

Aula 1: apresentando o R

Marcelo Prudente e Rafael Giacomini

6 de março de 2018

- 1 Objetivo do curso
- 2 Porque usar o R?
- 3 R
- 4 Pacotes
- 5 Obtendo Ajuda

Objetivo do curso

- Ensinar
-

Porque usar o R?

- O **R** é uma linguagem e um ambiente para computação estatística e criação de gráficos. + linguagem orientada a objetos
- É uma implementação da linguagem S, criada por Jonh Chamber em 1976
- Criado por Ross Ihaka e Robert Gentleman na Universidade de Auckland

- É uma ferramenta aberta (de graça);
- Tem as mesmas funcionalidades do Excel, porém é mais poderoso;
 - ▶ Ferramenta estatística e de programação;
- **Gráficos** e **mapas** com excelente qualidade de publicação;
- Faz análises de dados e estatísticas facilmente.

- Ampla comunidade de contribuidores;
- É fácil reproduzir os comandos para outros bancos de dados (maior transparência nas análises);
- É possível abrir inúmeros bancos de dados ao mesmo tempo;
- **Rmarkdown**: linguagem de texto integrada ao R + elaboração de relatórios + elaboração de slides
- Ferramenta cada vez mais utilizada por grandes companhias;

- Por ser livre, não há garantias legais:
 - ▶ o CRAN atua como uma espécie de moderador do sistema
- Não lida bem com bases grandes (carrega dados na memória):
 - ▶ pacote `data.table` - desenhado para lidar com bases grandes
 - ▶ MonetDB e MonetDBLite() - soluções para conexão com bases grandes

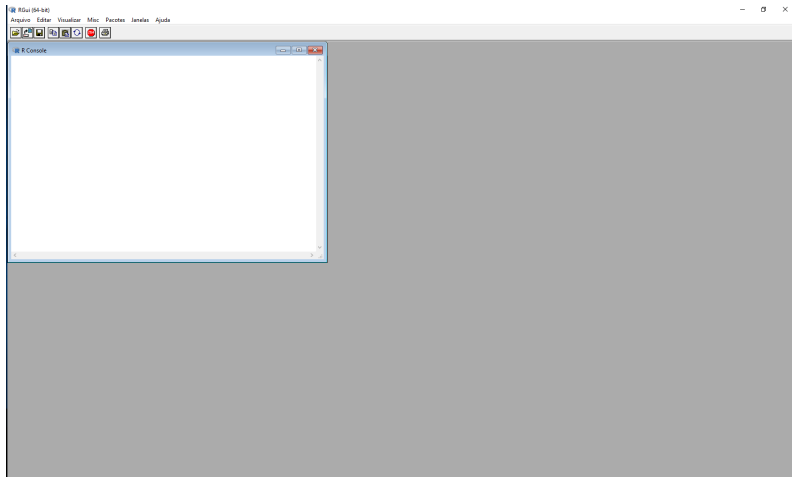
R

- O **R** é uma linguagem de programação voltada para a computação estatística;
- Acessamos o programa por meio de *linhas de comando*
- A abordagem de programação permite:
 - ▶ automatizar operações
 - ▶ documentar as operações realizadas (transparência e integralidade)

- O **R** é um conjunto integrado de instalações de softwares para manipulação de dados, cálculo e apresentação de gráficos.
 - ▶ não é apenas um programa estatístico
 - ▶ os pacotes do R podem expandir bastante as funcionalidades do programa
 - ▶ é possível escrever suas próprias funções

Como é o ambiente do R?

- O ambiente de trabalho não é tão amigável aos não programadores.



- Por isso, utilizaremos o **R Studio**

O **R Studio** integra o **R** como um Ambiente de Desenvolvimento Integrado, que apresenta maior funcionalidade. Por exemplo:

- Tem ferramentas que realçam a sintaxe e completam códigos;
- Executa códigos diretamente do editor

Para utilizar o **R Studio** é necessário ter o **R** previamente instalado no computador.

O ambiente de trabalho do R Studio

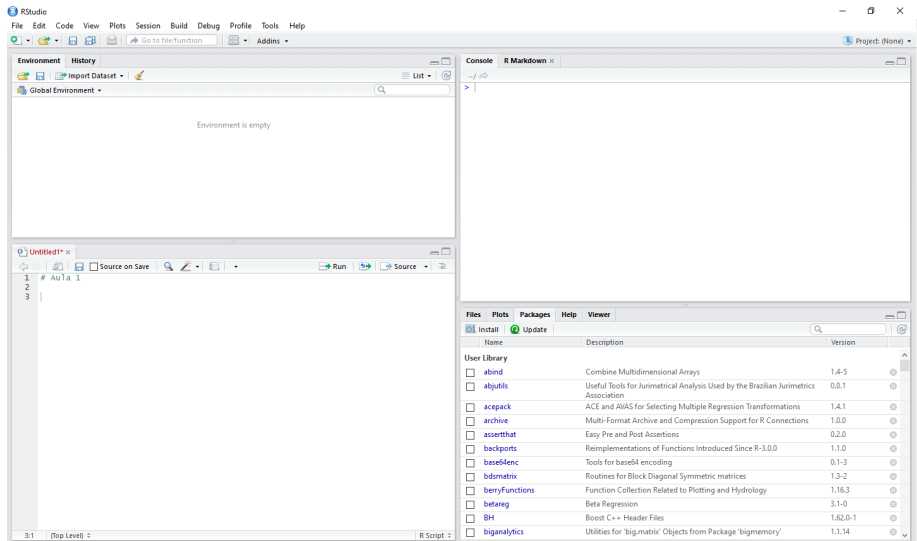


Figure 1: Ambiente do R Studio

- No *console* digitamos as nossas linhas de comando para o programa. Podemos realizar operações matemáticas, estatísticas, descrever funções entre outros.
- O símbolo $>$ indica que podemos digitar comandos ao programa.
- Em seguida, pressionamos **ENTER** para executá-los.

Podemos executar o seguinte comando no console:

```
50 * 15
```

```
## [1] 750
```

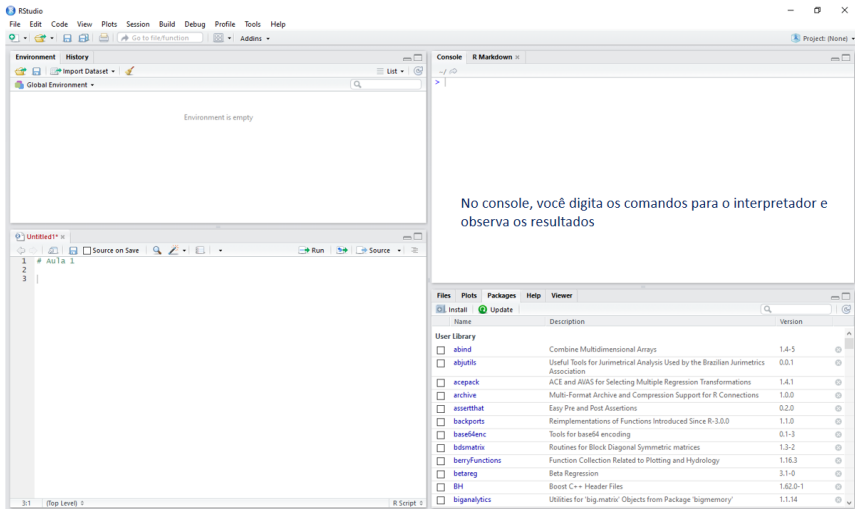



Figure 2: Console do R Studio

Script (1)

- Nos *scripts* passamos instruções que serão processadas no *console*.
- Porém, nesse caso gravamos os comandos em um arquivo de texto (.txt) que pode ser executado a qualquer momento:
 - ▶ reproducibilidade + transparência!
 - ▶ capacidade de executar diversas funções em projetos grandes
- Para executar um código do script, basta utilizar o atalho *Ctrl + Alt + ENTER*

Vamos digitar algumas linhas no *script* e, em seguida, selecionar os comandos e utilizar o atalho:

```
5 + 10
```

```
## [1] 15
```

```
"Boa tarde, pessoal"
```

```
## [1] "Boa tarde, pessoal"
```

Script (2)

- Os scripts permitem comentar os códigos com a #
 - ▶ reproducibilidade !

```
# Vamos escrever um texto  
"Para escrevermos um texto no R, precisamos das aspas"
```

```
## [1] "Para escrevermos um texto no R, precisamos das aspas"
```

- Os comentários podem demarcar sessões com o uso de quatro -

```
# Sessão 1: soma ----  
1 + 1
```

```
## [1] 2
```

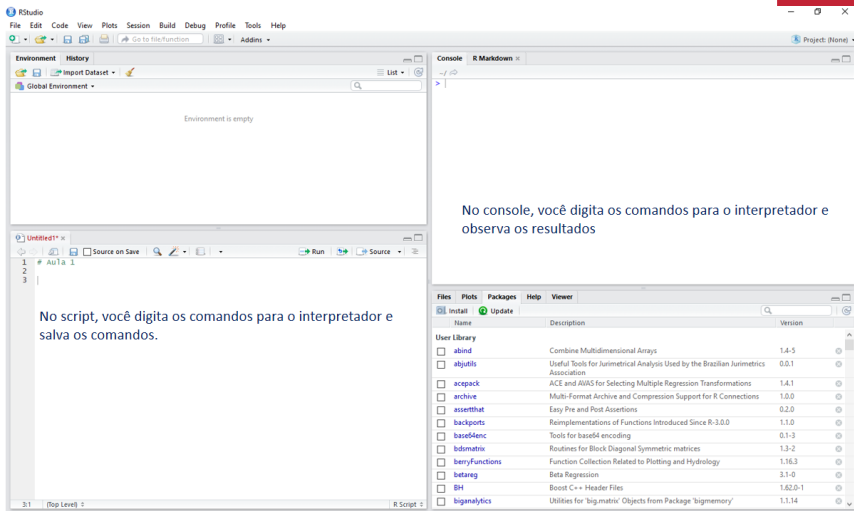


Figure 3: Console do R Studio

- A última parte do ambiente do **R** apresenta:
 - ▶ **Arquivos**: da pasta que estiver especificada no sistema + `getwd()`
 - ▶ **Plots**: mostra os gráficos eventualmente feitos
 - ▶ **Pacotes**: apresenta os pacotes instalados em nosso **R**
 - ▶ **Ajuda**: mais informações sobre o sistema

Pacotes

- Os pacotes do **R** empacotam(!!) códigos, dados, documentação e testes!
- Esses pacotes podem ser encontrados no CRAN ou no github
 - ▶ Há mais de 12.000 pacotes só no CRAN
- Para instalar pacotes do CRAN, basta ter conexão com a internet e digitar no *console* ou *script*:

```
# Instalando o pacote dplyr
install.packages("dplyr")
```

- Ou ainda, ir para a aba *packages* e, em seguida, em *install*

- Há também pacotes disponíveis no github. Para instalá-los, é necessário o pacote *devtools*

```
install.packages("devtools")  
devtools::install_github("tidyverse/readr")
```

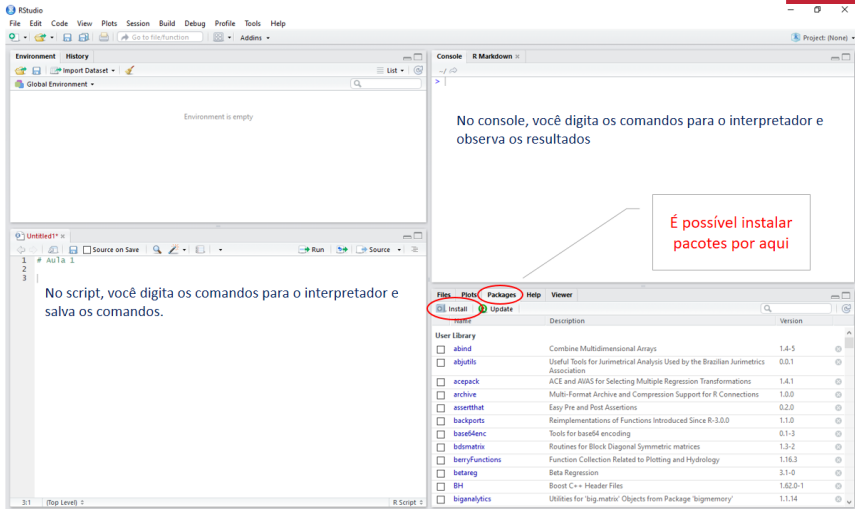



Figure 4: Console do R Studio

- Há três formas de colocar os pacotes em uso no R:
 - ▶ Clicando diretamente na caixa ao lado do nome do pacote
 - ▶ Ou utilizando dois comandos similares:

```
# Primeira forma  
require(dplyr)
```

```
# Segunda Forma  
library(dplyr)
```

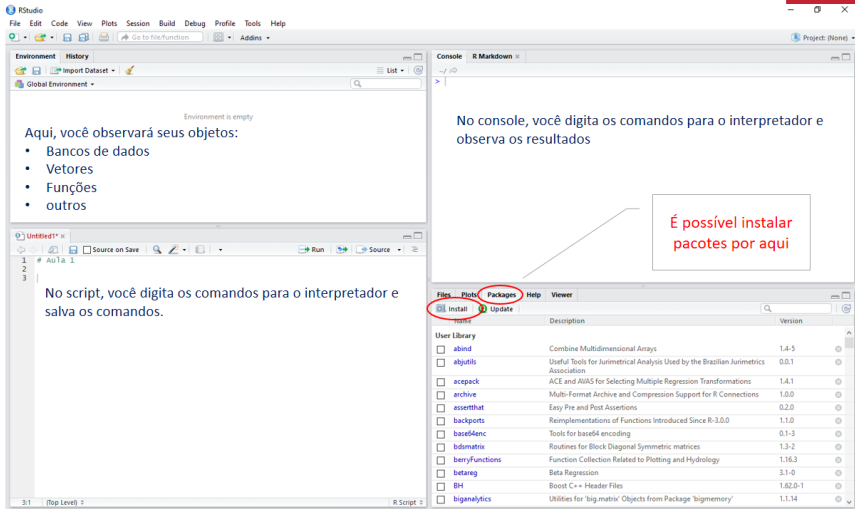


Figure 5: Console do R Studio

Obtendo Ajuda

- Em caso de dúvidas, o **R** possui uma extensiva documentação *offline* para auxiliá-lo a entender os comandos.

```
# Por exemplo, como funciona a função log?  
help(log)  
?log
```

- Para descobrir quais os argumentos das funções:

```
args(log)
```

- Ainda, é possível ver os exemplos do sistema para uma determinada função:

```
example(log)
```

- Se a ajuda do programa não esclarecer o problema, é possível encontrar fontes externas de ajuda na internet.
- Você pode procurar o seu problema no google, que provavelmente te retornará uma página:
 - ▶ **Stack Overflow**: o site tem uma extensa série de tópicos relacionados a problemas no **R**.
 - ▶ Provavelmente você encontrará sua resposta lá.