

Universidade Federal do Rio Grande do Norte  
Instituto Metr pole Digital  
**IMD0601 - Bioestat stica**

# R e Rstudio

---

Prof. Dr. Tetsu Sakamoto  
Instituto Metr pole Digital - UFRN  
Sala A224, ramal 182  
Email: [tetsu@imd.ufrn.br](mailto:tetsu@imd.ufrn.br)



## Baixe a aula (e os arquivos)

- Para aqueles que não clonaram o repositório:

```
> git clone https://github.com/tetsufmbio/IMD0601.git
```

- Para aqueles que já tem o repositório local:

```
> cd /path/to/IMD0601
```

```
> git pull
```



# R

IMD0601 - Bioestatística



Ross Ihaka



Robert Gentleman

- Linguagem de programação + Ambiente de desenvolvimento para análises estatísticos;
- Criado por **Ross Ihaka** e **Robert Gentleman** da Universidade de Auckland, Nova Zelândia em 1995;
- Baseado na linguagem **S**;



# R

IMD0601 - Bioestatística



Ross Ihaka



Robert Gentleman

## Características do R:

- Linguagem interpretada;
- Objeto orientado;
- Amplamente utilizado entre estatísticos e data miner;
- Ecossistema rica para realizar análise de dados;
- Open-source;
- Permite operações de matrizes;
- Principal diferencial: **análises estatísticas e visualização dos dados.**



# Rstudio

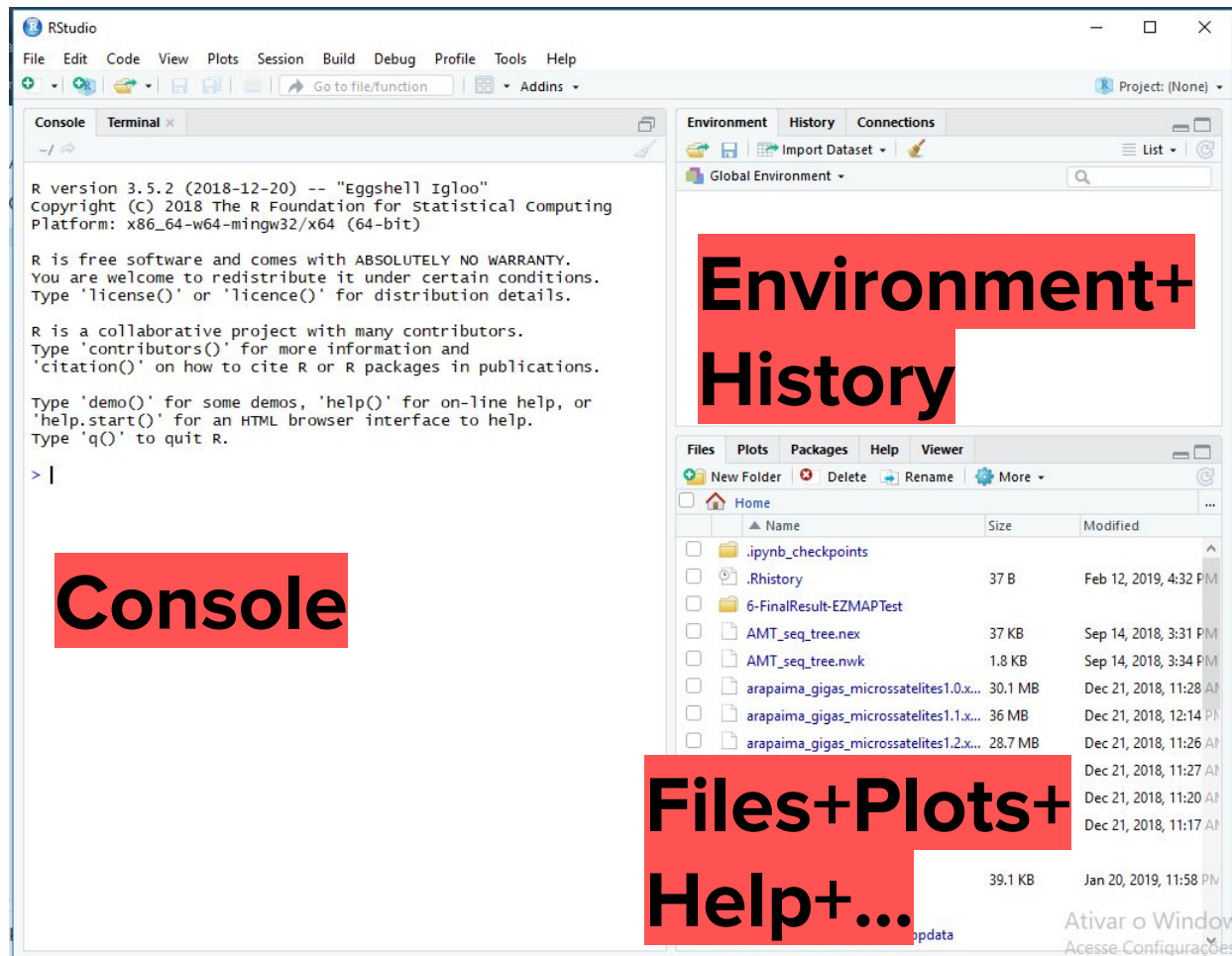
IMD0601 - Bioestatística



J J Allaire

- IDE para R;
- Criado por **Joseph J. Allaire**;
- Primeira versão em 2011;
- Versões gratuitos e pagos.

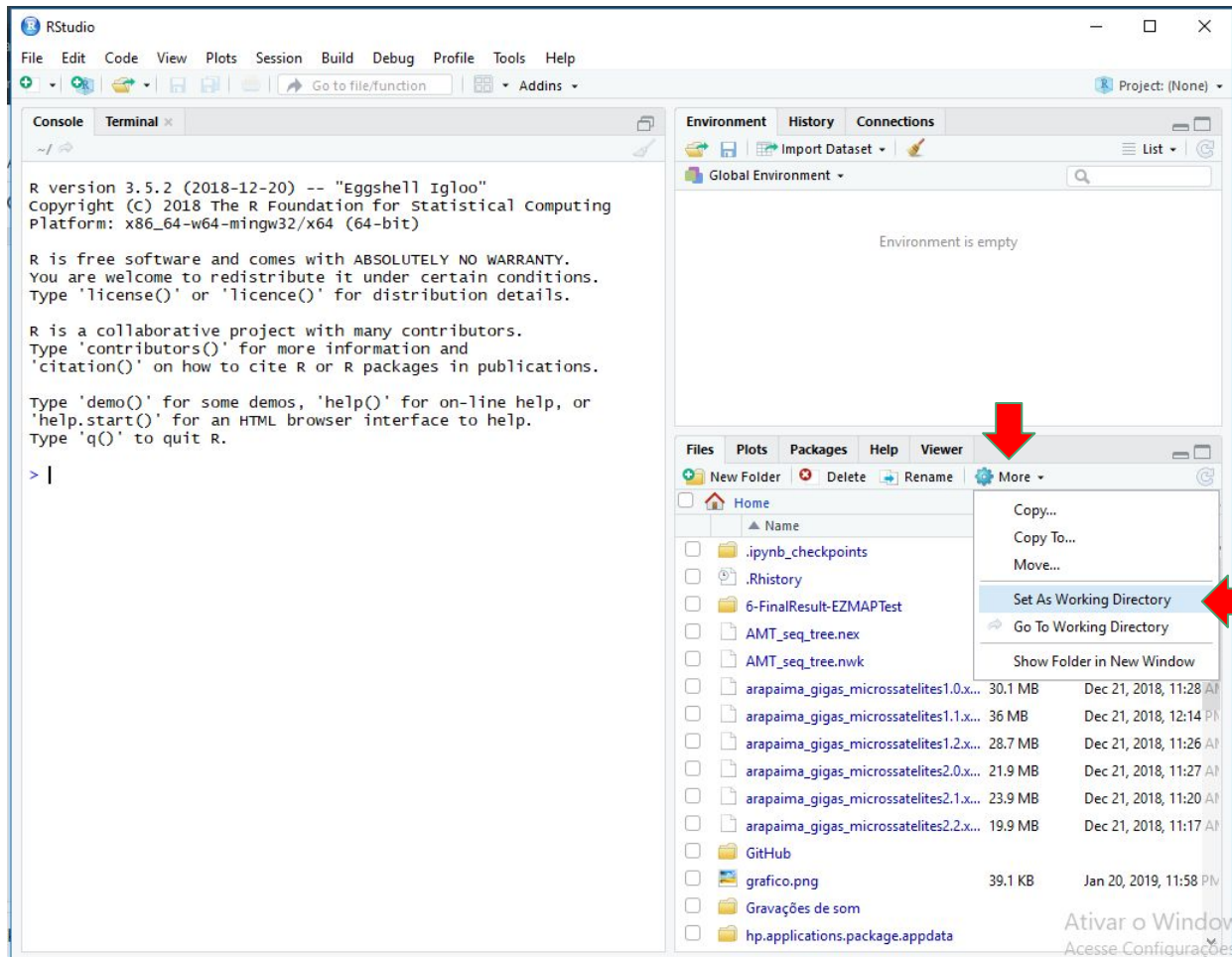
# Rstudio



# Rstudio

## Working directory

- **getwd()** e **setwd();**
- Diretório onde o R utilizará como referência para encontrar e salvar arquivos.
- Navegue na aba “Files” até a pasta desejada;
- Clique em “More” > “Set as Working Directory”;

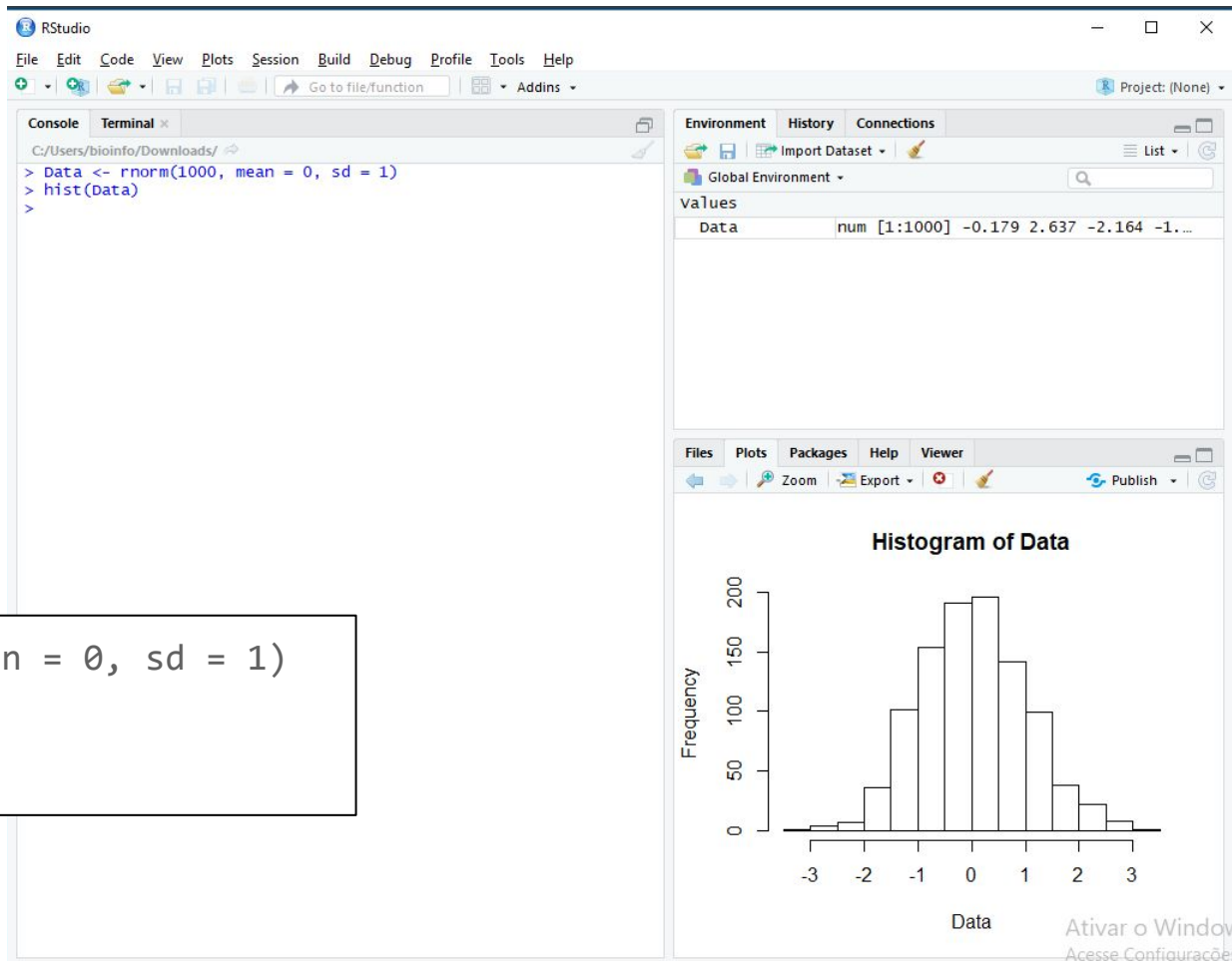


# Rstudio

## Gráficos

- Gráficos gerados aparecem na aba plot;
- Utilize os seguintes comando para gerar um gráfico:

```
> Data <- rnorm(1000, mean = 0, sd = 1)  
> hist(Data)
```





# Rstudio

## Instalação de pacotes

- **install.packages();**
- Para obter funções mais específicas;
- Vá na aba “Packages” > “Install” e procure pelo pacote “plm” e o instale.

The screenshot shows the RStudio interface with the following components:

- Console:** Displays the R code `Data <- rnorm(1000, mean = 0, sd = 1)` and `hist(Data)`.
- Environment:** Shows the 'Global Environment' with a variable 'Data' of type 'num' [1:1000].
- Packages:** The 'Install' button is highlighted. A dialog box for 'Install Packages' is open, showing the search results for 'plm'.
- System Library:** A list of installed packages is shown, including 'boot', 'class', 'cluster', 'codetools', 'compiler', 'datasets', 'foreign', 'graphics', 'grDevices', 'grid', 'KernSmooth', 'lattice', and 'MASS'.

Red numbers 1, 2, and 3 are overlaid on the image to indicate the steps:

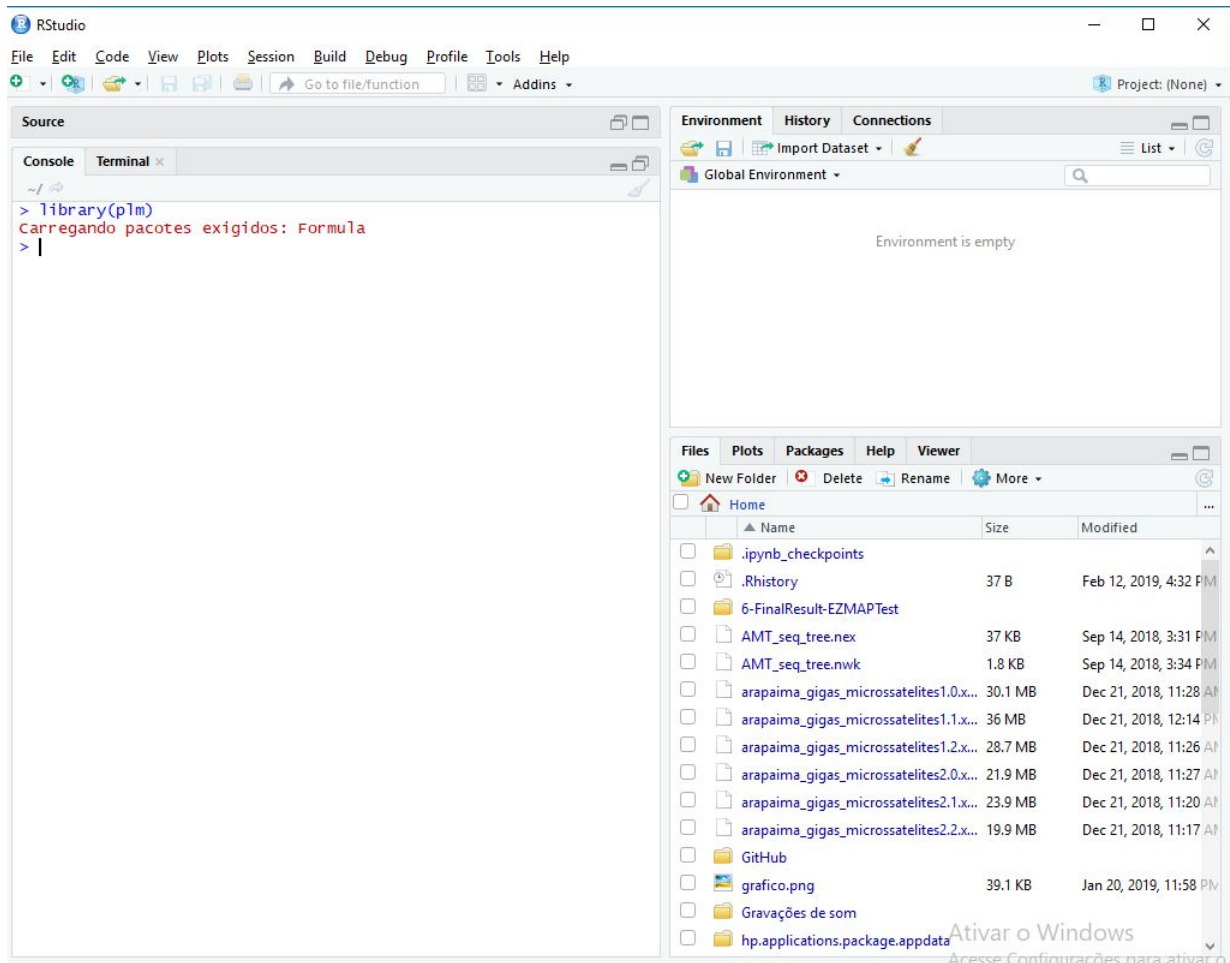
- 1: Points to the 'Packages' tab.
- 2: Points to the 'Install' button.
- 3: Points to the 'plm' package in the search results.

# Rstudio

## Carregar pacotes

- Após a instalação do pacote, você pode carregá-lo usando o comando:

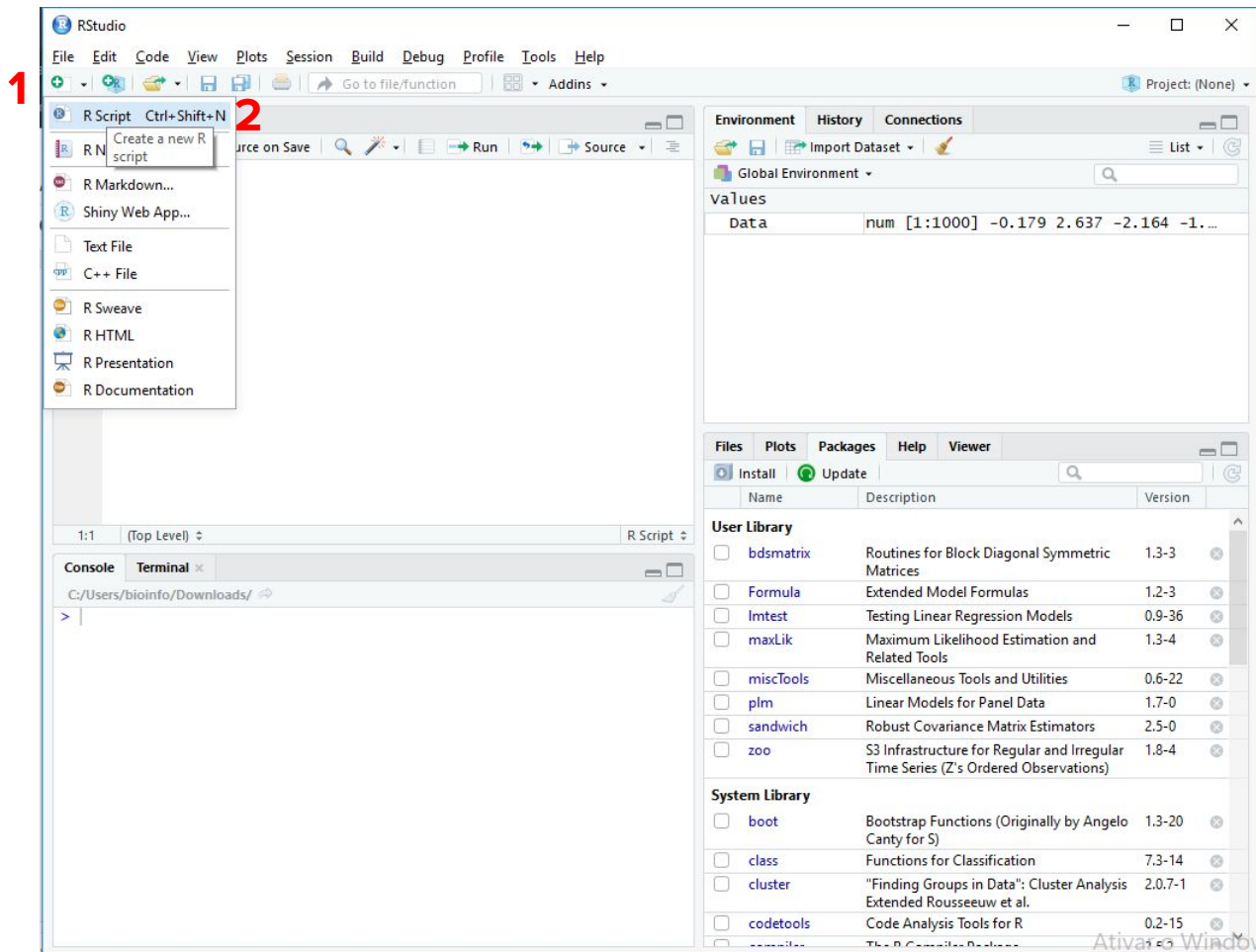
```
> library(plm)
```



# Rstudio

## Scripts

- Forma mais eficiente e prático de usar o R no Rstudio;
- Vá em “File” > “New File” > “R Scripts”.
- A interface fica dividida agora em 4 partes.



# Rstudio

## Scripts

- Coloque estes comandos no R Script e os execute:

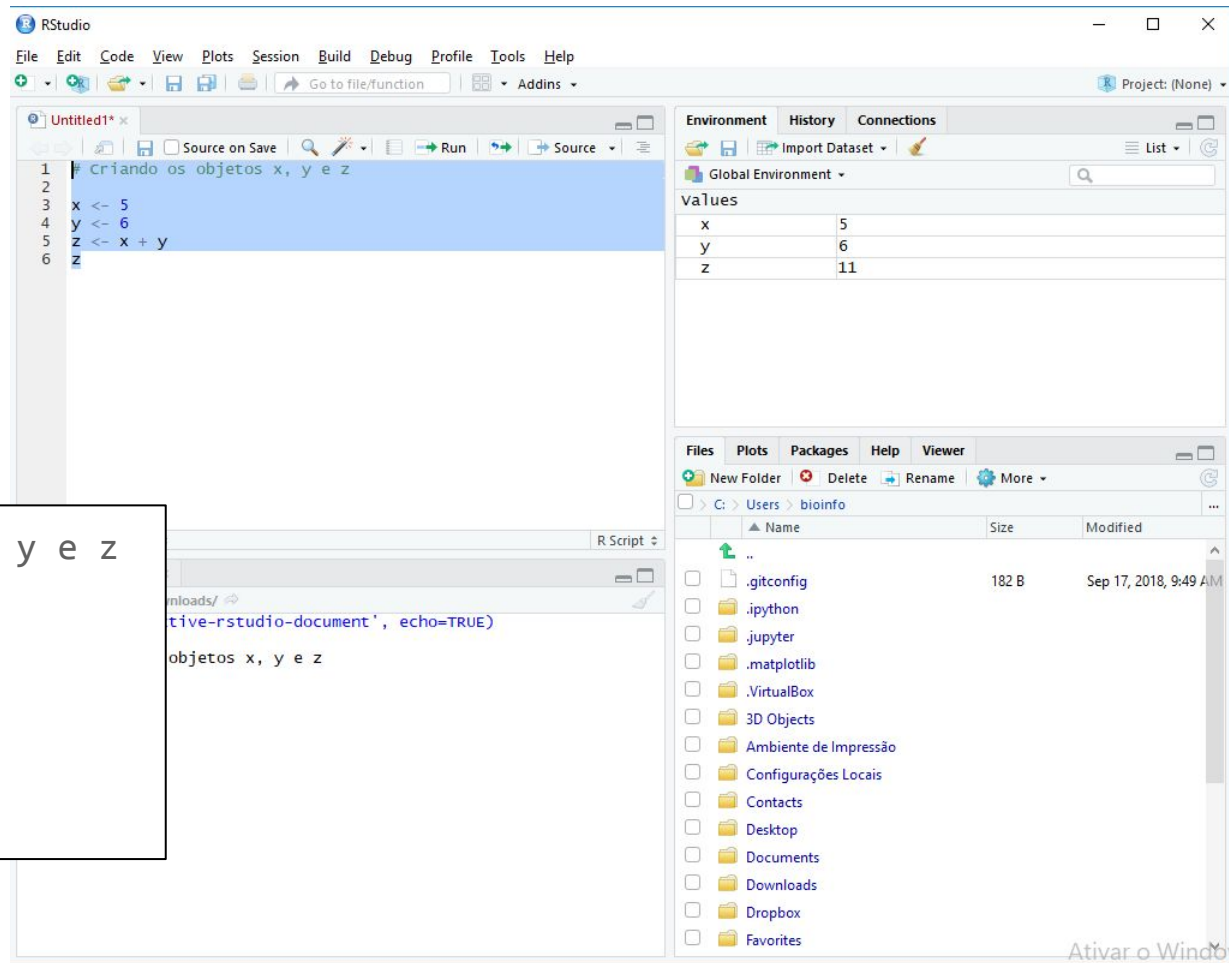
```
# Criando os objetos x, y e z
```

```
x <- 5
```

```
y <- 6
```

```
z <- x + y
```

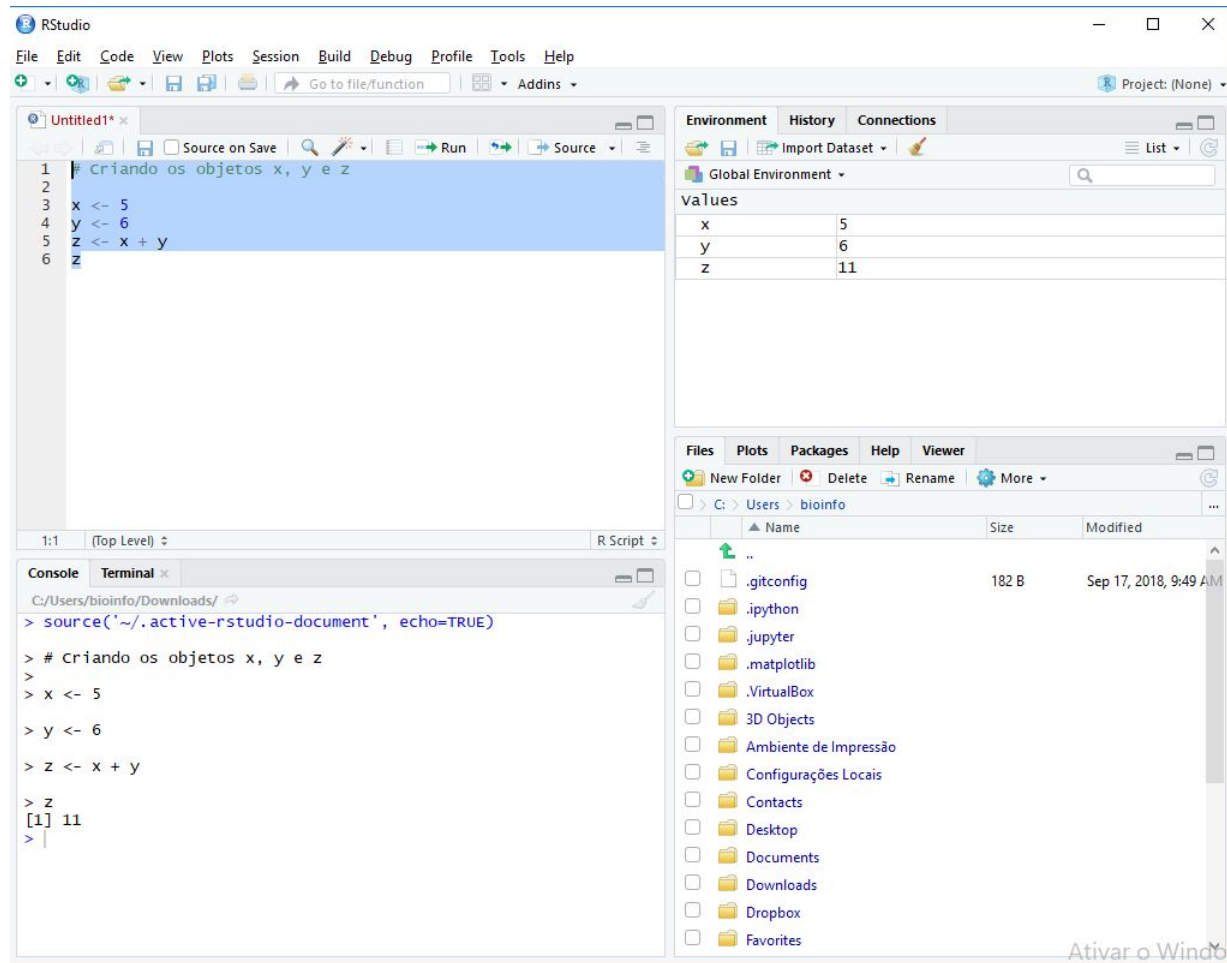
```
z
```



# Rstudio

## Scripts

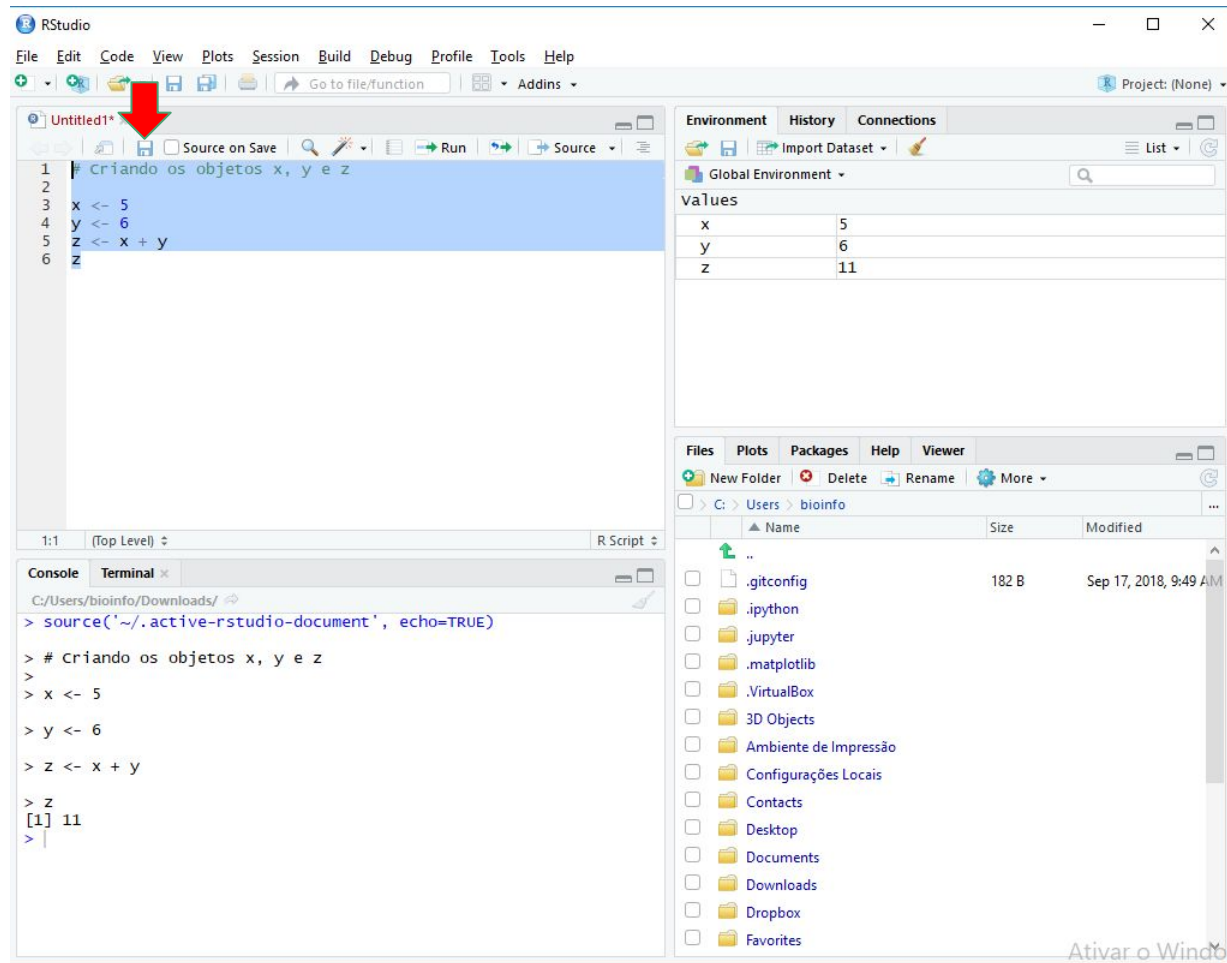
- Selecione as linhas que queira executar no Console e dê **Ctrl+Enter**;
- Ctrl+1, Ctrl+2 → move entre os scripts.



# Rstudio

## Scripts

- Para salvar o script, clique no ícone do disquete.
- A extensão padrão dos R Scripts é .R



# Aprendendo R

---

# Básico do R

Sinal de atribuição      `<-` (= aceito apenas no Rstudio) (Alt + -)

## Tipos de variáveis

|                           |             |               |
|---------------------------|-------------|---------------|
| character                 | "Char"      | # character() |
| numeric (real ou decimal) | 15.5, 2     | # numeric()   |
| integer                   | 2L, 10L     | # integer()   |
| logical                   | TRUE, FALSE | # logical()   |
| complex                   | 1+4i        | # complex()   |



# Básico do R

Funções:

|                           |   |
|---------------------------|---|
| <code>typeof()</code>     | Verifica o tipo de um dado (baixo nível)  |
| <code>class()</code>      | Verifica a classe de um dado (alto nível) |
| <code>length()</code>     | Retorna o tamanho ou dimensão do dado     |
| <code>attributes()</code> | Verifica se ele possui algum metadado     |
| Ajuda                     | <code>?attributes</code>                  |

# Básico do R

## Estrutura dos dados:

|               |   |
|---------------|---|
| Atomic vector | Vetor que acomoda um único tipo de dado;    |
| List          | Vetor que acomoda múltiplos tipos de dado;  |
| Matrix        | Matriz que acomoda um único tipo de dado;   |
| Data frame    | Matriz que acomoda múltiplos tipos de dado; |
| Factor        | Vetor que representa dados categóricos;     |

# Básico do R

Vetor numérico     `c(1,2,3,4)`

Vetor lógico     `c(TRUE, FALSE, NA)`

Vetor de caracteres     `c("My", "name", "is")`

Operações aritméticas     `c(1.1, 9, 3.14)*2+100 # 102.20 118.00 106.28`

Reciclagem     `c(1,2,3,4) + c(0,10) # 1 12 3 14`

# Básico do R

Sequência de números

```
1:20; 15:1; seq(1,20, by=0.5);  
seq(5,10,length=30)
```

Replicar números

```
rep(0, times=40); rep(c(0,1,2), times=10)
```

Tamanho do vetor

```
length(rep(0, times=40))
```

Juntar palavras

```
paste(c("My", "code"), collapse = " ")
```

# Básico do R

NA - not available

```
x <- c(44, NA, 5, NA)
```

```
3 * x
```

```
is.na(x) # mesmo que x == NA?
```

NaN - not a number

```
0 / 0
```

```
Inf - Inf
```

# Subconjunto de vetores

**No R, o índice se inicia do 1.**

|                             |  |
|-----------------------------|--|
| <code>x[1:10]</code>        | Extraí os 10 primeiros elementos de <code>x</code> ;         |
| <code>x[c(3,5,7)]</code>    | Extraí o 3º, 5º e o 7º elemento de <code>x</code> ;          |
| <code>x[c(-3,-5,-7)]</code> | Extraí os elementos de <code>x</code> menos o 3º, 5º e o 7º; |
| <code>x[-c(3,5,7)]</code>   | Extraí os elementos de <code>x</code> menos o 3º, 5º e o 7º; |
| <code>x[c(-3,5,7)]</code>   | <b>Erro, não misture índices positivos e negativos</b>       |
| <code>x[x &gt; 0]</code>    | Extraí elementos positivos de <code>x</code> ;               |
| <code>x[is.na(x)]</code>    | Extraí elementos NA de <code>x</code> ;                      |
| <code>x[!is.na(x)]</code>   | Extraí os elementos não NA de <code>x</code> ;               |

|                      |  |
|----------------------|--|
| <code>x[0]</code>    | Índice fora do intervalo ( <code>numeric(0)</code> ) |
| <code>x[3000]</code> | Índice fora do intervalo (NA)                        |

# Nomeando os elementos dos vetores

```
vector <- c(140, -50, 20, -120, 240)
days <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
names(vector) <- days
attributes(vector)
```

```
vector[2]           # -50
vector["Tuesday"]    # -50
vector[c(3,5)]       # (20,240)
vector[c("Wednesday", "Friday")] # (20,240)
```

```
vect <- c(foo = 11, bar = 2, norf = NA)
vect["foo"]          # 11
```

# Loops

For loops:

```
a <- 1:10
```

```
b <- 1:10
```

```
res <- numeric(length = length(a))
```

```
for (i in seq_along(a)) {
```

```
  res[i] <- a[i] + b[i]
```

```
}
```

```
res
```



Onde aprender R?



DataCamp



Udemy

coursera



Studio®

{swirl}



- <https://swirlstats.com/>
- Aprender R no próprio R!
- Instale o pacote “swirl” pelo RStudio (`> install.packages(“swirl”)`);
- Carregue o pacote “swirl” (`> library(swirl)`);
- Execute a função `swirl()`;



- Escolha o curso: R programming;
- Faça as tarefas:
  - 1. Basic Building Blocks
  - 2: Workspace and Files
  - 3. Sequences of Numbers
  - 4. Vectors
  - 5: Missing Values
  - 6: Subsetting Vectors