, mais do que duas dimensões)

```
my.array <- array(1:24, dim=c(3,4,2))
```

```
my.array <- array(1:24, dim=c(3,4,2))
```

```
my.array
```

```
, , 1
```

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

```
, , 2
```

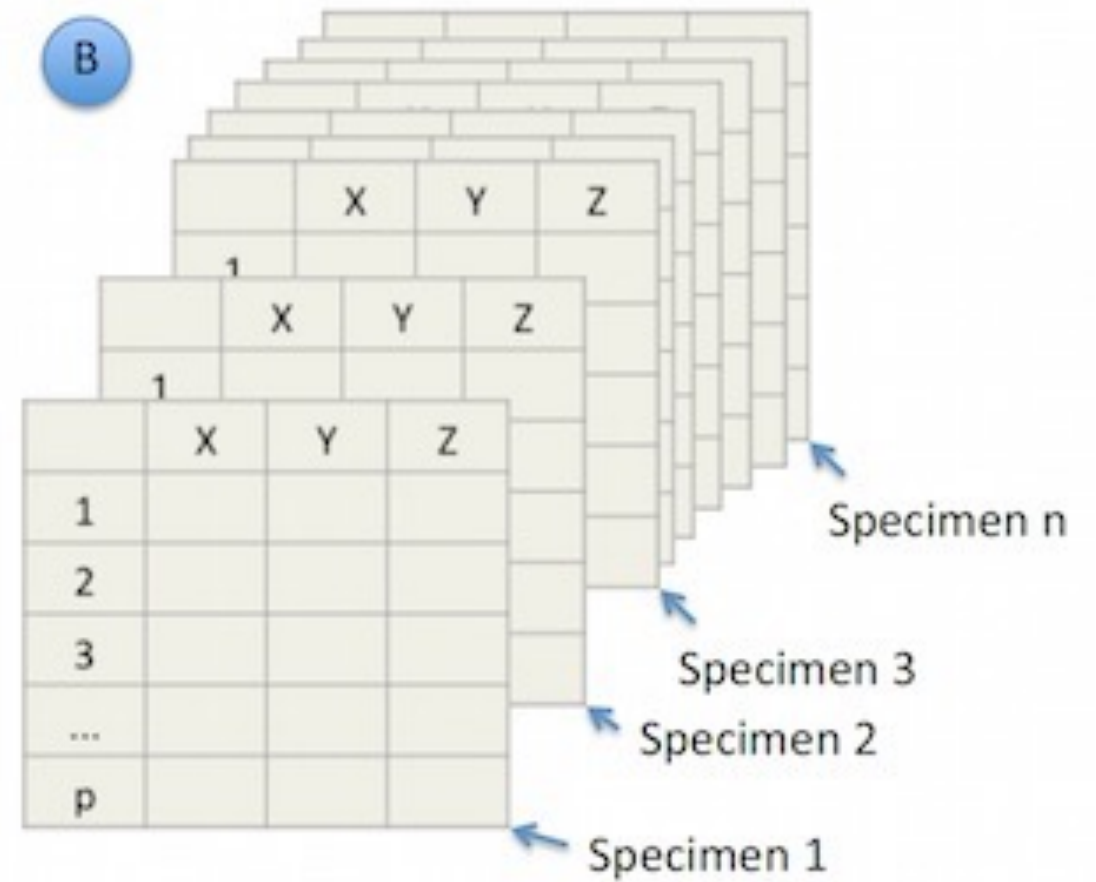
	[,1]	[,2]	[,3]	[,4]
[1,]	13	16	19	22
[2,]	14	17	20	23
[3,]	15	18	21	24

# Matriz e Array

A

	X1	Y1	Z1	X2	Y2	Z2	X3	Y3	...	Xp
Specimen 1										
Specimen 2										
Specimen 3										
...										
Specimen n										

B





# Tipos de objetos

---

## 4. Data frame

conjunto de dados organizados, similar a uma matriz. Entretanto, cada coluna do data frame deverá conter um tipo de dado.

Com a função `data.frame` reunimos vetores de mesmo comprimento em um só objeto.

# Tipos de objetos

---

## 4. Data frame

```
nome <- c("Didi","Dede","Mussum","Zacarias")
```

```
ano.nasc <- c(1936,1936,1941,1934)
```

```
vive <- c("V","V","F","F")
```

```
trapalhoes <- data.frame(nome,ano.nasc,vive)
```

```
trapalhoes
```

	nome	ano.nasc	vive
1	Didi	1936	V
2	Dede	1936	V
3	Mussum	1941	F
4	Zacarias	1934	F

O mesmo, em um só comando:

```
trapalhoes <- data.frame (nomes=c("Didi","Dedé","Mussum","Zacarias"),  
ano.nasc=c(1936,1936,1941,1934), vive=c("V","V","F","F"))
```

	nome	ano.nasc	vive
1	Didi	1936	V
2	Dede	1936	V
3	Mussum	1941	F
4	Zacarias	1934	F

# Tipos de objetos

---

## 5. Function

conjunto de comandos com entrada (input) e saída (output) definidos.

# Tipos de objetos

---

## 6. Listas

coleção arbitrária de outros objetos de R (os quais podem incluir outras listas). Pode conter objetos de diferentes classes.

`data(plethodon)` # carrega os dados

`plethodon`

São dados do pacote geomorph referentes à salamandra do gênero Plethodon.

`data(iris)`

São dados do pacote datasets referentes à medidas de 50 flores de 3 espécies de íris.

\$l and

, , 1

[ , 1] [ , 2]

[1, ] 8.89372 53.77644

[2, ] 9.26840 52.77072

[3, ] 5.56104 54.21028

[4, ] 1.87340 52.75100

...

\$links

[ , 1] [ , 2]

[1, ] 4 5

[2, ] 3 5

[3, ] 2 4

[4, ] 1 2

[5, ] 1 3

[6, ] 6 7

...

\$species

[1] Jord Jord Jord Jord Jord Jord Jord Jord Jord

Jord Teyah Teyah Teyah Teyah Teyah Teyah Teyah

[18] Teyah Teyah Teyah Jord Jord Jord Jord Jord Jord

Jord Jord Jord Jord Teyah Teyah Teyah Teyah

[35] Teyah Teyah Teyah Teyah Teyah Teyah

Levels: Jord Teyah

\$site

[1] Symp Symp Symp Symp Symp Symp Symp Symp Symp

Symp Symp Symp Symp Symp Symp Symp Symp Symp

Symp

Allo Allo Allo Allo Allo Allo Allo Allo Allo

Allo Allo Allo Allo Allo Allo Allo Allo Allo

Levels: Allo Symp

A base de dados plethodon é uma lista que contém quatro componentes:

- as coordenadas dos dados - landmarks (plethodon\$land);
- the wirelink addresses for plotting (plethodon\$links);
- dois conjuntos de variáveis como fatores species e sites



# Classes que pertencem os componentes da lista

---

```
class(plethodon$site)
```

```
[1] "factor"
```

```
class(plethodon$land)
```

```
[1] "array"
```

```
class(plethodon$links)
```

```
[1] "matrix"
```

Indexação

# Indexação - vetores

---

```
a = c(1,2,3,4,5,6)
```

```
a
```

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

```
a[1]
```

```
[1] 1
```

```
a[9]
```

```
[1] NA
```

```
a[a<5]
```

```
[1] 1 2 3 4
```

```
a[ -(2:4) ]
```

```
[1] 1 5 6 9
```

```
a[ -c(1,5,9) ]
```

```
[1] 2 3 4 6 9
```

# Indexação - vetores

---

```
biomas = c (4, 6, 2, 5)
```

```
biomas
```

```
[1] 4 6 2 5
```

```
names(biomas) = c ("Amazonia", "Cerrado", "Pantanal", "MA")
```

```
biomas
```

Amazonia	Cerrado	Pantanal	MA
4	6	2	5

```
biomas[ c("Cerrado","MA") ]
```

Cerrado	MA
6	5

# Indexação - matrizes e data frases

---

## Indexação [linhas,colunas]:

```
matrix.a = matrix (data=1:12,nrow=3,ncol=4)
```

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

```
matriz[1,1]
```

```
[1] 1
```

```
matriz[1:2,1]
```

```
[1] 1 2
```

```
matriz[1:2,1:2]
```

```
      [,1] [,2]  
[1,]    1    4  
[2,]    2    5
```

# Indexação - matrizes e data frames

---

`iris[1:5,c(2,4)]`

	Sepal.Width	Petal.Width
1	3.5	0.2
2	3.0	0.2
3	3.2	0.2
4	3.1	0.2
5	3.6	0.2

`iris[1:5,c("Sepal.Width","Petal.Width")]`

	Sepal.Width	Petal.Width
1	3.5	0.2
2	3.0	0.2
3	3.2	0.2
4	3.1	0.2
5	3.6	0.2

# Indexação - matrizes e data frases

---

```
iris[iris$Sepal.Length>7.0,c("Sepal.Width","Petal.Width")]
```

	Sepal.Width	Petal.Width
103	3.0	2.1
106	3.0	2.1
108	2.9	1.8
110	3.6	2.5
118	3.8	2.2
119	2.6	2.3
123	2.8	2.0
126	3.2	1.8
130	3.0	1.6
131	2.8	1.9
132	3.8	2.0
136	3.0	2.3

Funções básicas no R que ajudam a entender o formato dos dados



# Funções básicas no R

---

`class()` # a classe do objeto

`names()` # o(s) nome(s) de um objeto

`str()` # mostra a estrutura do objeto

`attributes()` # atributo do objeto (lista)

`dim()` # dimensões do objeto

`nrow()` ; `ncol()` # número de linhas/colunas de uma matriz

`dimnames()` # nome da dimensão do objeto

`rownames()` ; `colnames()` # nome de linhas e colunas

Vamos para a prática!