

MBA
USP
ESALQ

Introdução à Linguagem R

Rafael de Freitas Souza



Por Que Aprender uma Linguagem de Programação?

- Aprender a programar é muito importante quando desejamos entender os dados! Mais: é importante para que possamos interpretar esses dados; para que possamos transformá-los em informação!
- Se você trabalha ou deseja trabalhar com dados, programar é uma skill de extrema relevância.

Uma pergunta justa seria:

“Eu já trabalho ou desejo trabalhar com dados, e já possuo acesso a eles via Windows, OSx, Android, iOS, Chrome, Mozilla, Edge, Safari... Ainda assim, eu preciso aprender a programar?”

As Limitações de uma *Graphic User Interface* (GUI)



Reprodutibilidade

Automação

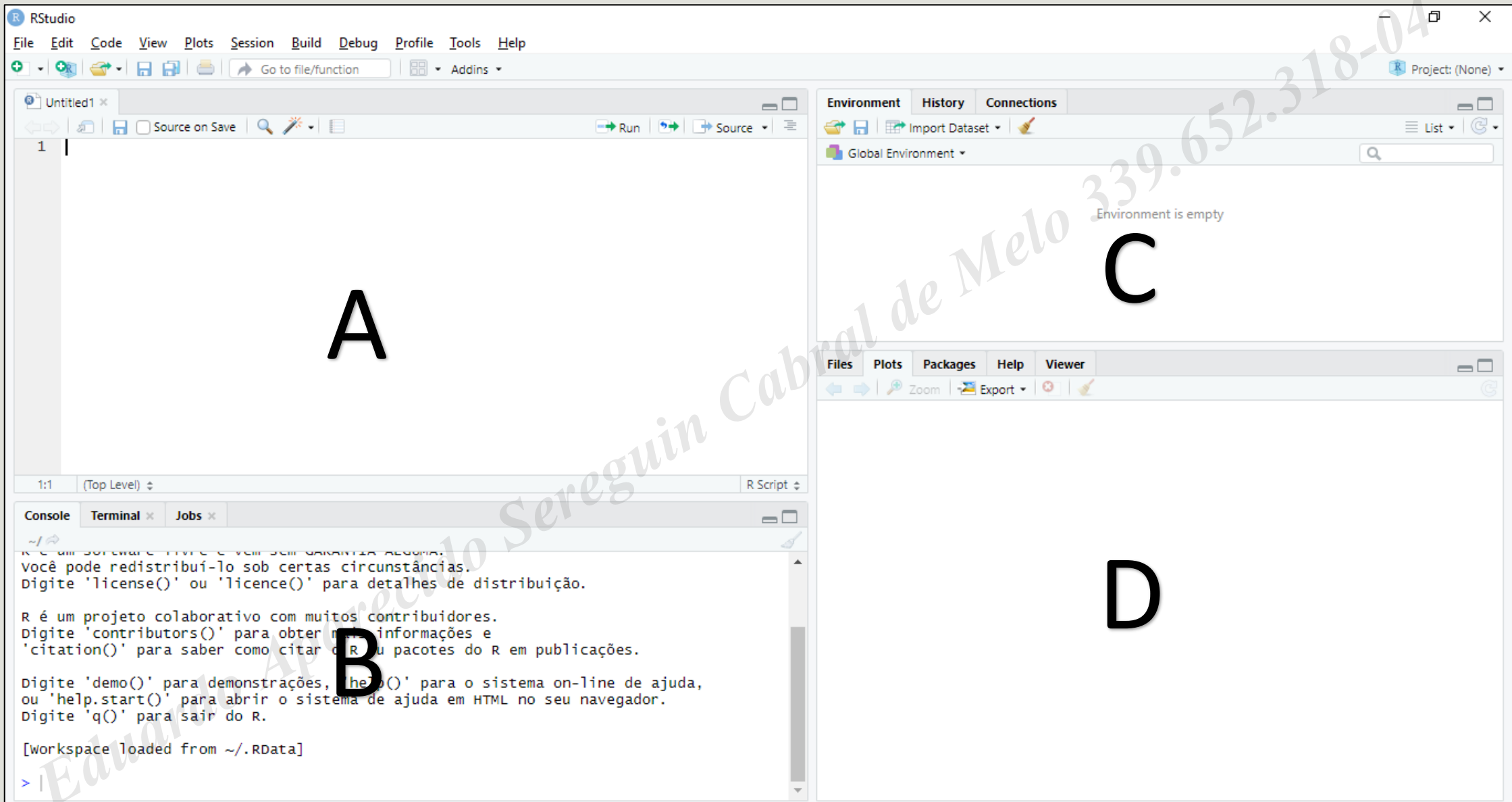
Comunicação

Por Que o R?



Sobre o R

- A linguagem R surgiu em 1995, derivada da linguagem S, e é orientada a objetos.
- Possui inúmeros pacotes (mais de 17 mil), com vantagem para a aplicação da Estatística Avançada e uma vasta comunidade de suporte, além de fortes capacidades voltadas ao Data Science.
- Comprehensive R Archive Network (CRAN) é o repositório da linguagem R em que cada usuário pode contribuir com novos pacotes (coleções de funções em R com código compilado). Esses pacotes podem ser facilmente instalados com uma linha de código.
- Leituras recomendadas para quem não ainda não conhece a linguagem R e deseja se aprofundar:
 - Hands-On Programming with R (Grolemund, 2014);
 - R for Data Science (Wickham & Grolemund, 2016);
 - Ciência de Dados com R (Oliveira, Guerra & McDonnell, 2018).



Objetos, Funções e Argumentos

Há autores que definem os objetos do R como sendo uma variável. Para o curso, entenderemos que variáveis correspondem a características de uma dada amostra ou população.

- Objetos são maneiras simples de se acessar algo que foi salvo na memória da máquina. Pode ser um valor, uma palavra, uma ou mais variáveis, uma URL, uma base de dados amostral ou populacional, uma lista de coisas distintas contendo informações e tamanhos distintos, um gráfico, um mapa, uma imagem, um novo comando, etc. **No R TUDO é um objeto!** Cada um desses objetos possui uma classe!
- Funções correspondem a ações, a ordens direcionadas à máquina;
- Argumentos correspondem a um refinamento ou um melhor direcionamento das ações ou ordens propostas pelas funções.

Criando um Objeto no R

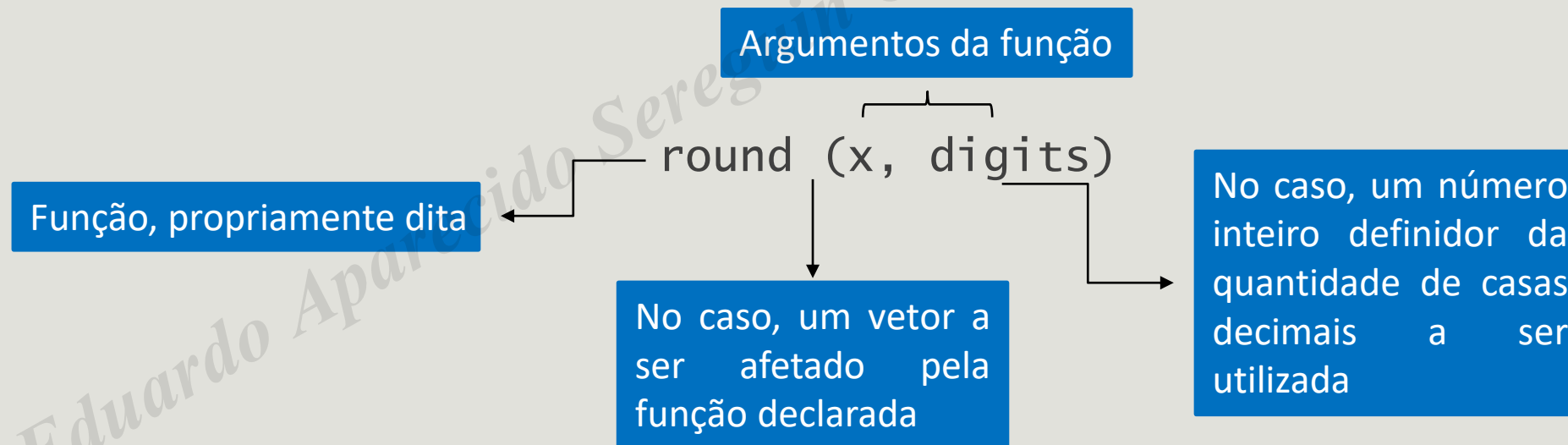
- Há duas formas de se criar um objeto na linguagem R:
 - Utilizando-se o símbolo =; ou
 - Utilizando-se o símbolo <- chamado operador de atribuição (prefira essa forma, reservando o símbolo = para as atribuições de valor dos argumentos das funções ou para operações matemáticas). O sinal de atribuição pode ser rapidamente declarado ao se pressionar conjuntamente as teclas 'Alt' e '-'.
- Os nomes dos objetos estabelecidos em R devem seguir certas regras:
 1. Não devem começar por números e nem por pontos;
 2. Não é desejável que contenham espaços, muito embora a situação seja possível com o uso dos símbolos ``;
 3. Suas nomenclaturas também não aceitam caracteres especiais, como: !, ~, \$, @, +, -, /, *.
 4. Recomenda-se evitar nomear os objetos com os mesmos nomes de funções já estabelecidas;
 5. Recomenda-se não utilizar acentos e, sempre que possível, evitar as letras maiúsculas, visto que a linguagem é *case sensitive*.

Principais Funções Introdutórias do Curso

Função	Serve para:
args()	Verificar os argumentos de uma dada função no R
round()	Arredondar números
sample()	Criar amostras
class()	Verificar as classes dos objetos do R
View()	Visualizar objetos em forma de planilha
head()	Visualizar as primeiras observações de uma base de dados
tail()	Visualizar as últimas observações de uma base de dados
str()	Observar a estrutura de uma base de dados
length()	Observar o comprimento de um vetor ou de uma lista de dados
dim()	Descobrir as dimensões de um objeto
nrow()	Contar o número de linhas de uma base de dados
ncol()	Contar o número de colunas de uma base de dados
rm()	Remover um objeto do ambiente de trabalho
install.packages()	Instalar pacotes
library()	Carregar pacotes

Utilizando funções no R

Para utilizar uma função no R, devemos conhecer sua forma funcional, isto é, devemos, em regra, declarar os argumentos inerentes a ela. Exemplo de utilização da função `round()`:



Pacotes na Linguagem R

- A linguagem R possui milhares de pacotes direcionados às mais diversas áreas do conhecimento, e a maioria não está instalada em nossos computadores. Para instalar um pacote, devemos comandar:

```
install.packages("nome do pacote aqui")
```

- A instalação de um pacote não basta para sua utilização. Assim, a cada nova seção aberta do RStudio, devemos chamá-los da seguinte maneira:

```
library(nome do pacote aqui)
```

Criando e Excluindo Variáveis em um *Dataset*

- Criando uma variável:

```
basededados$nova_variável <- NA
```

ou

```
mtcars["nova_variável"] <- NA
```

- Excluindo uma variável:

```
basededados$nova_variável <- NULL
```

Extraindo Valores de um Dataset

- Para extrair uma coluna de um *dataset*, utilize o operador \$:

`basededados$nova_variável`

- De maneira mais precisa, também se pode extrair valores do *dataset* com o operador `[,]`:

`basededados[,]`

Declaração de
qual linha se
quer acessar

Declaração de
qual coluna se
quer acessar

Funções `if`, `else` e `ifelse`:

```
if(teste lógico){
```

caso a resposta do teste lógico seja TRUE, faça isso

```
} else {
```

caso a resposta do teste lógico seja FALSE, faça essa outra coisa

```
}
```

```
ifelse(teste lógico,
```

yes = caso a resposta do teste lógico seja TRUE, faça isso ,

no = caso a resposta do teste lógico seja FALSE, faça essa outra coisa)

Funções `for`, `while` e `repeat`

```
y <- 10
```

```
for(i in 1:5){  
  print(y + i)  
}
```

Para cada `i` (poderia ser qualquer símbolo ou palavra), presente na sequência de 1 a 5, imprima o valor da soma entre `y` e `i`

Eduardo Aparecido Sereguin Cabral de Melo 339.652.318-04

Funções `for`, `while` e `repeat`

```
z <- 0
```

```
while (z < 10) {  
  print(z)  
  z <- z + 1  
}
```

Enquanto `z` for menor do que 10, imprima `z` e, depois, atualize o valor de `z` acrescentando o seu valor em uma unidade

Eduardo Aparecido Sereguin Cabral de Melo 339.652.318-04

Funções `for`, `while` e `repeat`

```
w <- 3
```

```
repeat{  
  print(w)  
  w <- w + 2  
  if(w > 18) break()  
}
```

Repita os passos a seguir:

- Imprima o valor de `w`;
- Atualize o valor de `w`, acrescentando em duas unidades;
- Se `w` passar a ser maior do que 18, pare tudo.

Visualização de Dados com o ggplot2

- A sintaxe mais básica do pacote ggplot2, para a criação de um gráfico a partir de um *data frame*, é a seguinte:

```
ggplot(data = base de dados aqui) +  
  geom_geometria escolhida aqui(aes(principais elementos do gráfico aqui))
```

Obrigado!



Rafael de Freitas Souza

[Linkedin](#)