

Prática no R! Roteiro 1 - Introdução ao software R/RStudio

Elaborado por Luis Felipe Bortolatto da Cunha

21 de setembro de 2020

Contents

1. Introdução	1
2. O que é o R?	1
3. O que é o RStudio?	2
4. Instalando o R e RStudio	2
5. Interface RStudio	2
6. Objetos	2
7. Funções	4
8. Operadores matemáticos	4
9. Operadores lógicos	5
10. Como buscar ajuda	5

1. Introdução

Você pode baixar este roteiro em formato PDF neste endereço.

Ao longo deste curso, os softwares R/RStudio serão usados como uma **ferramenta** para auxiliar na aplicação de **métodos e técnicas de análise da informação no planejamento**, uma etapa fundamental em pesquisas que buscam o aumento da compreensão de questões no nosso campo de interesse.

É importante ressaltar o uso do R/RStudio não pode ser dissociado do **processo de pesquisa**, que envolve a observação, formulação de hipóteses, coleta de dados e **análise de dados**, sendo este o foco deste curso.

2. O que é o R?

O **R** é um **programa estatístico** muito usado na **análise e visualização de dados**, onde a interação é feita através de uma **sintaxe** (linguagem de programação).

Algumas de suas vantagens são:

- É **gratuito**, de **distribuição livre** e de **código aberto**
- Possui uma **curva de aprendizado bastante amigável**, sendo muito fácil de se aprender (se comparado a outros programas estatísticos)
- Todas as rotinas criadas em R são **transparentes**, seguindo os princípios da ciência aberta

- Existe uma enorme quantidade de **pacotes** gratuitos, para diversos tipos de aplicações
- Existe uma enorme quantidade de **ajuda e tutoriais** disponíveis gratuitamente na internet
- É amplamente utilizado pela **comunidade acadêmica** e mercado

Um **script** refere-se à uma rotina de análise ou visualização de dados que:

- É um texto
- É reproduzível
- Pode ser compartilhado
- Pode ser colaborativo

3. O que é o RStudio?

O **RStudio** é o **ambiente de desenvolvimento integrado** mais usado na análise de dados com a linguagem de programação R. Além de permitir **escrever códigos em scripts** e **executar códigos no console**, é possível visualizar gráficos e mapas (estáticos ou dinâmicos), gerenciar pacotes, navegar pelos arquivos, dentre outras funções.

4. Instalando o R e RStudio

Primeiro, é necessário instalar o R. Você pode transferir a última versão neste endereço. Após a transferência, siga o passo a passo para a instalação.

Se a instalação do R foi bem sucedida, você pode transferir o RStudio Desktop (versão gratuita) neste endereço e seguir o passo a passo para a instalação.

5. Interface RStudio

A interface do RStudio deve exibir, na maior parte do tempo, quatro painéis:

1. Source: É a sua área de trabalho principal. Pode exibir várias janelas, com scripts, tabelas e outros conteúdos. Ao clicar em **Run** ou apertar os comandos **Ctrl + Enter**, dentro de um script, o conteúdo selecionado será enviado ao **Console** para ser executado.

2. Console: Todo código que você executar em um script será enviado ao **Console**, onde os resultados e avisos aparecerão. Você também pode escrever uma linha de código diretamente no **Console**.

3. Environment/History:

- **Environment:** É o ambiente de trabalho ativo. Todos os objetos e funções definidos devem aparecer no **Environment**.
- **History:** Exibe o histórico de cada linha de código executada, que podem ser enviadas à **Source** ou **Console**.

4. Files/Plots/Packages/Help:

- **Files:** Permite navegar pelos diretórios de trabalho.
- **Plots:** Exibe os gráficos, mapas e outras figuras produzidas.
- **Packages:** Permite o gerenciamento dos pacotes ativos e instalados.
- **Help:** Contém tutoriais e exibe ajuda, quando solicitada.

6. Objetos

Na linguagem de programação R, os objetos são a unidade básica de análise. Eles referem-se à um pequeno espaço na memória RAM do computador onde um conjunto de valores serão armazenados. Geralmente usamos o operador **<-** ou **=** para definir um objeto.

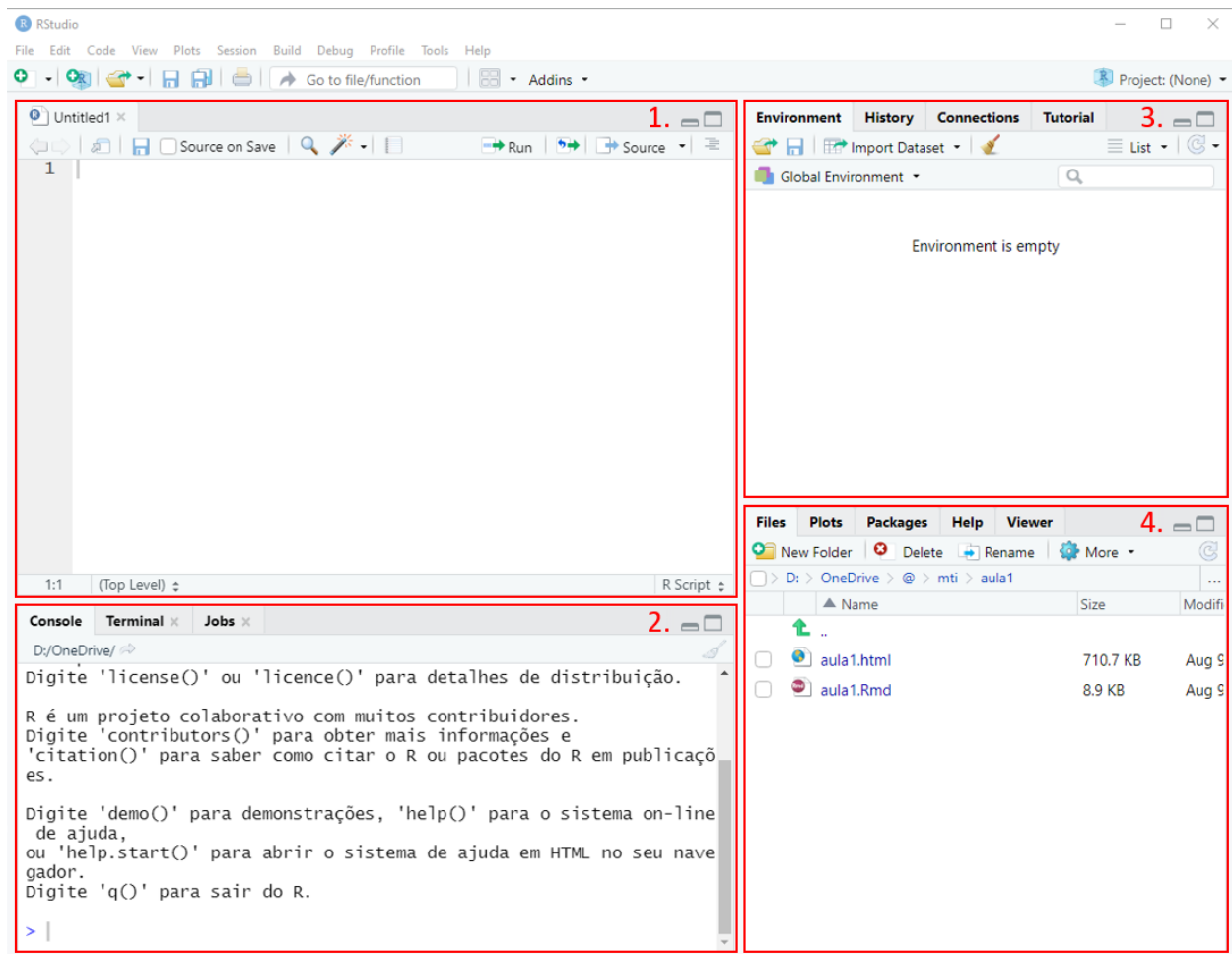


Figure 1: Interface RStudio

O R suporta diferentes tipos de objetos, sendo alguns dos mais comuns:

- Vetor (**vector**): um conjunto unidimensional de valores da mesma classe.
- Matriz (**matrix**): um conjunto bidimensional de valores da mesma classe (geralmente, numérico).
- Tabela (**data.frame** ou **tibble**): um conjunto bidimensional de valores, organizados em linhas e colunas. Todos os valores de uma coluna possuem a mesma classe.
- Raster (**raster**): uma matriz com atributos geoespaciais.
- Geoespacial (**sf**): uma tabela com atributos geoespaciais.
- Lista (**list**): um conjunto unidimensional de outros objetos (por exemplo, tabelas).

Nas primeiras aulas vamos explorar dados em formato de **tabela**! É nesse formato que os bancos de dados são geralmente disponibilizados na internet. As tabelas de bancos de dados sempre possuem uma estrutura comum: os valores são organizados em **linhas (observações)** e **colunas (variáveis)**, sendo consideradas um tipo de **dados estruturados**. As observações referem-se à unidade de análise (pessoas, domicílios, municípios etc.), enquanto as variáveis são os atributos dessa variável e possuem sempre a mesma classe (caractere, numérico, fator etc.).

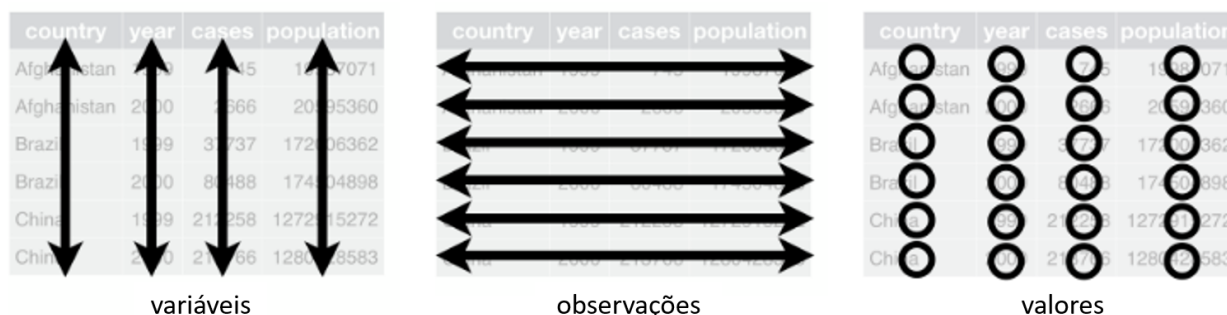


Figure 2: Dados estruturados. Fonte: adaptado de Grolemund & Wickham, 2017

7. Funções

Na linguagem de programação R, as **funções** referem-se à uma **sequência de comandos** que permitem **realizar operações complexas**. O R básico já vem com diversas funções estatísticas clássicas definidas:

- `sum()`: soma
- `mean()`: média
- `median()`: mediana
- `quantile()`: quantil
- `sd()`: desvio padrão
- `var()`: variância
- `sqrt()`: raiz quadrada
- `exp()`: exponencial
- `round()`: arredondamento

Novas funções podem ser adicionadas carregando pacotes.

8. Operadores matemáticos

Os **operadores aritméticos** já estão definidos na linguagem R, permitindo realizar operações com dados numéricos:

- Adição: +
- Subtração: -

- Multiplicação: `*`
- Divisão: `/`
- Módulo: `%%`
- Divisão por inteiro: `%/%`
- Exponenciação: `^` ou `**`

9. Operadores lógicos

Os **operadores lógicos** também já estão definidos na linguagem R, permitindo fazer perguntas sobre os dados e retornando sempre **TRUE** (verdadeiro) ou **FALSE** (falso).

- `x == y`: `x` é **igual** a `y`?
- `x != y`: `x` é **diferente** de `y`?
- `x > y`: `x` é **maior** que `y`?
- `x >= y`: `x` é **maior ou igual** a `y`?
- `x < y`: `x` é **menor** que `y`?
- `x <= y`: `x` é **menor ou igual** a `y`?
- `x & y`: `x` e `y` são verdadeiros?
- `x | y`: `x` ou `y` são verdadeiros?
- `!x`: **negativa** de `x`.
- `is.na(x)`: `x` é **igual a NA**?
- `!is.na(x)`: `x` é **diferente de NA**?

10. Como buscar ajuda

No decorrer das aulas práticas, você pode precisar de ajuda para entender uma função. Não se preocupe: ter dúvidas faz parte do processo de aprendizado. Na verdade, o R/RStudio conta com uma ferramenta que pode te auxiliar: é possível acessar tutoriais produzidos pelo R/RStudio na aba **Help**.

Help também exibe a documentação de cada função, que pode ser acessada com o uso das funções `help()` ou `?`.

Para acessar a documentação da função média, por exemplo, você pode digitar:

```
help(mean)
```

Ou:

```
?mean
```

Que deve exibir o seguinte conteúdo na aba **Help**:

- **Descrição:** Função genérica para a média aritmética (aparada).
- **Uso:** `mean(x, ...)`
- **Argumentos:**
 - **x** Um objeto R. Atualmente existem métodos para vetores numéricos / lógicos e objetos de data, hora e intervalo de tempo. Vetores complexos são permitidos apenas para `trim = 0`.
 - **trim** a fração (0 a 0,5) de observações a serem aparadas de cada extremidade de `x` antes que a média seja calculada. Os valores de corte fora desse intervalo são considerados como o ponto final mais próximo.
 - **na.rm** um valor lógico que indica se os valores de NA devem ser removidos antes que o cálculo continue.
 - ... argumentos adicionais passados para ou de outros métodos.
- **Valor:** Se o corte for zero (o padrão), a média aritmética dos valores em `x` é calculada, como um vetor numérico ou complexo de comprimento um. [...]

Importante: a seção `uso` descreve quais são os argumentos obrigatórios (apenas `x`, no caso da função `mean()`), sendo todos os outros argumentos descritos opcionais. Atente-se aos argumentos obrigatórios e opcionais ao usar uma função.

Infelizmente, a maior parte do conteúdo está escrito em inglês, assim como em outras linguagens de programação. :worried:

Você pode traduzir o conteúdo da documentação no tradutor de sua preferência, como fizemos para apresentar a documentação da função `mean()`.

Você também pode buscar ajuda em um buscador (Google, Bing, Yahoo etc.), pois já existem diversos materiais em português sobre a linguagem de programação R e análise de dados, mas estaremos disponíveis para tirar quaisquer eventuais dúvidas no decorrer deste curso.

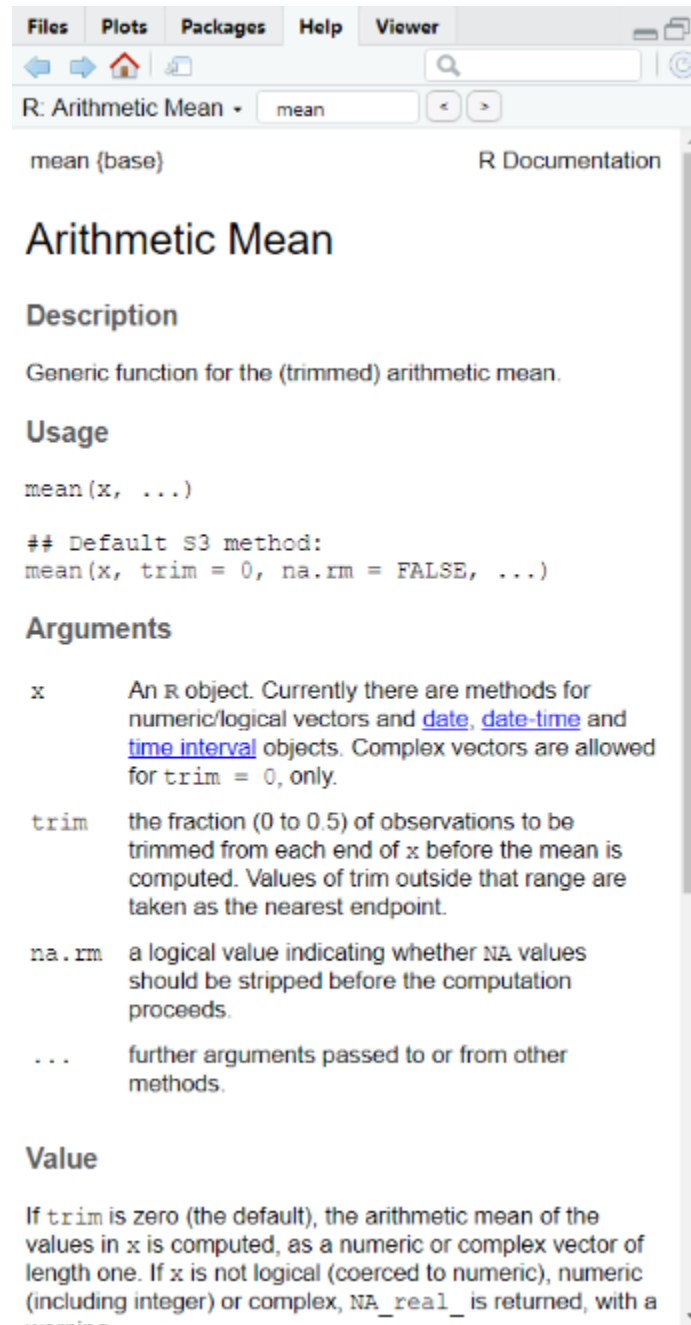


Figure 3: Documentação `mean()`