

# Introdução ao software R

Isabel de Sousa Amorim  
Paulo Henrique Sales Guimarães

Universidade Federal de Lavras - UFLA  
Departamento de Estatística - DES

2018

# R - Visão Geral

- 1 Software R - breve introdução
  - Primeira sessão com o R
  - Operações lógicas
  - Objetos
- 2 Estatística descritiva
- 3 Alguns gráficos no R

# Objetivos

Este curso tem como objetivo proporcionar um ponto de partida para aqueles que desejam começar a utilizar o software R e suas ferramentas estatísticas para análise de dados e pesquisas acadêmicas.

# R prazer em conhecê-lo

"R é uma linguagem computacional que permite que usuário programe algoritmos e utilize ferramentas que foram programadas por outras pessoas."

Zuur et. al. (2009) A Beginner's Guide to R. Use R! Springer.



# R prazer em conhecê-lo

"A chave para entender o R é que trata-se de uma linguagem.  
Uma linguagem para manipular objetos."

Bill Venables

"A pergunta certa sobre uma análise em R não é se é possível fazê-la,  
e sim como fazê-la."

Paulo Justiniano Ribeiro

# R prazer em conhecê-lo

O R foi criado originalmente por Ross Ihaka e por Robert Gentleman na Universidade de Auckland, Nova Zelândia, desenvolvido por um esforço colaborativo por pessoas de todo o mundo.

Trata-se de uma linguagem de programação especializada em computação com dados. Sendo ao mesmo tempo uma linguagem de programação e um ambiente para análises estatísticas e gráficas.

Algumas das suas principais características são o seu caráter gratuito e a sua disponibilidade para uma gama bastante variada de sistemas operacionais.

Com o R podemos...

- fazer cálculos;
- realizar análises estatísticas;
- gerar gráficos com qualidade de publicação;
- construir funções e programas para necessidades específicas.

## Por que aprender R?

- Custo (\$\$): é gratuito!
- Disponibilidade para plataformas Windows, MacOS e UNIX.
- Software de livre distribuição na forma de código fonte aberto.
- Contém um grande número de colaboradores que desenvolvem pacotes específicos para as mais diversas áreas.
- Contém implementações de métodos avançados, que não são facilmente encontrados em outros programas estatísticos.
- Permite que você crie e modifique suas próprias funções e pacotes.
- Capacidade de produção de gráficos de alta qualidade.



# Instalação do R

## 1 Primeiro Passo

- O programa de instalação do R pode ser baixado a partir do site:  
<https://www.r-project.org> (The R Project for Statistical Computing).

# Instalação do R

## 1 Primeiro Passo

- O programa de instalação do R pode ser baixado a partir do site: <https://www.r-project.org> (The R Project for Statistical Computing).
- Devemos escolher o *CRAN Mirrors* (por exemplo, Universidade Federal do Paraná).

# Instalação do R

## 1 Primeiro Passo

- O programa de instalação do R pode ser baixado a partir do site: <https://www.r-project.org> (The R Project for Statistical Computing).
- Devemos escolher o *CRAN Mirrors* (por exemplo, Universidade Federal do Paraná).
- Sistema Operacional (por exemplo, *Windows*)

# Instalação do R

## 1 Primeiro Passo

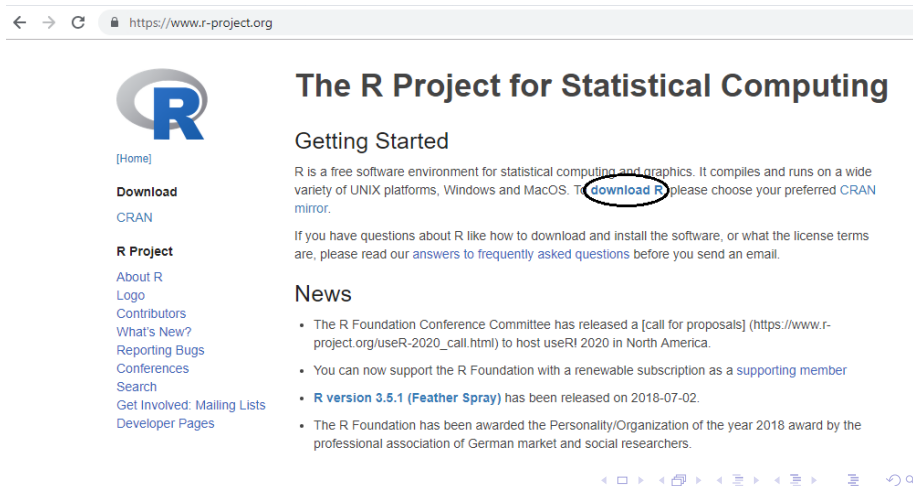
- O programa de instalação do R pode ser baixado a partir do site: <https://www.r-project.org> (The R Project for Statistical Computing).
- Devemos escolher o *CRAN Mirrors* (por exemplo, Universidade Federal do Paraná).
- Sistema Operacional (por exemplo, *Windows*)
- Install R for the first time (última versão: R.3.5.1 for *Windows*).

# Instalação do R

## 1 Primeiro Passo


- O programa de instalação do R pode ser baixado a partir do site: <https://www.r-project.org> (The R Project for Statistical Computing).
- Devemos escolher o *CRAN Mirrors* (por exemplo, Universidade Federal do Paraná).
- Sistema Operacional (por exemplo, *Windows*)
- Install R for the first time (última versão: R.3.5.1 for *Windows*).

O programa de instalação do R pode ser baixado a partir do site:  
<https://www.r-project.org> (The R Project for Statistical Computing).



The screenshot shows the homepage of the R Project for Statistical Computing. At the top, there is a navigation bar with a back arrow, a forward arrow, a refresh icon, and the URL <https://www.r-project.org>. Below the navigation bar, the R logo is displayed on the left, and the main heading "The R Project for Statistical Computing" is centered. Under the heading, the "Getting Started" section is visible, followed by a paragraph about R being a free software environment. A link labeled "download R" is circled in red. To the left of the main content, there is a sidebar with links for "Download", "CRAN", "R Project", "About R", "Logo", "Contributors", "What's New?", "Reporting Bugs", "Conferences", "Search", "Get Involved: Mailing Lists", and "Developer Pages". At the bottom of the page, there is a "News" section with a list of recent events and releases. The footer of the page contains navigation icons and the page number "10 / 92".

← → ↻ <https://www.r-project.org>



[Home]

**Download**

CRAN

**R Project**

About R  
Logo  
Contributors  
What's New?  
Reporting Bugs  
Conferences  
Search  
Get Involved: Mailing Lists  
Developer Pages

## The R Project for Statistical Computing

### Getting Started

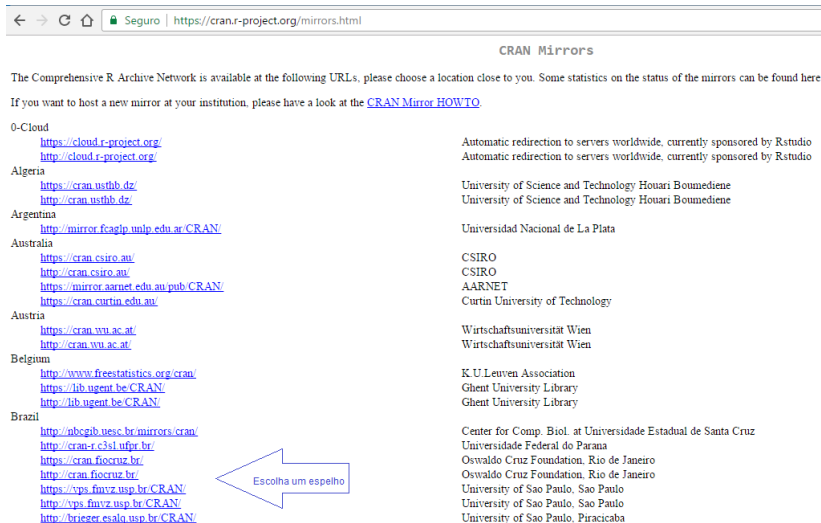
R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

### News

- The R Foundation Conference Committee has released a [call for proposals] ([https://www.r-project.org/useR-2020\\_call.html](https://www.r-project.org/useR-2020_call.html)) to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a [supporting member](#)
- **R version 3.5.1 (Feather Spray)** has been released on 2018-07-02.
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

Devemos escolher o *CRAN Mirrors* (por exemplo,  
<http://cran-r.c3sl.ufpr.br/> (Universidade Federal do Paraná)



The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here:  
 If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#)

0-Cloud  
<https://cloud.r-project.org/>  
<http://cloud.r-project.org/>

Algeria  
<https://cran.usthb.dz/>  
<http://cran.usthb.dz/>

Argentina  
<http://mirror.fcaglp.unlp.edu.ar/CRAN/>

Australia  
<https://cran.csiro.au/>  
<http://cran.csiro.au/>  
<https://mirror.aarnet.edu.au/pub/CRAN/>  
<https://cran.curtin.edu.au/>

Austria  
<https://cran.wu.ac.at/>  
<http://cran.wu.ac.at/>

Belgium  
<http://www.freeststatistics.org/cran/>  
<https://lib.ugent.be/CRAN/>  
<http://lib.ugent.be/CRAN/>

Brazil  
<http://nbcgib.uesc.br/mirrors/cran/>  
<http://cran-r.c3sl.ufpr.br/>  
<https://cran.fiocruz.br/>  
<http://cran.fiocruz.br/>  
<https://vps.fmvz.usp.br/CRAN/>  
<http://vps.fmvz.usp.br/CRAN/>  
<http://brieger.esalq.usp.br/CRAN/>

Automatic redirection to servers worldwide, currently sponsored by Rstudio  
 Automatic redirection to servers worldwide, currently sponsored by Rstudio

University of Science and Technology Houari Boumediene  
 University of Science and Technology Houari Boumediene

Universidad Nacional de La Plata

CSIRO  
 CSIRO  
 AARNET  
 Curtin University of Technology

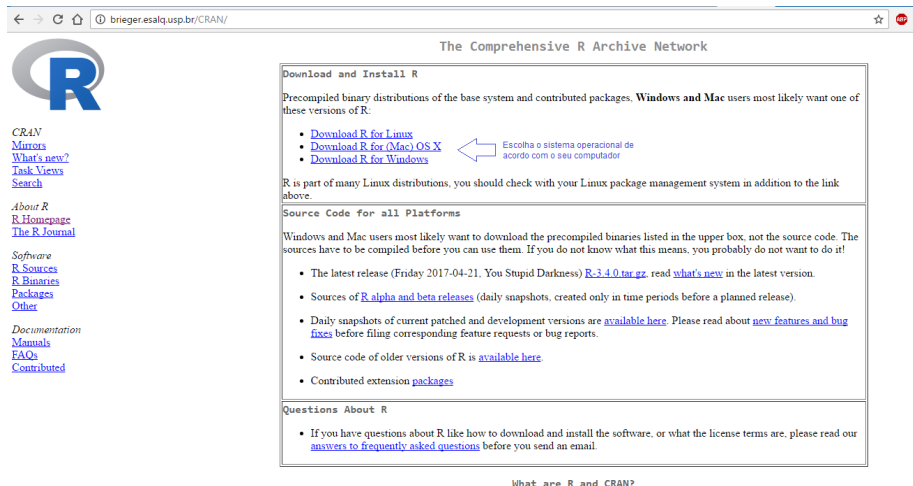
Wirtschaftsuniversität Wien  
 Wirtschaftsuniversität Wien

K.U.Leuven Association  
 Ghent University Library  
 Ghent University Library


Center for Comp. Biol. at Universidade Estadual de Santa Cruz  
 Universidade Federal do Parana  
 Oswaldo Cruz Foundation, Rio de Janeiro  
 Oswaldo Cruz Foundation, Rio de Janeiro  
 University of Sao Paulo, Sao Paulo  
 University of Sao Paulo, Sao Paulo  
 University of Sao Paulo, Piracicaba

Escolha um espelho

## Escolher o Sistema Operacional (por exemplo, *Windows*)



← → ↺ Ⓜ ⓘ briegeer.esalq.usp.br/CRAN/ ☆



CRAN  
[Mirrors](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)

About R  
[R Homepage](#)  
[The R Journal](#)

Software  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Other](#)

Documentation  
[Manuals](#)  
[FAQs](#)  
[Contributed](#)

### The Comprehensive R Archive Network

#### Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

Escolha o sistema operacional de acordo com o seu computador

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

#### Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (Friday 2017-04-21, You Stupid Darkness) [R-3.4.0.tar.gz](#). read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

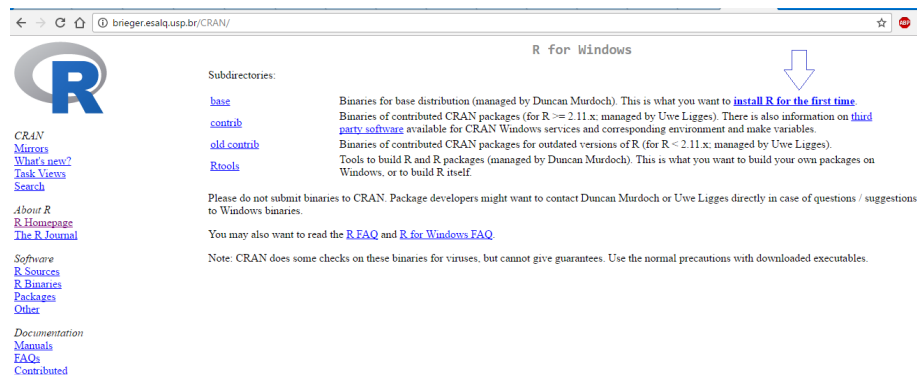
#### Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?



# Clicar em “Install R for the first time”



The screenshot shows a web browser window with the address bar displaying "brieger.esalq.usp.br/CRAN/". The page title is "R for Windows". On the left side, there is a large blue "R" logo and a sidebar menu with links: "CRAN", "Mirrors", "What's new?", "Task Views", "Search", "About R", "R Homepage", "The R Journal", "Software", "R Sources", "R Binaries", "Packages", "Other", "Documentation", "Manuals", "FAQs", and "Contributed". The main content area is titled "Subdirectories:" and lists four links: "base", "contrib", "old.contrib", and "Rtools". To the right of these links, there are four paragraphs of text. The first paragraph, under "base", says "Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#)". The second paragraph, under "contrib", says "Binaries of contributed CRAN packages (for R >= 2.11.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables." The third paragraph, under "old.contrib", says "Binaries of contributed CRAN packages for outdated versions of R (for R < 2.11.x; managed by Uwe Ligges)." The fourth paragraph, under "Rtools", says "Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself." Below these paragraphs, there is a note: "Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions: to Windows binaries." and another note: "You may also want to read the [R FAQ](#) and [R for Windows FAQ](#)." At the bottom of the page, there is a footer with navigation icons and the text "Semana da Estatística (UFLA)", "Conhecendo o R", "2018", and "13 / 92".

CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

Documentation

[Manuals](#)

[FAQs](#)

[Contributed](#)

R for Windows

Subdirectories:

[base](#)

[contrib](#)

[old.contrib](#)

[Rtools](#)

Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).

Binaries of contributed CRAN packages (for R >= 2.11.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

Binaries of contributed CRAN packages for outdated versions of R (for R < 2.11.x; managed by Uwe Ligges).

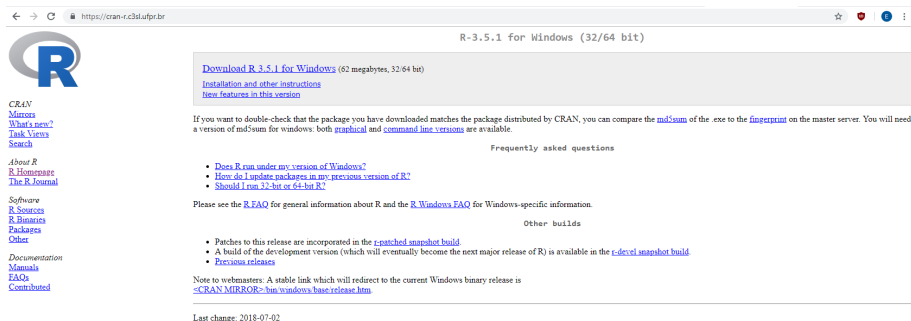
Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions: to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Faça o download da versão mais atual (última versão: R.3.5.1 for *Windows*):



The screenshot shows the CRAN website for R 3.5.1 for Windows (32/64 bit). The page includes navigation links on the left, a main content area with download instructions, and a footer with the last change date.

CRAN

[Mirrors](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)

[About R](#)  
[R Homepage](#)  
[The R Journal](#)

[Software](#)  
[R Sources](#)  
[R Binaries](#)  
[Packages](#)  
[Other](#)

[Documentation](#)  
[Manuals](#)  
[FAQs](#)  
[Contributed](#)

R-3.5.1 for Windows (32/64 bit)

[Download R 3.5.1 for Windows](#) (62 megabytes, 32/64 bit)  
[Installation and other instructions](#)  
[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched\\_snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel\\_snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is  
[<CRAN.MIRROR>bin/windows/base/release.htm](#)

Last change: 2018-07-02

# Instalação do R

- 1 Segundo Passo: basta seguir a rotina de instalação:
  - Abra o programa de instalação do R que acabou de baixar (selecione o idioma).


# Instalação do R


- ① Segundo Passo: basta seguir a rotina de instalação:
  - Abra o programa de instalação do R que acabou de baixar (selecione o idioma).
  - Clique em “Avançar”  
item No final clique em “Concluir”.

Instale também uma interface gráfica (GUI).  
Recomendamos o RStudio:  
<http://www.rstudio.com/ide/download/desktop>



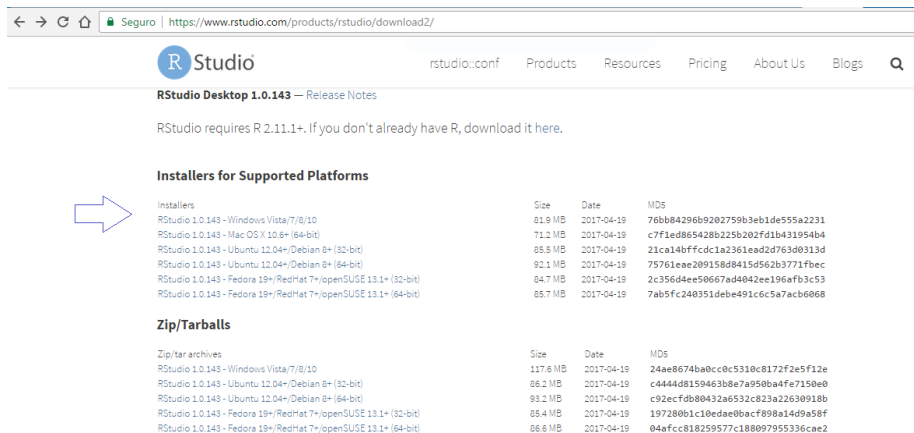
# Instalando RStudio

← → ↻ 🏠 Seguro | <https://www.rstudio.com/products/rstudio/download2/> ☆ 

 [rstudio::conf](#) [Products](#) [Resources](#) [Pricing](#) [About Us](#) [Blogs](#) 🔍

	RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Server Pro Commercial License	RStudio Server Pro + RStudio Connect Commercial License
	FREE	\$995 per year	FREE	\$9,995 per year	\$29,995 per year
Integrated Tools for R	●	●	●	●	●
Priority Support		●		●	●
Access via Web Browser			●	●	●
Enterprise Security				●	●
Project Sharing				●	●
Manage Multiple R Sessions & Versions				●	●
Admin Dashboard				●	●
Load Balancing				●	●
Push-Button Publishing					●
Self-Managed Content					●

# Instalando RStudio



← → ↺ 🏠 Seguro | <https://www.rstudio.com/products/rstudio/download2/>

**R Studio** [rstudio::conf](#) [Products](#) [Resources](#) [Pricing](#) [About Us](#) [Blogs](#) 🔍

**RStudio Desktop 1.0.143 — Release Notes**

RStudio requires R 2.11.1+. If you don't already have R, download it [here](#).

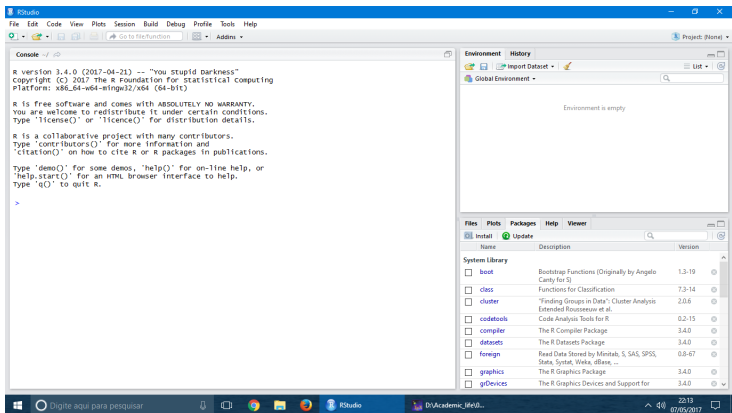
**Installers for Supported Platforms**

Installers	Size	Date	MD5
RStudio 1.0.143 - Windows Vista/7/8/10	81.9 MB	2017-04-19	76bb84296b9202759b3eb1de555a2231
RStudio 1.0.143 - Mac OS X 10.6+ (64-bit)	71.2 MB	2017-04-19	c7f1ed865428b225b202fd1b431954b4
RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ (32-bit)	85.5 MB	2017-04-19	21ca14bffcddc1a2361ead2d763d0313d
RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ (64-bit)	92.1 MB	2017-04-19	75761eae209158d8415d562b3771fbec
RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2017-04-19	2c356d4ee50667ad4042ee196afb3c53
RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2017-04-19	7ab5fc240351debe491c6c5a7acb6068

**Zip/Tarballs**

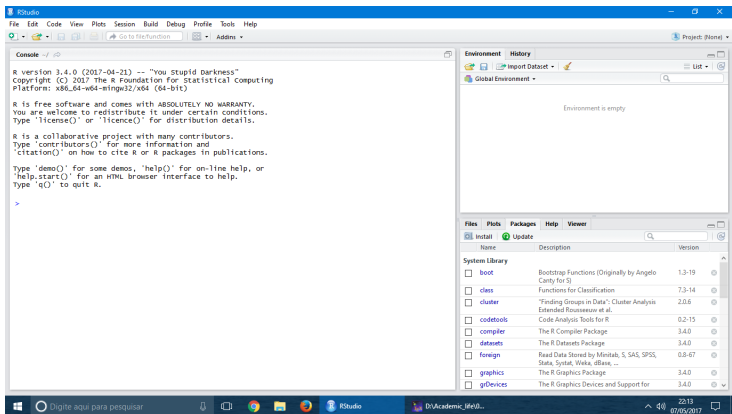
Zip/tar archives	Size	Date	MD5
RStudio 1.0.143 - Windows Vista/7/8/10	117.6 MB	2017-04-19	24ae8674ba0cc0c5310c8172f2e5f12e
RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ (32-bit)	86.2 MB	2017-04-19	c4444d8159463b8e7a950ba4fe7150e0
RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ (64-bit)	93.2 MB	2017-04-19	c92ecfdb80432a6532c823a22630918b
RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	85.4 MB	2017-04-19	197280b1c10edae0bacf898a14d9a58f
RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	86.6 MB	2017-04-19	04afc8c818259577c188097955336cae2

Ao clicar no ícone do Rstudio, o R abrirá automaticamente o “Console” que é a janela na qual os comandos são digitados. Internamente ao Console, se encontra o *prompt* “>”, que é um sinal indicador de que o R está pronto para receber comando.



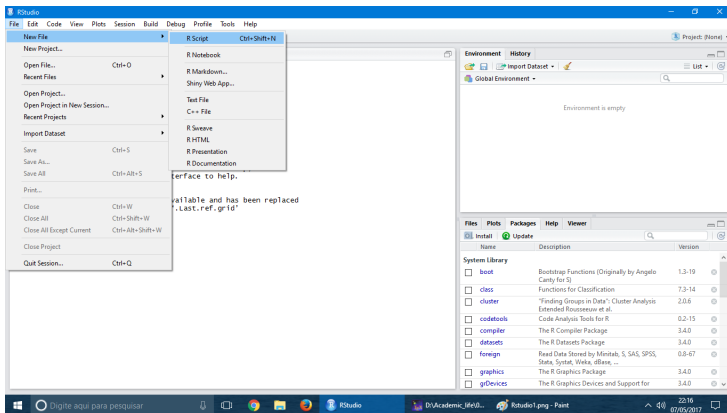


O processo de utilização do R requer que o usuário digite comandos no console, na frente do *prompt* “>” e aperte "enter" para que o comando seja executado.



# Novo script no RStudio

Recomendamos que seu código seja digitado em um *script*:



# Instalando e usando pacotes (packages) do R

O programa R é composto de 3 partes básicas:

- o R-base.
- os pacotes recomendados (recommended packages).
- os pacotes contribuídos (contributed packages).

# R-base

É o "coração" do R e contém as funções principais disponíveis quando iniciamos o programa.

# Recommended packages

São pacotes que são instalados junto com o R-base mas não são carregados quando iniciamos o programa.

Por exemplo os pacotes MASS, lattice, nlme são pacotes recomendados. E há vários outros.

Para usar as funções destes pacotes deve-se carregá-los antes com o comando `library()`.

Por exemplo, para carregar o pacote MASS utilize o comando:

```
>library(MASS)
```

# Contributed packages

Não são instalados junto com o R-base.

Estes pacotes estão disponíveis na página do R e são pacotes adicionais que fornecem funcionalidades específicas.

Para serem utilizados, deve-se baixar, instalar e carregar o pacote de interesse, conforme explicado a seguir.

Para ver a lista deste pacotes com uma descrição de cada um deles acesse a página do R e siga os links para CRAN e “contributed extension packages”.

# Instalando packages

Basta usar o comando `install.packages()` com o nome do pacote desejado entre aspas.

Por exemplo para instalar o pacote `lme4` digite:

```
> install.packages('lme4')
```

O pacote vai ser instalado no sistema e ficar disponível para todos os usuários.

Para usar o pacote basta digitar:

```
> library(lme4)
```

# Alguns símbolos e comandos importantes em R

=====	
COMANDO	DESCRIÇÃO
=====	
#	R ignore o que será digitado após este comando
;	separa dois comandos numa mesma linha
NA	representa um dado ausente
is.na()	verificar se há valores ausentes
=====	



# Operações aritméticas no R:

R utiliza os seguintes símbolos para realizar operações aritméticas:

- $+$  : adição
- $-$  : subtração
- $*$  : multiplicação
- $/$  : divisão
- $()$  : ordenamento de operações
- $^$  : potência
- `sqrt()`: raiz quadrada

Em cada caso, uma expressão matemática é inserida, avaliada e o resultado da operação é impresso na tela.

# Exemplos:

```
> 5/2  
[1] 2.5
```

```
> 5^2  
[1] 25
```

```
> ((2*8)/4)  
[1] 4
```

```
> sqrt(16)  
[1] 4
```

Em cada caso, uma expressão matemática é inserida, avaliada e o resultado da operação é impresso na tela.

Note que cada resultado é precedido por "[1]". O prefixo [1] indica que o resultado é o primeiro elemento do vetor de saída.

# Operações lógicas em R

=====	
OPERADOR	DESCRIÇÃO
=====	
==	igualdade
!=	diferente
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
&	e (and)
	ou (or)
!	não
=====	

# Comandos Básicos

Outras operações matemáticas:

Função	Descrição
<code>abs(x)</code>	Valor absoluto de $x$
<code>ceiling(x)</code>	Arredondamento de $x$ para o maior valor
<code>exp(x)</code>	Exponencial elevado a $x$
<code>floor(x)</code>	Arredondamento de $x$ para o menor valor
<code>length(x)</code>	Número de elementos do vetor $x$
<code>log(x)</code>	Logaritmo natural de $x$
<code>log(x,a)</code>	Logaritmo de $x$ com base $a$
<code>max(x)</code>	Seleciona o maior valor de $x$
<code>min(x)</code>	Menor valor de $x$
<code>sqrt(x)</code>	Raiz quadrada de $x$

# Sinais de Atribuição e de Igualdade

No R, um dos sinais de atribuição é `<-` ou alternativamente pode-se usar o sinal de igual (`=`):

```
>a <- sqrt(25) #sinal de atribuição para nomear a variável "a"  
>a  
[1] 5
```

```
>a = sqrt(25)  #sinal de igual para nomear a variável "a"  
>a  
[1] 5
```

Recomendamos o uso do sinal `<-` para atribuir valor a um objeto.

Um detalhe importante é diferenciar um sinal de igualdade (`==`) de um sinal de atribuição (`=`).

O sinal de igualdade (`==`) faz uma comparação entre dois elementos.

Este operador retorna o resultado do teste lógico "`a igual b?`", que só pode ter dois valores, `TRUE` (verdadeiro), ou `FALSE` (falso).

O sinal de igual (`=`) atribui um valor a um objeto.

Exemplo:

```
> a = 2 + 2
```

```
> a == 4
```

```
[1] TRUE
```

```
> a == 2
```

```
[1] FALSE
```

# Interface do R

R é uma linguagem de expressões com regras e sintaxe muito simples. Faz distinção entre maiúsculas e minúsculas, de modo que os caracteres “x” e “X” são interpretados como sendo diferentes.

A função `round(x, digits = n)` arredonda os elementos de `x` para `n` casas decimais.

```
> round(exp(1), digits=2)
[1] 2.72
```

```
> round(exp(1), digits=0)
[1] 3
```



# Alguns objetos do R

- Variable
- Vetor
- Matrix
- List
- Data Frame

# Variáveis (*variables*)

Nomes de variáveis podem incluir letras, números e caracteres, desde que iniciem com uma letra ou ".".

Para visualizar o valor de uma variável, basta digitar o seu nome.

Procure nomear suas variáveis de maneira informativa, afim de melhorar a inteligibilidade de seu código.

# Tipos de variáveis

Basicamente tem-se quatro tipos variáveis no R: numéricas, caracteres, lógicas e números complexos.

```
> x <- 30
```

```
> x
```

```
[1] 30
```

# Tipos de variáveis

Basicamente tem-se quatro tipos variáveis no R: numéricas, caracteres, lógicas e números complexos.

```
> x <- 30
```

```
> x
```

```
[1] 30
```

```
> frase <- "aprendendo R"
```

```
[1] aprendendo R
```

# Tipos de variáveis

Basicamente tem-se quatro tipos variáveis no R: numéricas, caracteres, lógicas e números complexos.

```
> x <- 30
```

```
> x
```

```
[1] 30
```

```
> frase <- "aprendendo R"
```

```
[1] aprendendo R
```

```
> 3 > 7
```

```
[1] FALSE
```

# Tipos de variáveis

Basicamente tem-se quatro tipos variáveis no R: numéricas, caracteres, lógicas e números complexos.

```
> x <- 30
```

```
> x
```

```
[1] 30
```

```
> frase <- "aprendendo R"
```

```
[1] aprendendo R
```

```
> 3 > 7
```

```
[1] FALSE
```

```
> 9 + 4i
```

```
[1] 9 + 4i
```

Determinadas palavras (além do nome das funções) estão reservadas no R, pois são utilizadas com significado especial:

- `pi` - constante (`pi = 3.141593`);
- `Inf` - infinito;
- `NaN` - indeterminado (*Not a Number*), normalmente resultado de uma operação matemática indeterminada;
- `NA` - indeterminado (*Not Available*), normalmente caracterizando uma observação perdida (*missing value*).

Na operações matemáticas, `NaN` e `NA` atuam sempre como indeterminado.

# Vetores (*Vector*)

O R é um ambiente vetorial, isto é, quase todas suas operações atuam sobre um conjunto de valores, que genericamente chamamos de vetores. Vetores são conjuntos de dados unidimensionais. No R, um vetor é uma coleção ordenada de elementos do mesmo tipo (*mode*).

Para criar um vetor, podemos usar a função `c` (`c` = concatenar). Essa função simplesmente junta todos os argumentos dados a ela, formando um vetor. Exemplo:

```
> v1 <- c(10.5, 11.3, 12.4, 5.7)
> v1
[1] 10.5 11.3 12.4 5.7
```



## Exemplos:

```
> v0 <- c()
```

```
> v0
```

```
NULL
```

```
> v2 <- c(5.4, -3.7, 11.2, 0.78, 21.6)
```

```
> v2
```

```
[1] 5.40 -3.70 11.20 0.78 21.60
```

```
> v3 <- c(FALSE, TRUE, TRUE, TRUE, FALSE)
```

```
> v3
```

```
[1] FALSE TRUE TRUE TRUE FALSE
```

```
> v4 <- c("Ana","Paulo","Zé", "Maria", "João")
```

```
> v4
```

```
[1] "Ana" "Paulo" "Zé" "Maria" "João"
```

# Vetores: Criação de Sequências

Para criar vetores de números com intervalo fixo unitário (intervalo de 1) se utiliza o operador sequencial (:):

```
> v5 = 1:8
```

```
> v5
```

```
[1] 1 2 3 4 5 6 7 8
```

```
> v6 = 20:32
```

```
> v6
```

```
[1] 20 21 22 23 24 25 26 27 28 29 30 31 32
```

```
> v7 = 2.5:10
```

```
> v7
```

```
[1] 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5
```

# Vetores: Criação de Sequências

Uma forma mais flexível de criar sequências de números (inteiros ou reais) é usando a função 'seq':

```
> seq(from=10, to=30)
```

```
[1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24  
[15] 25 26 27 28 29 30
```

```
> seq(from=10, to=30, by=2)
```

```
[1] 10 12 14 16 18 20 22 24 26 28 30
```

# Vetores de Valores Repetidos

Também é fácil criar uma sequência de números repetidos utilizando a função `rep`:

```
> rep(5, 3)
```

```
[1] 5 5 5
```

```
> rep(1:5, 3)
```

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

```
> rep(1:5, each=3)
```

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

# Operações matemáticas com vetores

```
> v8 <- seq(10,80, by = 10)
```

```
> v8
```

```
[1] 10 20 30 40 50 60 70 80
```

```
> 2*v8
```

```
[1] 20 40 60 80 100 120 140 160
```

Se as variáveis que trabalhamos são vetores, podemos realizar operações matemáticas entre variáveis pareando os elementos dos vetores:

```
> v8
```

```
[1] 10 20 30 40 50 60 70 80
```

```
> v5
```

```
[1] 1 2 3 4 5 6 7 8
```

```
> v8*v5
```

```
[1] 10 40 90 160 250 360 490 640
```

```
> v8-v5
```

```
[1] 9 18 27 36 45 54 63 72
```

# Matrizes

Uma matriz é um conjunto retangular de números símbolos ou expressões, organizados em linhas e colunas.

Cada um dos itens de uma matriz é chamado de elemento.

No R, uma matriz é uma coleção de dados, todos do mesmo tipo, referenciados por dois índices.

Assim a matriz pode ser vista como uma generalização para duas dimensões de um vetor, definida pelo número de linhas ( $n$ ), número de colunas ( $m$ ) e um conjunto de ( $n \times m$ ) valores.

# Matrizes

A função *matrix* cria uma matriz com os valores do argumento *data*.  
O números de linhas e colunas são definidos pelos argumentos *nrow* e *ncol*:

```
> m1 <- matrix(data=1:12, nrow=3, ncol=4)  
> m1
```

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



O argumento preenche a matriz por linhas:

```
> m2 <- matrix(data=1:12, nrow=3, ncol=4, byrow = TRUE)
```

```
> m2
```

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

Para alterar um valor em qualquer posição da matriz, basta utilizar os comandos:

```
> m2[1,2] <- 5
```

```
> m2
```

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

```
> m3 <- matrix(data=1:16, 4, 4)
```

```
> m3
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	5	9	13
[2,]	2	6	10	14
[3,]	3	7	11	15
[4,]	4	8	12	16

```
> t(m3)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	2	3	4
[2,]	5	6	7	8
[3,]	9	10	11	12
[4,]	13	14	15	16

```
> det(m3)
```

```
[1] 0
```

# Matrizes

Algumas operações básicas com matrizes:

Função	Descrição
$A \%*\% B$	Produto matricial de A por B
$t(A)$	Transposta da matriz A
$solve(A)$	Inversa da matriz A
$x = solve(A, b)$	Resolve o sistema linear $Ax = b$
$det(B)$	Determinante de B
$eigen(A)$	Retorna os autovalores e autovetores de A

**Lista:** serve para armazenar diferentes objetos dentro de um mesmo objeto.

```
> l1 <- list(v2, v3, v4, m2)
```

```
> l1
```

```
[[1]]
```

```
[1] 5.40 -3.70 11.20 0.78 21.60
```

```
[[2]]
```

```
[1] FALSE TRUE TRUE TRUE FALSE
```

```
[[3]]
```

```
[1] "Ana" "Paulo" "Zé" "Maria" "João"
```

```
[[4]]
```

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

**Data frame:** é um matriz de dados em que as colunas podem ser de diferentes tipos.

```
> df <- data.frame(v2, v3, v4)
> df
```

	v2	v3	v4
1	5.40	FALSE	Ana
2	-3.70	TRUE	Paulo
3	11.20	TRUE	Zé
4	0.78	TRUE	Maria
5	21.60	FALSE	João

# Operações com vetores: Estatística descritiva

As funções matemáticas sobre vetores operam elemento-a-elemento.  
Já as funções estatísticas operam no vetor como um todo.

Exemplo: Considere o conjunto de dados:

```
a <- c(32, 12, 45, 9, 78, 16, 54, 14)
a
[1] 32  12  45  9  78  16  54  14
```

Para ordenar os dados

```
a <- sort(a)
a
[1] 9  12  4  16  32  45  54  78
```

# Exemplo: Estatística descritiva

## Medidas de posição

Média:

```
n <- length(a)
n
media <- sum(a)/n
media
[1] 32.5
```

Ou você pode utilizar a função pronta do R para cálculo da média:

```
mean(a)
[1] 32.5
```



# Exemplo: Estatística descritiva

## Medidas de posição

Mediana:

```
n <- length(a)
n
mediana <- (a[n/2] + a[(n+2)/2])/2
mediana
[1] 24
```

Ou você pode utilizar a função pronta do R para cálculo da mediana:

```
median(a)
[1] 24
```

# Exemplo: Estatística descritiva

## Medidas de posição

Moda:

Para calcular a moda devemos instalar o pacote `modeest` que não vem instalado junto com a versão básica do R.

```
install.packages('modeest')  
library(modeest)
```

```
mfv(a)  
[1] 9 12 14 16 32 45 54 78
```

Como os valores são todos diferentes, o conjunto é amodal.

# Exemplo: Estatística descritiva

## Medidas de posição

Moda: Exemplo usando a função `mfv` do pacote `modeest`

```
x<- c(1, 2, 3, 4, 5, 5, 5, 5, 5, 5, 5, 6, 6, 7, 7, 8, 9)
mfv(x)
[1] 5
```

# Exemplo: Estatística descritiva

## Medidas de dispersão

### Variância amostral

```
a <- c(32, 12, 45, 9, 78, 16, 54, 14)
var(a)
[1] 610.8571
```

### Desvio padrão amostral

```
sqrt(var(a))
[1] 24.71552
```

```
sd(a)
[1] 24.71552
```

# Exemplo: Estatística descritiva

## Medidas de dispersão

### Amplitude

```
a <- c(32, 12, 45, 9, 78, 16, 54, 14)
```

```
Amp <- max(a) - min(a)
```

```
Amp
```

```
[1] 69
```

### Coeficiente de variação (CV)

```
CV <- sd(a)/mean(a)*100
```

```
CV
```

```
[1] 76.04777
```

## Exemplo: Estatística descritiva

Erro padrão da média

```
a <- c(32, 12, 45, 9, 78, 16, 54, 14)
n <- length(a)
sd(a)/sqrt(n)
8.738257
```

Resumo dos dados (mínimo e máximo, quartis, mediana e a média)

```
summary(a)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
9.00	13.50	24.00	32.50	47.25	78.00

# Tabelas e gráficos no R

O R é uma excelente ferramenta para construção de gráficos. Podemos criar histogramas, boxplots (diagrama de caixas), barplots (gráficos de barras), pie plots (gráficos de setores), ogivas, curvas de distribuição, e muito mais. Veja alguns exemplos:

```
demo(graphics)
```

R Graph Gallery (<http://www.r-graph-gallery.com/>) é uma página com diversos exemplos de gráficos no R e os comandos para produzi-los.

## Variáveis qualitativas nominal ou ordinal

Exemplo: Considere a variável sexo, com 7 indivíduos do sexo masculino (M) e 11 do sexo feminino (F).

```
> sexo <- c(rep("M", 7), rep("F", 11))
```

```
> sexo
```

```
[1] "M" "M" "M" "M" "M" "M" "M" "F" "F" "F" "F" "F" "F" "F" "F"
```

```
> Fi <- table(sexo)
```

```
> Fi
```

```
  sexo
```

```
 F    M
```

```
11    7
```



## Variáveis qualitativas nominal ou ordinal

Tabelas de proporção:

```
> prop.table(Fi)
```

sexo

F	M
---	---

0.6111111	0.3888889
-----------	-----------

```
> prop.table(Fi)*100
```

sexo

F	M
---	---

61.11111	38.88889
----------	----------

# Criando tabelas e gráficos no R

Variáveis qualitativas nominal ou ordinal  
Barplot(Gráfico de Barras)

```
barplot(Fi, col = c("skyblue", "violet"),  
        main="Gráfico de colunas para variável sexo",  
        ylab="Número de pessoas", xlab="Sexo")
```

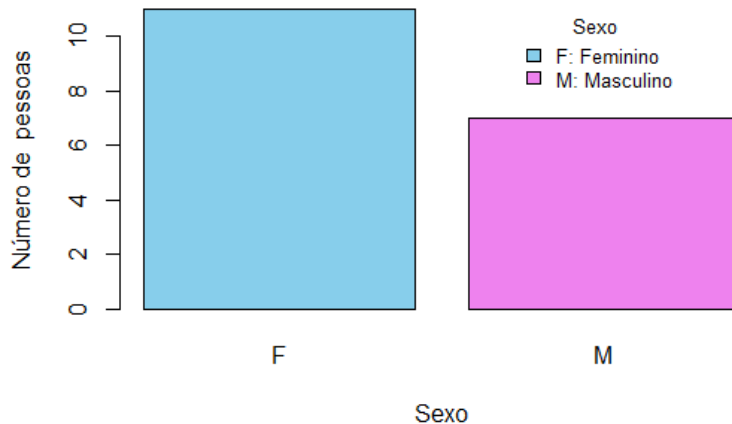
```
#legenda
```

```
legend(1.55,11, c("F: Feminino", "M: Masculino"),  
      cex=0.8, bty="n", fill=c("skyblue", "violet"),  
      title = "Sexo")
```

```
#see all colours
```

```
colours()
```

## Gráfico de colunas para variável sexo



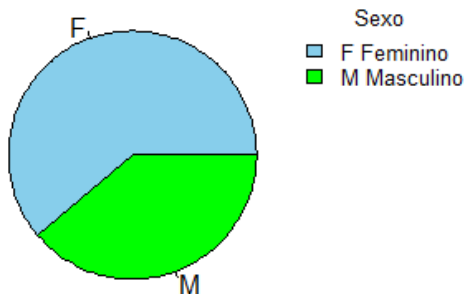
## Variáveis qualitativas nominal ou ordinal

### Pie plot (Gráfico de Setores)

```
pie(Fi, col=c("skyblue", "green"),  
main="Gráfico de setor para variável sexo")
```

```
#legenda  
legend(1, c("F Feminino", "M Masculino"),  
      cex=0.8, bty="n", fill=c("skyblue", "green"),  
      title = "Sexo")
```

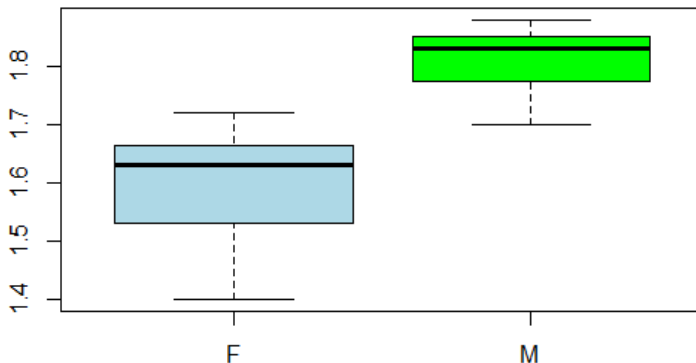
## Gráfico de setor para variável sexo



## Boxplot

```
sexo <- c(rep("M", 7), rep("F", 11))  
altura <- c(1.75, 1.70, 1.80, 1.83, 1.87, 1.83,  
            1.88, 1.40, 1.56, 1.63, 1.68, 1.47,  
            1.65, 1.50, 1.69, 1.58, 1.63, 1.72)  
  
dados <- data.frame(sexo, altura)  
dados  
  
boxplot(dados$altura ~ dados$sexo,  
        col = c("lightblue", "green"),  
        main = "Boxplot para altura dentro de sexo")
```

## Boxplot para altura dentro de sexo



## Gráfico linear

```
x <- seq(0,100, by=5)
```

```
x
```

```
y <- 2*x + 1
```

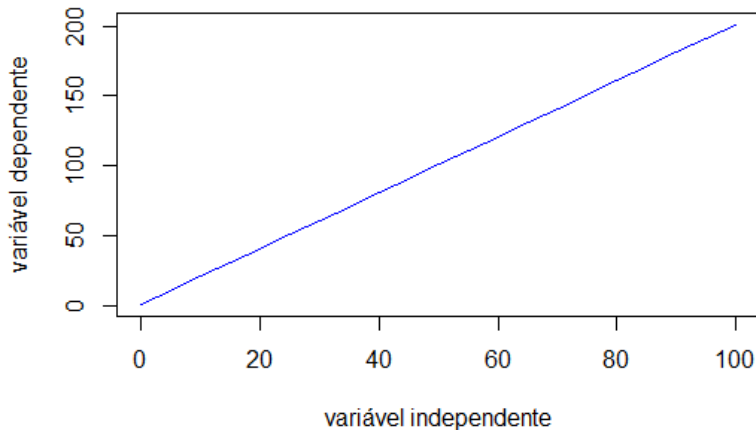
```
plot(x,y)
```

```
plot(x, y, type="l")
```

```
plot(x, y, type="l", main = "Equação de 1o grau", col= "blue",  
xlab = "variável independente", ylab = "variável dependente")
```



## Equação de 1o grau



## Gráfico da equação do 2o grau

```
x <- seq(-100, 100, by = 5)
```

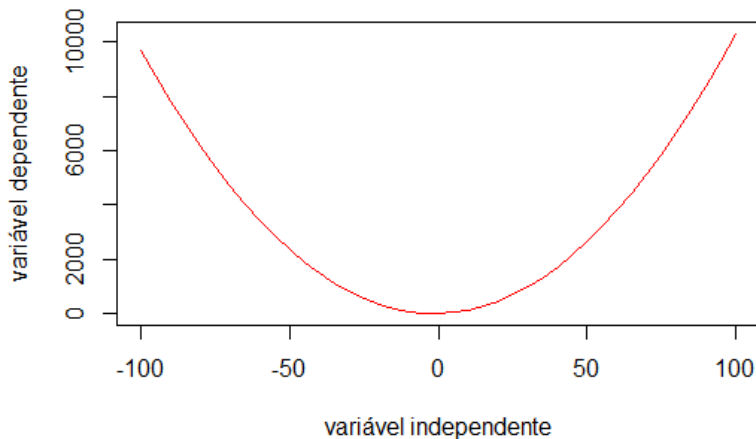
```
y <- x^2 + 3*x - 1
```

```
plot(x,y)
```

```
plot(x,y, type="l")
```

```
plot(x,y, type="l", main = "Equação de 2o grau", col= "red",  
xlab = "variável independente", ylab = "variável dependente")
```

## Equação de 2o grau



# Gráficos

Alguns comandos básicos para gráficos:

Argumentos Básicos	Descrição
pch	Muda o padrão dos pontos
lwd	Muda a largura das linhas
lty	Muda o estilo das linhas
main	Título do gráfico
xlab	Título do eixo x
ylab	Título do eixo y
text	Adiciona texto
points	Adiciona pontos
lines	Adiciona linhas

# Criar conjunto de dados

Um conjunto de dados pode ser digitado diretamente no R, como visto anteriormente, ou lido de arquivos externos utilizando as funções:

- `scan()`
- `edit()`

# Criar conjunto de dados

Um conjunto de dados pode ser digitado diretamente no R, como visto anteriormente, ou lido de arquivos externos utilizando as funções:

- `scan()`
- `edit()`

Podemos ainda digitar os dados em uma planilha eletrônica e importar utilizando as funções:

- `read.table()`
- `read.xlsx()`

# Criar conjunto de dados

## Usando o comando `scan()`

Esta função tem como finalidade a entrada de dados do console ou de um arquivo para armazená-lo(s) em formato de vetor ou lista.

Você pode criar um vetor utilizando uma coluna que já tenha sido digitada em um editor de texto qualquer ou numa planilha eletrônica (excel).

## Exemplo:

```
> x <- scan()
```

```
1:
```

#Após copiar os dados para a área de transferência,  
#basta colar no console do R e teclar "enter".

```
1: 10
```

```
2: 20
```

```
3: 30
```

```
4: 40
```

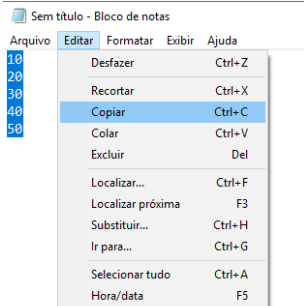
```
5: 50
```

```
6:
```

```
Read 5 items
```

```
> x
```

```
[1] 10 20 30 40 50
```





# Exemplo:

```
> y <- scan()
```

```
1: 20
```

```
2: 40
```

```
3: 60
```

```
4: 80
```

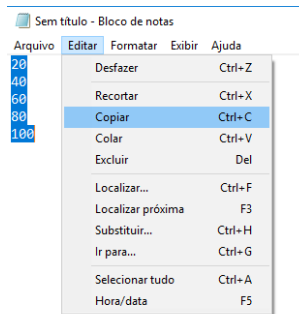
```
5: 100
```

```
6:
```

```
Read 5 items
```

```
y
```

```
[1] 20 40 60 80 100
```



# Criar conjunto de dados

Para criar um conjunto de dados com os dois vetores `x` e `y` criados anteriormente, podemos utilizar a função `cbind()`.

```
> dados1 <- cbind(x,y)
```

```
> dados1
```

	x	y
[1,]	10	20
[2,]	20	40
[3,]	30	60
[4,]	40	80
[5,]	50	100

## Criar conjunto de dados

Para evitar erros futuros, recomenda-se eliminar da memória do R os objetos auxiliares `x` e `y`.

```
> rm(x, y)
```

Para verificar se o objeto `dados1` é um `data.frame` use o comando:

```
> is.data.frame(dados1)  
[1] FALSE
```

O R retorna `FALSE` indicando que este objeto não um `data.frame`.

## Criar conjunto de dados

Para evitar erros futuros, recomenda-se eliminar da memória do R os objetos auxiliares `x` e `y`.

```
> rm(x, y)
```

Para verificar se o objeto `dados1` é um `data.frame` use o comando:

```
> is.data.frame(dados1)
[1] FALSE
```

O R retorna `FALSE` indicando que este objeto não um `data.frame`.

Caso queira transformá-lo em um `data.frame` basta digitar: Para verificar se o objeto `dados1` é um `data.frame` use o comando:

```
> dados1 <- as.data.frame(dados1)
```

# Criar conjunto de dados

## Usando o comando `edit()`

Esta função, como o próprio nome sugere, é útil na edição de objetos já existente, sejam eles vetores, matrizes, `data.frame`, etc. Basta digitar `edit(dados1)` uma janela de edição de dados será aberta.

# Criar conjunto de dados

Usando o comando `edit()`

	x	y	var3	var4	var5	var6	var7
1	10	20					
2	20	40					
3	30	60					
4	40	80					
5	50	100					
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

# Criar conjunto de dados

Usando o comando `read.table()`

Para ler um arquivo-texto (.txt) podemos utilizar o comando `read.table`.

Para isso, você deve informar o endereço e o nome do arquivo (entre aspas). Em seguida, informe se a primeira linha do arquivo corresponde ao cabeçado e qual o caractere de separação dos elementos (por exemplo: se for vírgula use ",", " ")

```
> dados <- read.table(      #Lê os dados de um arquivo texto
  "D:/alunos.txt",         # caminho e nome do arquivo de dados
  header=TRUE,             # primeira linha é cabeçalho
  sep=" ")                 # espaço como separador
> dados
```

Nome	altura	peso	idade
Julio	Cesar	175	71.1 36
Andre	Lima	170	70.8 29
Maria	Paula	168	65.5 32
Marcio	Melo	165	60.2 37
Carla	Andrea	163	58.4 26



# Criar conjunto de dados

Usando o comando `read.xlsx()`

Para ler um arquivo-texto de uma planilha eletrônica podemos utilizar o comando `read.xlsx`.

Para isso, você deve informar carregar o pacote

```
> library("openxlsx")      # carregar o pacote
> dados2 <- read.xlsx(      # Lê os dados de um arquivo texto
  "D:/alunos.xlsx")
> dados2
```

Nome	altura	peso	idade	
Julio	Cesar	175	71.1	36
Andre	Lima	170	70.8	29
Maria	Paula	168	65.5	32
Marcio	Melo	165	60.2	37
Carla	Andrea	163	58.4	26

# Obtendo ajuda

- `help.start()`: ajuda geral

# Obtendo ajuda

- `help.start()`: ajuda geral
- `help("mean")` ou `?mean`: obter ajuda sobre o comando `mean`

# Obtendo ajuda

- `help.start()`: ajuda geral
- `help("mean")` ou `?mean`: obter ajuda sobre o comando `mean`
- `help.search("mean")` ou `?mean`: procura por `mean` em todos os pacotes instalados.

# Obtendo ajuda

- `help.start()`: ajuda geral
- `help("mean")` ou `?mean`: obter ajuda sobre o comando `mean`
- `help.search("mean")` ou `?mean`: procura por `mean` em todos os pacotes instalados.
- `example("mean")`: mostra um exemplo da função `mean`.

# Obtendo ajuda

- `help.start()`: ajuda geral
- `help("mean")` ou `?mean`: obter ajuda sobre o comando `mean`
- `help.search("mean")` ou `?mean`: procura por `mean` em todos os pacotes instalados.
- `example("mean")`: mostra um exemplo da função `mean`.
- `data()`: lista todos os exemplos de conjuntos de dados contidos nos pacotes que estão carregados.

# Obtendo ajuda

- `help.start()`: ajuda geral
- `help("mean")` ou `?mean`: obter ajuda sobre o comando `mean`
- `help.search("mean")` ou `?mean`: procura por `mean` em todos os pacotes instalados.
- `example("mean")`: mostra um exemplo da função `mean`.
- `data()`: lista todos os exemplos de conjuntos de dados contidos nos pacotes que estão carregados.
- `vignette()`: lista todos os tutoriais dos pacotes instalados.



# Obtendo ajuda

- `help.start()`: ajuda geral
- `help("mean")` ou `?mean`: obter ajuda sobre o comando `mean`
- `help.search("mean")` ou `?mean`: procura por `mean` em todos os pacotes instalados.
- `example("mean")`: mostra um exemplo da função `mean`.
- `data()`: lista todos os exemplos de conjuntos de dados contidos nos pacotes que estão carregados.
- `vignette()`: lista todos os tutoriais dos pacotes instalados.
- `vignette("tutorial")`: mostra o tutorial de um pacote específico.

Assim termina nossa primeira sessão com o R.

... E este é apenas o começo!

# Referências

Batista, J.L.F., Prado, P.I. e Oliveira, A. A. (2009). Introdução ao R.

Ferrari, D. B., Introdução ao Software Estatístico R, (2011).

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Peternelli, L. A.; Mello, M. P., Conhecendo o R - Uma visão estatística. Série Didática, Editora UFV, 2013.

Introdução à Linguagem R para Pós-Graduandos em Ecologia

<http://ecologia.ib.usp.br/bie5782/doku.php?id=start>

Introdução ao Ambiente Estatístico R

<http://www.leg.ufpr.br/paulojus/embrapa/Rembrapa/>

Ridículas - Dicas curtas sobre o R

<https://ridiculas.wordpress.com/>

[www.rstudio.com/training](http://www.rstudio.com/training)