



# Introdução à Programação com R

GILSON FRANCISCO DE OLIVEIRA CASTRO

---

[gilson.castro@feac.ufal.br](mailto:gilson.castro@feac.ufal.br)

[gilsoncastroblog.wordpress.com](http://gilsoncastroblog.wordpress.com)

22/01/2019

# O que vamos aprender ?

---

- Apresentação do R
- Operações , objetos
- Funções e pacotes
- Visualização de dados
  - Análise de dados
- Manipulação de dados

# Uso da linguagem R

## ❑ Como as empresas estão usando R

- A Ford usa R para melhorar o design de seus veículos.
- Basicamente, o Twitter usa R para monitorar a experiência do usuário.
- O Serviço Nacional de Meteorologia dos EUA usa R para prever inundações severas.
- O Human Rights Data Analysis Group usa R para quantificar o impacto da guerra.
- R está sendo usado pelo The New York Times para criar infográficos.
- O Google usa R para calcular o ROI de campanhas publicitárias.

---

The New York Times

---



PVCORPALL.COM

# Companies that use R for Analytics



# Por que aprender a programar?

- A programação está se tornando um requisito praticamente essencial na formação de um bom profissional
- Tudo isso torna uma ferramenta realmente útil com a qual podemos gerar meios, retornos ou gráficos que refletem a evolução de um ativo a partir de seus dados históricos com uma linha de código simples.
- Todos os dias, mais e mais empresas estão solicitando aos trabalhadores conhecimentos de programação
- Nas palavras de Tim Cook, CEO da Apple Inc., "a programação é mais importante do que saber inglês". Tanto que a China adotou em 2016 uma reforma educacional que incluiu como disciplina obrigatória a programação para crianças a partir de 6 anos.
- O mundo profissional está se tornando cada vez mais competitivo graças à globalização e toda vez que você pode encontrar profissionais melhor qualificados

## Carreira em ciência de dados promete salários de R\$ 22 mil

Pouco conhecida no Brasil, a profissão é promissora e muito procurada empresas. Saiba mais sobre o trabalho desenvolvido por quem escolhe nessa área

FALA BRASIL

Do R7

ECONOMIA

## Tecnologia ameaça 63% das vagas de trabalho do mundo

Inteligência artificial vai informar, produzir, operar, diagnosticar, dirigir taxi e mais

TECNOLOGIA

## No futuro, trabalhadores assumirão funções menos rotineiras

A tendência é que, com o avanço da tecnologia, os trabalhos mecânicos percam espaço, dando lugar a tarefas mais satisfatórias

Núcleo Digital



COMPARTILHE



POR GIOVANNA SUTTO - EM CARREIRA / EDUCACAO - © 14 SET, 2018 17H09

## Diploma não será mais requisito para contratação com revolução tecnológica

Udacity, Kroton, LinkedIn e Khan Academy fizeram parceria para discutir como a tecnologia vai mudar a educação e o mercado de trabalho

OPINIÃO Edição impressa de 22/10/2018. Alterada em 22/10 às 01h00min

## Analistas e cientistas de dados lideram profissões mais demandadas

30/11/2017 - 19H36 - ATUALIZADA ÀS 09H27 - POR ÉPOCA NEGÓCIOS ONLINE

## Robôs devem roubar 800 milhões de empregos até 2030

Novo estudo da McKinsey mostra como a automação afetará o mercado de trabalho. Mas atenção: as consequências variam muito dependendo do cargo e do país

CARREIRA - VOCÊ S/A

## Você é uma das 120 milhões de pessoas que terão que ser requalificadas?

Em entrevista exclusiva, a vice-presidente de Recursos Humanos da IBM conta como a IA vai mudar todas as indústrias e profissões - começando pelo próprio RH

Por **Luísa Granato**

© 28 nov 2018, 12h39

# Porque a programação computacional pode ser útil para economistas ?

---

- Bases de dados (Raspagem de dados ,Mineração de dados)
- Modelos baseados em econometria não-linear e aprendizagem de máquinas ou deep-learning (Modelos Preditivos de PIB,Risco financeiro , previsões macroeconômicas etc)
- Experimentos sociais (Economia comportamental,Políticas públicas)
- Modelos baseados em programação dinâmica ou controle ótimo(Microeconometria)
- Modelos computacionais (ABM,Sistemas Econômicos Complexos)

# 7 Motivos para você aprender R:

---

- É completamente gratuito e de livre distribuição;
- Curva de aprendizado bastante amigável, sendo muito fácil de se aprender;
  - Enorme quantidade de tutoriais e ajuda disponíveis gratuitamente na internet;
- É excelente para criar rotinas e sistematizar tarefas repetitivas;
- Amplamente utilizado pela comunidade acadêmica e pelo mercado;
- Quantidade enorme de pacotes, para diversos tipos de necessidades;
- Ótima ferramenta para criar relatórios e gráficos.

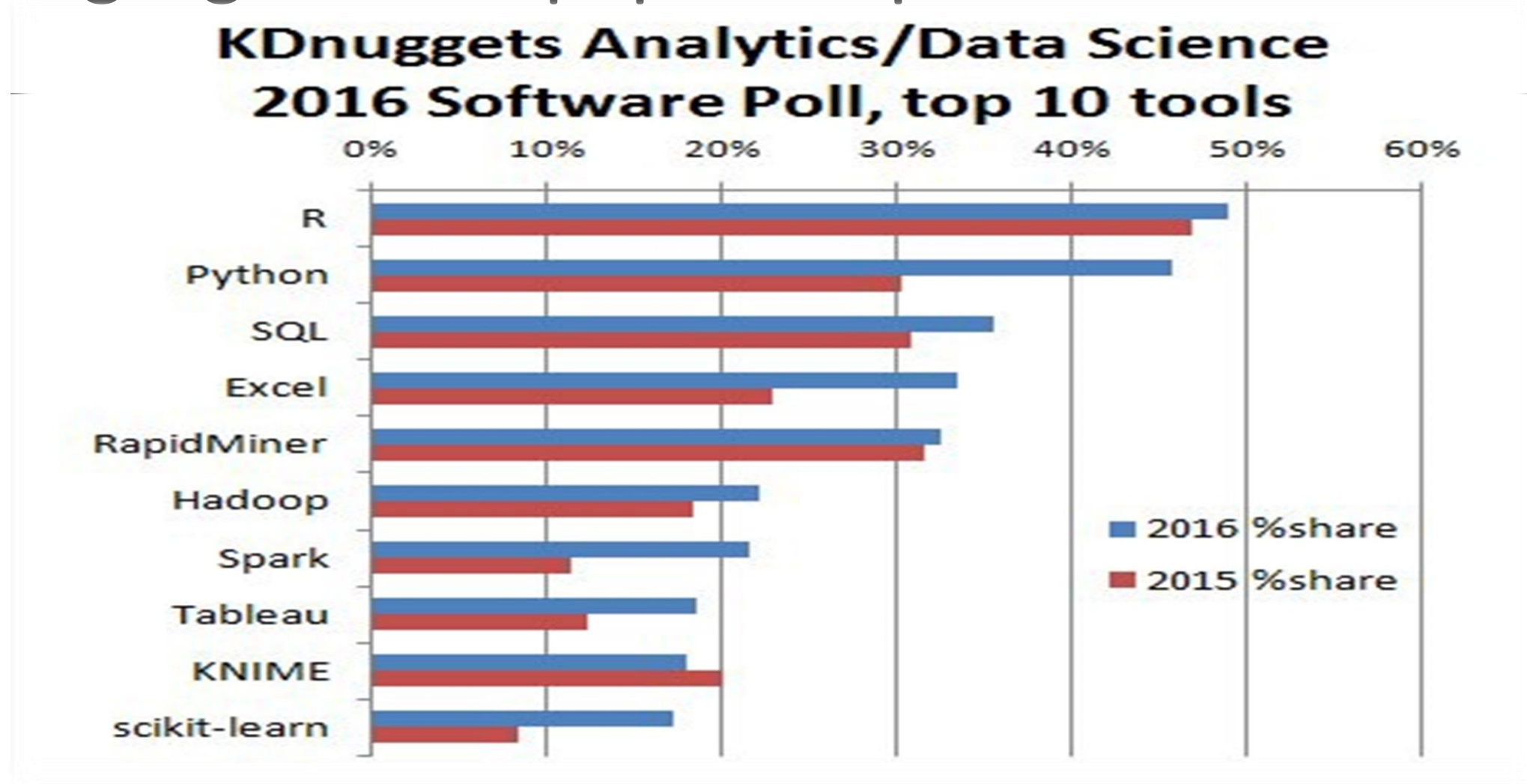


# O que é o R ?


---

- R é uma **linguagem de programação de software livre** e de código aberto e um ambiente de software para computação estatística e gráficos. Distribuído sob a GNU General Public License versão 2, o R é uma linguagem fácil de aprender e comumente usada para o desenvolvimento de software de **análise de dados** e estatística. compila e é executado em uma ampla variedade de plataformas UNIX, Windows e MacOS tendo sua popularidade aumentada nos últimos anos com a ascensão da ciência de dados, big data , IA.
- O R fornece uma ampla variedade de técnicas estatísticas (modelagem linear e não linear, testes estatísticos clássicos, análise de séries temporais, classificação, agrupamento, ...) e técnicas gráficas e é altamente extensível. atua como uma alternativa aos pacotes estatísticos tradicionais, como o SPSS, o SAS e o Stata.

# As linguagens mais populares para a ciência de dados



---

Instalando o 

1º Acesse a página oficial do projeto  
R: <http://www.r-project.org/>

---



[Home]

**Download**

[CRAN](#)

**R Project**

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting Bugs](#)

[Development Site](#)

[Conferences](#)

[Search](#)

**R Foundation**

[Foundation](#)

[Board](#)

[Members](#)

[Donors](#)

[Donate](#)

## 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

- [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.
- [R version 3.5.0 \(Joy in Playing\)](#) has been released on 2018-04-23.

### News via Twitter



The R Foundation Retweeted



**useR! 2019**

[@UseR2019\\_Conf](#)

Nathalie [@Natty\\_V2](#) is presenting [#user2019](#) at [#user2018](#) : Hope to see you all in Toulouse

# 2º Escolha um servidor, de preferência em seu país

<https://cran.ms.unimelb.edu.au/>

<https://cran.curtin.edu.au/>

School of Mathematics and Statistics, University of Melbourne  
Curtin University of Technology

Austria

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

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

Wirtschaftsuniversität Wien  
Wirtschaftsuniversität Wien

Belgium

<http://www.freeststatistics.org/cran/>

<https://lib.ugent.be/CRAN/>

<http://lib.ugent.be/CRAN/>

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

Brazil

<http://nbcgib.uesc.br/mirrors/cran/>

<https://cran-r.c3sl.ufpr.br/>

<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/>

<https://brieger.esalq.usp.br/CRAN/>

<http://brieger.esalq.usp.br/CRAN/>

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

Bulgaria

<https://ftp.uni-sofia.bg/CRAN/>

<http://ftp.uni-sofia.bg/CRAN/>

Sofia University  
Sofia University

Canada

<https://mirror.its.sfu.ca/mirror/CRAN/>

<http://cran.stat.sfu.ca/>

<https://munn.ca/mirror/cran/>

Simon Fraser University, Burnaby  
Simon Fraser University, Burnaby  
Manitoba Unix User Group

# 3º Escolha uma versão do seu Sistema operacional



[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](#)

## 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](#)

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

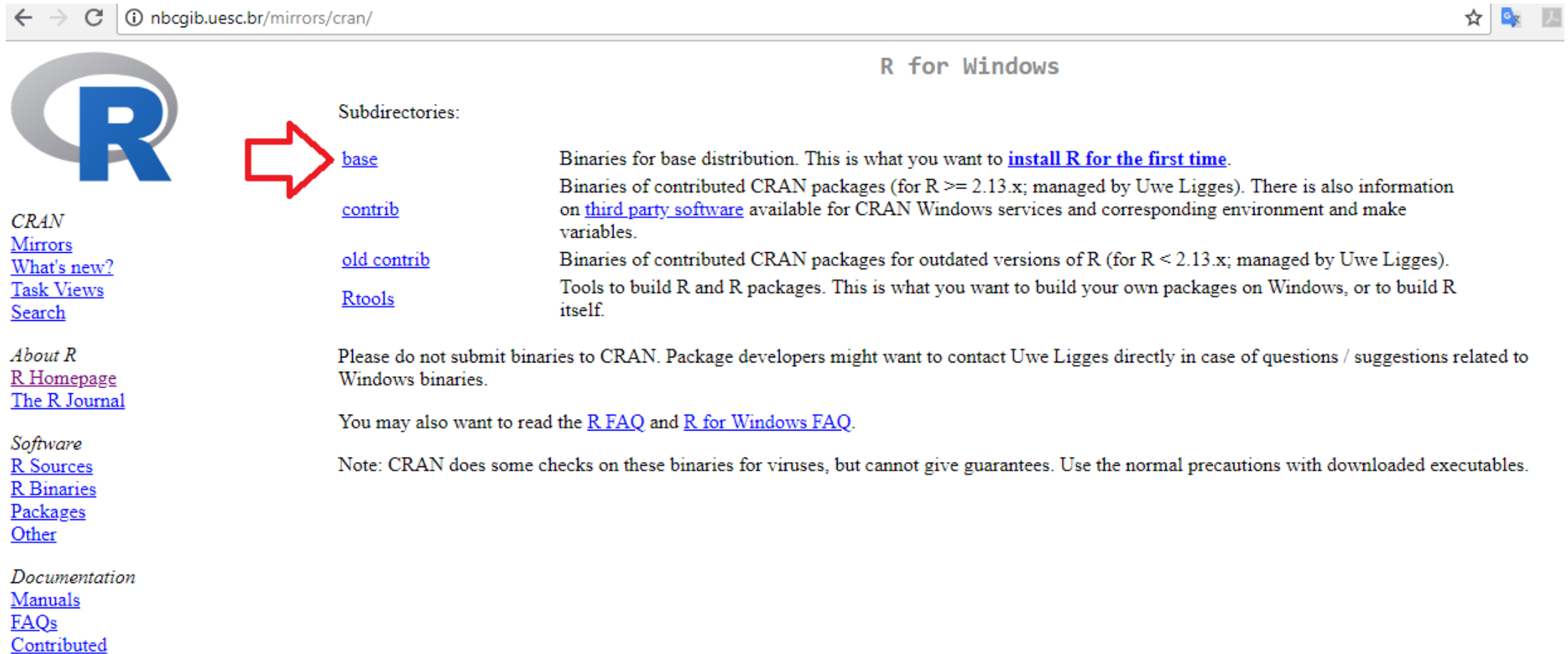
## Source Code for all Platforms

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 (2018-07-02, Feather Spray) [R-3.5.1.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


# 4º Selecione a opção “base” e faça o download



The screenshot shows a web browser window with the address bar displaying `nbcgib.uesc.br/mirrors/cran/`. The page title is "R for Windows". On the left, there is a large blue "R" logo. Below the logo, there are several 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". In the center, under the heading "Subdirectories:", there are four links: "base", "contrib", "old contrib", and "Rtools". A red arrow points to the "base" link. To the right of the "base" link, there is a paragraph of text: "Binaries for base distribution. This is what you want to [install R for the first time](#). Binaries of contributed CRAN packages (for R >= 2.13.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 < 2.13.x; managed by Uwe Ligges). Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself." Below this, there is a note: "Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries." and another note: "You may also want to read the [R FAQ](#) and [R for Windows FAQ](#)." At the bottom, there is a note: "Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables."

← → ↻ ⓘ nbcgib.uesc.br/mirrors/cran/ ☆

## R for Windows



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](#)

Subdirectories:

- [base](#)
- [contrib](#)
- [old contrib](#)
- [Rtools](#)





Binaries for base distribution. This is what you want to [install R for the first time](#).  
Binaries of contributed CRAN packages (for R >= 2.13.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 < 2.13.x; managed by Uwe Ligges).  
Tools to build R and R packages. 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 Uwe Ligges directly in case of questions / suggestions related 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.

# Instalando Rstudio





[Fazer login](#)

[Todas](#) [Vídeos](#) [Imagens](#) [Notícias](#) [Livros](#) [Mais](#) [Configurações](#) [Ferramentas](#)

Aproximadamente 5.760.000 resultados (0,25 segundos)

**RStudio – Open source and enterprise-ready professional software for R**  
<https://www.rstudio.com/> ▼ Traduzir esta página  
RStudio is an active member of the R community. We believe free and open source data analysis software is a foundation for innovative and important work in ...





[Download RStudio](#)  
RStudio is a set of integrated tools designed to help you be ...

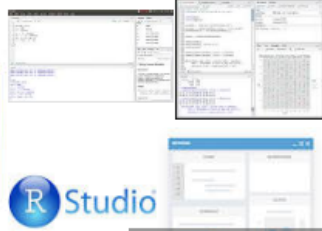



[RStudio Server](#)  
Server. Centralize access and computation. RStudio Server.

[Download R Studio](#)  
RStudio is a set of integrated tools designed to help you be ...

[R Packages](#)  
The RStudio team contributes code to many R packages and ...

[Online Learning](#)  
Online learning. A wealth of tutorials, articles, and examples ...

[Shiny](#)  
Gallery - Tutorial - Hosting and deployment - Reference - Articles

[Mais imagens](#)

## RStudio

Software

RStudio é um software livre de ambiente de desenvolvimento integrado para R, uma linguagem de programação para gráficos e cálculos estatísticos.  
[Wikipédia](#)

Linguagem: C++, Qt



# Selecione a opção baixar e escolha seu Sistema operacional

 **RStudio** conjunto de ferramentas integradas projetadas para facilitar a vida do usuário. O RStudio é um editor de destaque de sintaxe que suporta execução direta de código e uma variedade de ferramentas robustas para plotagem, visualização de histórico, depuração e gerenciamento de seu espaço de trabalho. Saiba mais sobre os recursos do RStudio.

Produtos Recursos Preços Sobre nós Blogs

Área de trabalho do RStudio  
Licença Open Source

LIVRE

 **BAIXAR**  
Saber mais

Área de trabalho do RStudio  
Licença comercial

US \$ 995 por ano

**COMPRAR**  
Saber mais

Servidor RStudio  
Licença Open Source

LIVRE

**BAIXAR**  
Saber mais

RStudio Server Pro  
Licença comercial

US \$ 9.995 por ano

**BAIXAR**  
Saber mais

RStudio Server Pro + RStudio Connect  
Licença comercial

US \$ 29.995 por ano

**CONVERSA**  
Saber mais

Ferramentas Integradas para R	●	●	●	●	●
Suporte prioritário		●		●	●
Acesso via navegador da web			●	●	●
Segurança Empresarial				●	

Try RStudio Server Pro for free!

Source on Save Run Source

plotabin Next Prev Replace Replace All

☐ In selection ☐ Match case ☐ Whole word ☐ Regex ☒ Wrap

```
73 n=10
74 theta = 0.5
75 alfa = 0.1
76 monoc=FALSE
77 plotabin(arq, n, theta, alfa, monoc)
78
79
80 png(filename='binomial01.png', width = 600, height = 400, pointsize = 1
81 par(mai=c(1.2,1.1,0.5,0.2))
82 # Distribuicao Binomial com n=10 e theta=1/2
83 n=10
84 theta = 0.5
85
86 x = 0:n
87 f = dbinom(x, size=n, prob=theta)
88 barplot(f, names.arg=x, main='Binomial Distribution(10, 0.5)', cex.main=
89 xlab='x', ylab='P(X=x|theta)')
90 abline(0,0)
91
92 dev.off()
93
94
95
96
97 png(filename='dist_normal03.png', width = 600, height = 400, pointsize =
98 par(mai=c(1.2,1.1,0.5,0.2))
99 # Distribuicao Normal Padrão
100 x = seq(-3.5, 3.5, by=0.05)
101
```

Editor

Environment History

To Console To Source

colnames(v)  
n=10  
theta = 0.5  
x = 0:n  
f = dbinom  
barplot(f, names.arg=x, main='Binomial Distribution(10, 0.5)', cex.main=1,  
xlab='x', ylab='P(X=x|theta)')  
abline(0,0)

```
2 2 b
3 3 r
> v[[2]]
[1] b b r
Levels: b r
> v = data.frame(a=1:3, b=c('b', 'b', 'r'), stringsAsFactors=FALSE)
> v[[2]]
[1] "b" "b" "r"
> v[,2]
[1] "b" "b" "r"
> v$b
[1] "b" "b" "r"
> v[, 'b']
[1] "b" "b" "r"
> colunas = c('a', 'b')
> v[colunas]
  a b
1 1 b
2 2 b
3 3 r
> colnames(v)
[1] "a" "b"
> n=10
> theta = 0.5
```

Console do R

Files Plots Packages Help Viewer

Zoom Export Clear All

Binomial Distribution(10, 0.5)

A bar chart showing the probability mass function of a binomial distribution with n=10 and p=0.5. The x-axis is labeled 'x' and ranges from 0 to 10. The y-axis is labeled 'P(X=x|theta)' and ranges from 0.00 to 0.15. The bars are centered at x=5, which has the highest probability (approximately 0.246).

Diretório, gráfico, pacotes, ajuda

# Sintaxe do R


---

- O R possui um conjunto de sintaxe com expressões simples, a linguagem faz distinções das letras maiúsculas e minúscula como também os caracteres "A" E "a" são compreendidos como símbolos diferentes ou variáveis. Já os comandos ou ordens são chamados de expressões ou atribuições.
- É importante lembrar que todas as funções terminam com (). Quando você quiser fechar o R, use a função quit() assim o programa perguntará se você quer salvar a área de trabalho. Escolha "Sim" se quiser continuar o trabalho da próxima vez que abrir o R na mesma pasta.
- Para conhecer o manual completo do R e dos pacotes adicionais instalados pode ser acessado com a função `help.start()`

Digitando help.start() temos essa seguinte tela no navegador

← → ↻ ⓘ 127.0.0.1:16715/doc/html/index.html ☆

---

Statistical Data Analysis 

---

**Manuals**

[An Introduction to R](#)  
[Writing R Extensions](#)  
[R Data Import/Export](#)

[The R Language Definition](#)  
[R Installation and Administration](#)  
[R Internals](#)

**Reference**

[Packages](#)

[Search Engine & Keywords](#)

**Miscellaneous Material**

[About R](#)  
[License](#)  
[NEWS](#)

[Authors](#)  
[Frequently Asked Questions](#)  
[User Manuals](#)

[Resources](#)  
[Thanks](#)  
[Technical papers](#)

**Material specific to the Windows port**

[CHANGES up to R 2.15.0](#)

[Windows FAQ](#)

# Usando algumas funções no R

Existe diversas funções no R para você deseja trabalhar , podemos pesquisar elas usando uma função seguinte **help.search()**. Vamos aprender como calcular logaritmos no R

---

Usamos os seguintes comandos:

```
> help.search("Logarithms") #Buscar informações sobre logaritmos
```

Nas versões mais recentes do R a função help.search() pode ser substituída por apenas ?? (duas interrogações)

```
> ??logarithms
```

É possível buscar ajuda na internet, no website do R usando a função RSiteSearch()

```
> RSiteSearch("logarithms") # abre uma página na internet, mas só funcionará se seu computador estiver conectado à internet.
```

Caso você deseja ver os arquivos de ajuda basta usar o comando help(nome.da.função)

```
>help(log) OU >?log
```

Vamos usar a **função sqrt** que é uma função para calcular a raiz quadrada de números sugiro que use a função **help** para entender sobre o arquivo

> sqrt(9) # Tira a raiz quadrada dos argumentos entre parênteses, no caso 9

> sqrt(3\*3^2) # raiz quadrada de 27

---

> sqrt((3\*3)^2) # raiz quadrada de 81

> prod é a função para multiplicação

> prod(2,2) # O mesmo que 2x2

> prod(2,2,3,4) # 2x2x3x4

log é a função para calcular o logaritmo

> log(3) # log natural de 3

> log(3,10)# log de 3 na base 10

> log10(3)# o mesmo que acima! log 3 na base 10

# Tipos de dados

- Existe quatro tipos de dados no R : **numéricos** , **caracteres** , **lógicos** e **números complexos**. No R cada objeto possui dois atributos dado por tipo (mode) e tamanho (length) logo , essas informações são relevantes para manipulação de dados abaixo vejamos alguns

```
1  #Númerico
2  idade <- 25
3  idade
4  25
5  #Caracteres
6  string<-"Hello word !"
7  string
8
9  #Lógicos
10 4<7
11 TRUE OU FALSE ?
12 #Números complexos
13 bc <- 3 + 2i
14
```

# Objetos e operações

---

- Atribuição de Valores
- Comandos Auxiliares
- Operações e funções matemáticas no R



# Atribuindo valores

Podemos atribuir valores para algumas variáveis no R utilizaremos os símbolos "`->`" e "`=`" para atribuição de valores , abaixo vejamos alguns exemplos

```
1 x<-40 #x é a variável que recebe o valor 40;
2 0.68->x #x é a variável que recebe o valor 0.68;
3 x = -45 #x é a variável que recebe o valor -45;
4 assign("x", 6i) #x é a variável que recebe o imaginário 6i;
```

Console

Terminal x



~/ ↩



```
> x<-40
> x
[1] 40
> 0.68->x
> x
[1] 0.68
> x=-45
> x
[1] -45
> assign("x",6i)
> x
[1] 0+6i
> |
```

# Tabela de comando auxiliares

Função	Descrição
<i>ls()</i> ou <i>objects()</i>	lista curta de variáveis definidas
<i>ls.str()</i>	lista detalhada de variáveis definidas
<i>str(x)</i>	ver informações detalhadas de <i>x</i>
<i>ls.str(ab)</i>	ver informações detalhadas sobre todas as variáveis com " <i>ab</i> " em seu nome
<i>rm(x)</i>	deletar variável <i>x</i>
<i>rm(x, y)</i>	deletar as variáveis <i>x</i> e <i>y</i>
<i>rm(list = ls())</i>	deletar todas as variáveis (limpar a workspace)
<i>class(x)</i>	ver que tipo de objeto é <i>x</i>
<i>q()</i>	sair do R com a opção de salvar a workspace em um arquivo ("Name.RData") e o histórico de comandos em outro arquivo ("Name.RHistory")
<i>ctrl + L</i>	no teclado, pressione "ctrl+L" para limpar a tela da console

# TABELA DE FUNÇÕES

Função R	Estatística
<code>sum()</code>	Soma de valores
<code>mean()</code>	Média
<code>var()</code>	Variância
<code>median()</code>	Mediana
<code>summary()</code>	Resumo Estatístico
<code>quantile()</code>	Quantis

Fonte: OLIVEIRA, Paulo Felipe de; GUERRA, Saulo; MCDONNEL, Robert. Ciência de Dados com R – Introdução. Brasília: Editora IBPAD, 2018.

# Tabela de funções matemáticas

Função	Descrição
$abs(x)$	valor absoluto de $x$
$log(x, b)$	logaritmo de $x$ com base $b$
$log(x)$	logaritmo natural de $x$
$log10(x)$	logaritmo de $x$ com base 10
$exp(x)$	exponencial elevado a $x$
$sin(x)$	seno de $x$
$cos(x)$	cosseno de $x$
$tan(x)$	tangente de $x$
$round(x, digits = n)$	arredonda $x$ com $n$ decimais
$ceiling(x)$	arredondamento de $x$ para o maior valor
$floor(x)$	arredondamento de $x$ para o menor valor
$length(x)$	número de elementos do vetor $x$
$sum(x)$	soma dos elementos do vetor $x$
$prod(x)$	produto dos elementos do vetor $x$
$max(x)$	seleciona o maior elemento do vetor $x$
$min(x)$	seleciona o menor elemento do vetor $x$
$range(x)$	retorna o menor e o maior elemento do vetor $x$

Fonte : [www.uft.edu.br/engambiental/prof/catalunha/arquivos/r/r\\_bruno.pdf](http://www.uft.edu.br/engambiental/prof/catalunha/arquivos/r/r_bruno.pdf)

# Operações e funções matemáticas no R

No R podemos escrever expressões aritméticas em que são construídas utilizando operadores e suas regras conforme mostraremos mais adiante, usamos também os operadores relacionais.

Operador	Descrição	Exemplo
+	adição	$2 + 3$
-	subtração	$2 - 3$
*	multiplicação	$2 * 3$
/	divisão (divisão inteira trunca o resultado)	$4 / 2$

Fonte : [pgdocptbr.sourceforge.net/pg80/functions-math.html](http://pgdocptbr.sourceforge.net/pg80/functions-math.html)

Símbolo	Nome do Operador	Exemplo	Significado
>	Maior que	$x > y$	x é maior que y?
>=	Maior ou igual	$x \geq y$	x é maior ou igual a y ?
<	Menor que	$x < y$	x é menor que y?
<=	Menor ou igual	$x \leq y$	x é menor ou igual a y ?
==	Igualdade	$x == y$	x é igual a y?
!=	Diferente de	$x != y$	x é diferente de y?

```
1 #Exemplo
```

```
2 > x=4; y=5; z=6
```

```
3 > h = 5*sqrt(6*x) + 5/(y-z) - x^6
```

```
4 > h
```

```
5 > -4076.505
```

```
6
```

# O que são objetos no R?

Segundo Landeiro, V. L. (2011) Existe muitos tipos de objetos portanto vamos conhecer apenas os básicos a seguir:

---

- Vetores: uma seqüência de valores numéricos ou de caracteres (letras, palavras).
- Matrizes: coleção de vetores em linhas e colunas, todos os vetores dever ser do mesmo tipo (numérico ou de caracteres).
- Dataframe: O mesmo que uma matriz, mas aceita vetores de tipos diferentes (numérico e caracteres). Geralmente nós guardamos nossos dados em objetos do tipo data frame, pois sempre temos variáveis numéricas e variáveis categóricas (por exemplo, temperatura , volume ,e nome de pessoas, respectivamente).
- Listas: conjunto de vetores, dataframes ou de matrizes. Não precisam ter o mesmo comprimento, é a forma que a maioria das funções retorna os resultados.
- Funções: as funções criadas para fazer diversos cálculos também são objetos do R. No decorrer da apostila você verá exemplos de cada um destes objetos.



# VETORES

---

- Para criar vetores usamos a função `vector()` que recebe o argumento `mode` e `length` (comprimento). Os argumentos podem ser do tipo `logical`, `numeric` e `character`. A função `c()` é usada para criar vetores com valores já pré-determinados, para guardar usamos o símbolo `<-`. Em algumas situações não teremos informações sobre os elementos do vetor logo, atribuímos o valor `NA`.
- Um outro comando para criar vetores é o `rep()` pode ser usado dois argumentos: `valor` a ser repetido e o `número de repetições`. Podemos também converter vetores em diferentes tipos usando `fator`.

```
1 vector(mode = "character", length = 4)
2 # [1] "" "" "" "" ""
3 vector(mode = "numeric", length = 5)
4 # [1] 0 0 0 0 0 0 0
5 vector(mode = "logical", length = 6)
6 # [1] FALSE FALSE FALSE FALSE
7 -----
8 rep("UFAL", 5)
9 rep(4, 9)
10 -----
11     codigo <- c(22, 23, 22, 23, 22, 23, 23, 22)
12 uf <- factor(codigo, levels = c(22, 23), labels = c("Piauí", "Ceará"))
13 uf
14 -----
```

É possível nomear elementos de um vetor recebendo nomes nesse caso , a função `names()` pode ser usada para atribuir nomes aos elementos do objeto conforme mostraremos abaixo. Outro exemplo útil em análises estatística é a função `sum()` pois podemos questionar nosso conjunto de dados :

**>Quais Estados do Nordeste com IDH acima e abaixo de 0.72 em 2005 ?**

```
1
2 idh05 <- c(0.677, 0.742, 0.723, 0.683, 0.718, 0.718 )
3 names(idh05) <- c("AL", "BA", "CE", "MA", "PB", "PE")
4 idh05
5 -----
6     sum(idh05 > 0.72)
7     sum(idh05<0.72)
8     -----|
```

❑ Vamos agora aprender a criar dataframe sendo uma classe de objetos mais utilizadas pois esse é formato que o R as tabelas importadas de outros softwares como bloco de notas , excel , SPSS

---

para criamos um dataframe usamos a função data.frame

```
> tabela1<-data.frame(Cursos=c("Economia","Ciências  
Contábeis","Administração"),Alunos_Matriculados=c(221,334,267))
```

```
> tabela1
```

❑ Podemos identificar os valores da tabela "nome" basta digitar tabela\$Nome ou tabela1[2] , já para criar uma nova coluna no dataframe basta repetir o nome do objeto e a tabela que você deseja criar tabela1\$nota\_enade<-c(5.8,6.2,7.2).

# MATRIZES

As matrizes no R podem ser criadas usando duas funções importantes , as funções `cbind()` [ **JUNTAR COLUNAS** ] E `rbind()` [ **JUNTAR LINHAS** ]. Abaixo é possível visualizar um exemplo usando as duas funções , podemos também salvar nossas matrizes resultante das funções utilizando `colnames()` para atribuir nomes às colunas e `rownames()` as linhas.

```
9
10     x <- c(8, 7, 3, 4, 1)
11     y <- c(5, 2, 3, 8, 3)
12     z <- c(10, 6, 5, 3, 2)
13     cbind(x, y, z)
14     -----
15     m <- cbind(x, y, z)
16     colnames(m) <- c("preço", "quantidade", "receita")
17     rownames(m) <- c("Gilson", "José", "Maria", "Lucas", "Joana")
18
19 |
```

# Objetos vetores com caracteres (letras, variáveis categóricas).

---

- ❑ No R podemos criar objetos que contêm letras ou palavras ao invés de números. No entanto, as letras ou palavras devem vir entre aspas " ".

Ex:

```
municipios<-c("Maceio","Arapiraca","Coruripe","Feira Grande","Coité do  
Noia","Igaci")
```

```
Estados<-c("AL","BA","RN","SP","RJ")
```

# Criando objetos

- Objetos vetores com valores numéricos : Vamos criar um conjunto de dados que contém o número de funcionários coletados de **10 empresas em Maceió.**
- 

```
> funcionarios<-c(22,20,37,37,13,24,45,5,33,32)
```

- O comando **<-** (sinal de menor e sinal de menos) indica assinalar (assign). representa tudo que vem após este comando será salvo com o nome que vem antes. É o mesmo que dizer "salve os dados a seguir com o nome funcionários .
- A letra **c** significa **concatenar** (colocar junto). Entenda como "**agrupe os dados entre parênteses dentro do objeto que será criado**" neste caso no objeto funcionários .
- Para visualizar os valores (o conteúdo de um objeto), digite o nome do objeto na linha de comandos. > funcionarios assim , a função length fornece o número de observações (n) dentro do objeto. > length(funcionarios )

# Estruturas de Condição no R

O R possui estruturas condicionais (SE , ENTÃO ETC) como outras linguagens de programação vamos usar esses operadores com a função `if()` vejamos abaixo como podemos executar um determinado código apenas se a condição for verdadeira e se a condição for falsa.

```
1  numero <- 1
2
3  if (numero == 1) {
4    cat("o numero é igual a 1")
5  } else {
6    cat("o numero não é igual a 1")
7  }
8  ## o numero é igual a 1
9
10 Ainda é possível encadear diversos if() else em sequência
11 numero <- 10
12
13 if (numero == 1) {
14   cat("o numero é igual a 1")
15 } else if (numero == 2) {
16   cat("o numero é igual a 2")
17 } else {
18   cat("o numero não é igual nem a 1 nem a 2")
19 }
20 ## o numero não é igual nem a 1 nem a 2
21 |
```



- Se quisermos executar um código quando a condição for verdadeira e outra for falsa usamos o "else" dada pela seguinte estrutura

```
1 if (condição) {  
2  
3     #Comandos a serem executados caso a condição nos parênteses seja verdadeira  
4  
5 } else {  
6  
7     #Comandos a serem executados caso a condição nos parênteses seja falsa  
8  
9 }
```

**Exemplo:** Vamos criar dois objetos a e b que seja numéricos e diferentes e informar o maior entre eles

```
1 #dois objetos numéricos  
2 a<-5  
3 b<-7  
4  
5 #Estrutura de condição  
6 if(a>b){  
7     cat(a)  
8 }else{  
9     cat(b)  
10 }
```

Imprima a saída na tela ou em um arquivo. Use cat para imprimir informações para um usuário final de uma função. cat também é útil para escrever informações que estão sendo processadas ou geradas, uma ou mais linhas de cada vez, em um arquivo.

**Exemplo 2:** Calcular a média do conjunto de dados: 10, 15, 9, 7, 6, 12, 17.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Fonte : FGV IBRE – Estatística com R

```
1 dados<-c(10,15,9,7,6,12,17)
2 n<-length(dados)
3 #ESTRUTURA DE REPETIÇÃO
4 soma<-0
5 for(i in 1:n){
6   soma<-dados[i]+soma
7 }
8 media<-soma/n
9 media|
```

# ESTRUTURA DE REPETIÇÃO

No R temos a estrutura de repetição definida pelo comando **for** em que significa "**fazer loopings**" dada pela seguinte estrutura :

```
for(i in 1:n) {  
  #comandos aqui  
} #fim do for i
```

Logo , isso quer dizer que o i vai calcular os comandos que estão entre as chaves. Já o "i in 1:n" indica os valores de i que serão respectivamente i=1 até i=n.

Vamos elaborar um exemplo intuitivo :

Usamos o for aqui para somar e elevar i valores ao quadrado:

```
1 s<-numeric(0)
2 for (i in 1:4) {
3     s[i]<-(i+1)^2
4
5
6 }
7 s #visualizar resultado
```

# Exemplo 1:

Considere a tabela abaixo referente á temperatura mínima em alguns municípios de Alagoas:

Classifique as temperaturas em:

- Baixa se menor do que 16
- Média se estiver entre 16 e 20
- Elevada se for maior do que 20

MUNICÍPIO	DATA	MÍNIMA
Água Branca	23/05/2018	18,2
Arapiraca	12/03/2018	19,2
Coruripe	14/06/2018	20,3
Maceió	05/06/2018	19,2
Palmeira dos Índios	24/05/2018	18
Pão de Açúcar	06/06/2018	18,1
Porto de Pedras	31/05/2018	20,5
São Luís do Quitunde	05/06/2018	19
Delmiro Gouveia	06/06/2018	16,8
Teotônio Vilela	26/05/2018	20

Fonte: <http://www.tnh1.com.br/noticias/noticias-detalle/alagoas/inverno-se-aproxima-e-temperatura-baixa-em-alagoas-minima-chega-a-168o/>

# Resolvendo...

```
1 #CRIANDO UM DATAFRAME
2 dados<-data.frame(temp=c(18.2,19.2,20.3,19.2,18,18.2,20.5,19,16.8,20))
3 #ATRIBUINDO O NIVEL
4 dados$nivel<-0
5 #EXECUTANDO
6 dados
7 #ESTRUTURA DE REPETIÇÃO
8 for (i in 1:7) {
9   ifelse(dados[i,1]<16,dados[i,2]<-"Baixa",ifelse(dados[i,1]<20,dados[i,2]<-"Média",dados[i,2]<-"Elevada"))
10
11 }
```

# Criando Funções no R

---

❑ Conforme vimos anteriormente a definição de função do R podemos criar nossas próprias funções essa tarefa pode ser trivial e variando de acordo com o objetivo da função.

❑ Sintaxe

Usamos o comando `function ( )`, que diz que estamos criando uma função

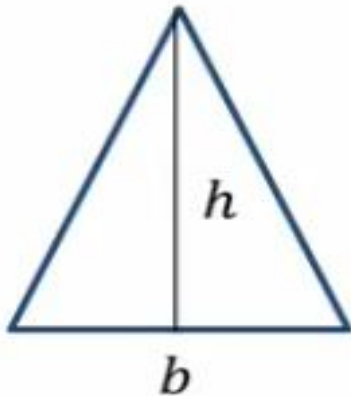
```
nome = function (argumento1 , ... , argumento n) {
```

Comandos da função

```
}
```

# Exemplo 1

**Exemplo 1:** Suponha que se queira calcular a área de um triângulo dado sua base e altura.



$$A = \frac{b \times h}{2}$$

Fonte de imagem : (NMEC) da FGV/IBRE



```
1 #Criamos uma função para efetuar o cálculo da área de um triângulo
2
3 area<-function(b,h){
4
5     A<-(b*h)/2
6     return(A)
7 }
8 #Para executar nossa função basta atribuir valores a B e H
9 area(4,7)
10 area(h=7,b=4)|
```

Fonte de imagem: Elaboração própria

# O que são pacotes no R

- Segundo VL Landeiro (2011 ) O R é um software bem leve ocupando pouco espaço na memória do seu PC , quando fazemos a sua instalação ele vem com as configurações mínimas de uso para seu funcionamento porém , quando precisamos fazer algumas tarefas complicadas é essencial instalar pacotes adicionais (packages) por exemplo para análise de dados , simulação computacional.

- A sintaxe dada para fazer a instalação do pacote é :

```
> install.package("nome_do_pacote") #dplyr
```

- Para fazer seu carregamento basta utilizar a expressão "library" da seguinte forma:

```
library(nome_do_pacote) # dplyr
```

```
require(nome_do_pacote)
```

# Leitura de dados no R

---

No R podemos fazer leitura de dados em diversos formatos realizar nossas tarefas como estatística descritiva , testes estatístico porém existe algumas funções também para fazer ler esses arquivos vejamos abaixo alguns formatos :



- CSV (Comma separated values)
- Dados em tabela, usando outros separadores que não a vírgula
- Dados espaçados por TAB



- Arquivos Xls (Excel)
- Linhas de arquivo de texto
- HTML, XML, Json
- E muitos outros (HDF5, SPSS, Stata)

---

CARREGANDO BANCO DE DADOS NO R

# Construindo banco de dados

Podemos construir nosso banco de dados no R usando um data.frame que é constituído pela concatenação de várias colunas numa única list. Assim como uma matrix, todas as colunas de um data.frame têm que ter o mesmo número de elementos, mas, ao contrário de uma matrix, um data.frame pode conter colunas de tipos diferentes, com variáveis numéricas, categóricas, booleanas etc. Podemos localizar no nosso data.frame pelo nome da coluna usando o \$ e ver um sumário dos nossos dados

```
> summary(b$estado.civil)
```

```
1
2  #CRIANDO AS VARIÁVEIS
3  sexo <- c("Masculino", "Feminino", "Feminino", "Masculino", "Feminino")
4  idade <- c(40, 37, 17, 13, 10)
5  estado.civil <- c("casado", "casado", "solteiro", "solteiro",
6                    "solteiro")
7  #CRIANDO UM DATA.FRAME COM AS VARIÁVEIS
8  b <- data.frame(sexo, idade, estado.civil)
```

O R trabalha com o conceito Working Directory mais conhecido como pasta de trabalho onde vai ler e escrever os dados. Para encontrar o diretório vamos usar o comando `getwd()` agora para informar o R qual pasta ele deve ler nossos arquivos usamos o `set.working directory`.

`setwd('D:/caminho/do/arquivo/arquivo.csv')`. No R base possui funções para ler arquivos mas vamos usar um pacote específico chamado "readr" lembrando que o pacote Tidyverse já inclui o pacote readr.

Assim usamos a função vistas anteriormente para carregar nosso pacote:

```
library(tidyverse) # carregar o readr
```

ou

```
library(readr)
```

# Working Directry

```
> getwd()
[1] "C:/Users/Gilson/Documents"
> b<-read.csv("C:/Users/Gilson/Documents/bd_exemplo2.csv")
> b
```

	uf	candidato	partido	logvotos	sexo
1	RO	ACIR MARCOS GURGACZ	PDT	12.257	Masculino
2	ES	AFONSO CEZAR CORADINE	PSOL	9.937	Masculino
3	DF	AGNELO SANTOS OLIVEIRO	PT do B	13.207	Masculino

## Import

readr  
readxl  
haven  
httr  
rvest  
xml2

## Tidy

tibble  
tidyr

## Program

purrr  
magrittr

## Transform

dplyr  
forcats  
hms  
lubridate  
stringr

## Visualise

ggplot2

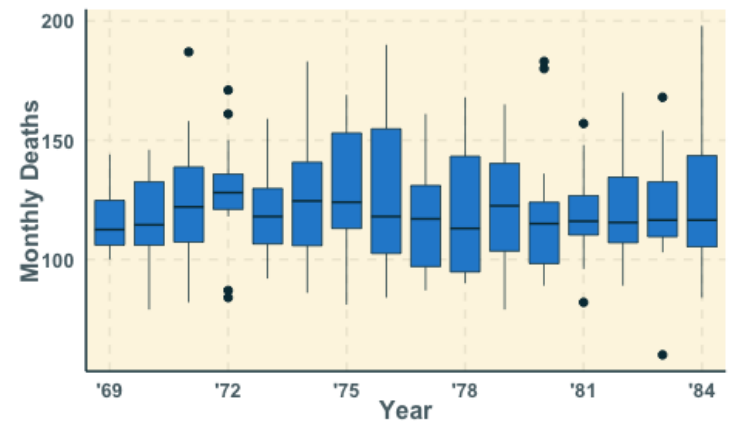
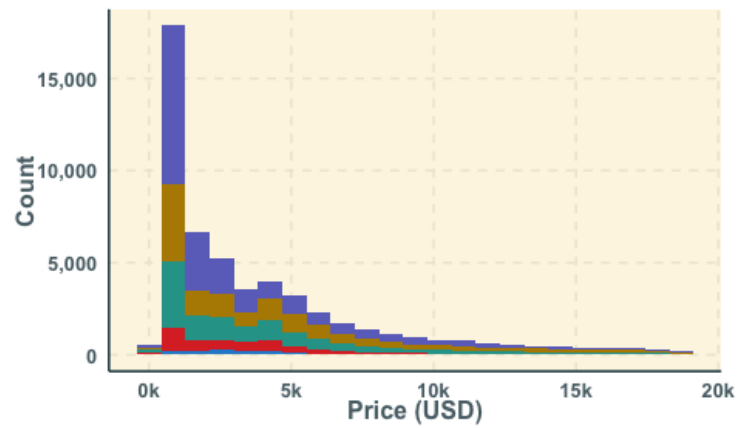
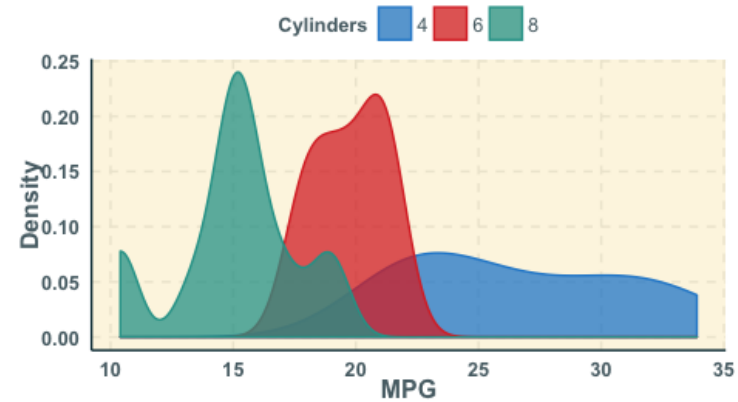
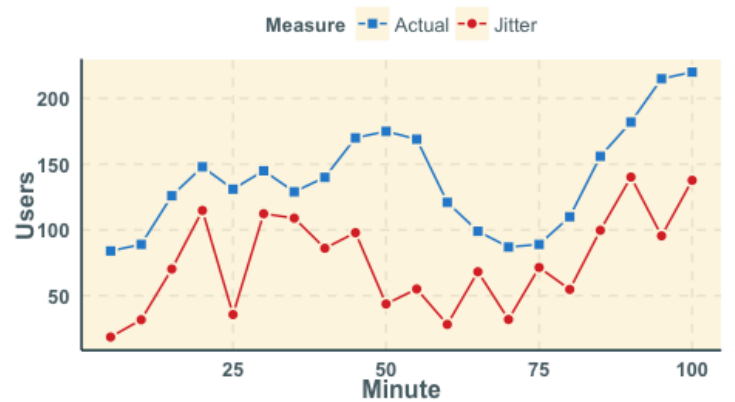
## Model

broom  
modelr



- Usamos o head , tail que são funções para visualizar "cabeça" e "rabo" do nosso banco de dados ou nossas amostras. Outros comandos importantes para começar investigar nossos dados são o str() e class() , summary().
- Quando carregamos uma base de dados para trabalho , certamente precisamos fazer algumas limpezas e manipulação afim de executar modelos econométricos, visualização de dados e produzir relatórios .
- Na ciência de dados alguns especialistas costuma afirmar que 80% do trabalho é encontrar uma boa fonte de dados para limpar e preparar esses dados e 20% restante é o resultado de aplicar modelos , análises .

# Introdução ggplot2



- 
- O ggplot2 é um pacote para visualizar dados criado por Hadley Wickham em 2005 baseado no seu livro "Grammar of Graphics" de Leland Wilkinson.
  - A curva de aprendizagem do ggplot2 é maior do que as funções gráficas tradicionais (plot, boxplot, hist) do R.
  - O ggplot2 pode ser plotado em um dataframe, sempre teremos que transformar dados em um dataframe.
  - O comando básico para gerar nossos gráficos é dado pela seguinte estrutura : `ggplot(dados, aes(x=Preditora, y=Resposta))`

# Carregando pacote

Console


Terminal x

~/

```
> library(tidyverse)
```

```
> mpg
```

```
# A tibble: 234 x 11
```



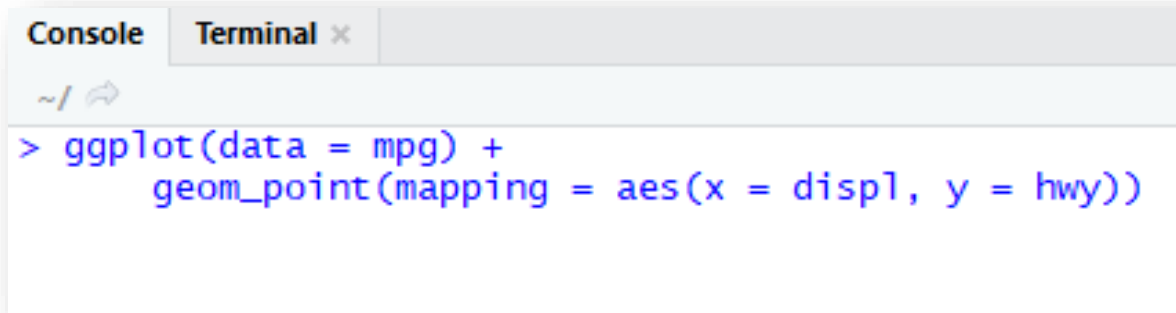
	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>	<int>	<chr>	<chr>
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

```
# ... with 224 more rows
```

```
>
```

Após instalar o pacote tidyverse vamos usar dataset mpg para plotar um ggplot , executamos o seguinte código:

---

A screenshot of a terminal window with two tabs: 'Console' and 'Terminal x'. The 'Console' tab is active, showing a prompt '~/' followed by the R code: 

```
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

```
Console Terminal x  
~/  
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

O gráfico mostra uma relação negativa entre o tamanho do motor (displ) e a eficiência de combustível (hwy). Usando o ggplot2 iniciamos com a função **ggplot()** criando assim sistemas de coordenadas em que podemos adicionar camadas. O primeiro argumento na função é **data=mpg** logo , a função **geom\_point()** adiciona camadas de ponto no nosso gráfico tornando assim um gráfico de dispersão . O pacote agrega muitas funções geométricas com diferentes camadas em um gráfico.

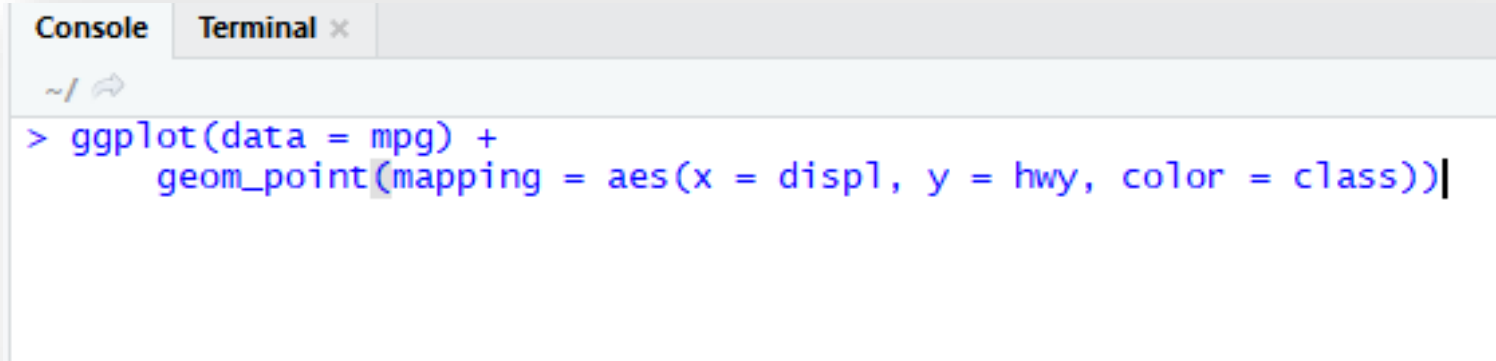
Outro argumento é o **mapping** define como as variáveis no conjunto de dados são mapeadas para propriedades visuais . No argumento também sempre atribuímos **aes()** e os argumentos x e y de aes() especificar quais variáveis mapear para os eixos x e y. ggplot2 procura as variáveis mapeadas no data argumento, neste caso mpg,.

# Exercícios

1. Execute `ggplot(data = mpg)`. O que você vê?
2. Quantas linhas estão em `mpg`? Quantas colunas?
3. O que a `drv` variável descreve? Leia a ajuda para `?mpg` descobrir.
4. Faça um gráfico de dispersão da `hwy` vs `cyl`.
5. O que acontece se você fizer um gráfico de dispersão de `class` vs `drv`? Por que o enredo não é útil?

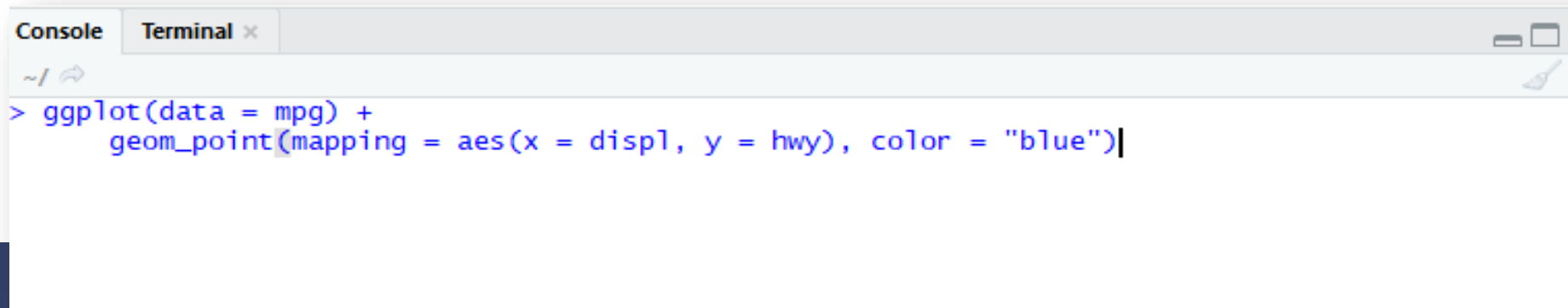
# Mapeamentos estéticos em gráficos

Usamos agora uma terceira variável chamada **class**, Por exemplo, você pode mapear as cores dos seus pontos para a class variável para revelar a classe de cada carro.



```
Console Terminal x
~/
> ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))|
```

Podemos *definir* as propriedades estéticas do nosso geom manualmente. Por exemplo, podemos fazer todos os pontos no nosso enredo azul:

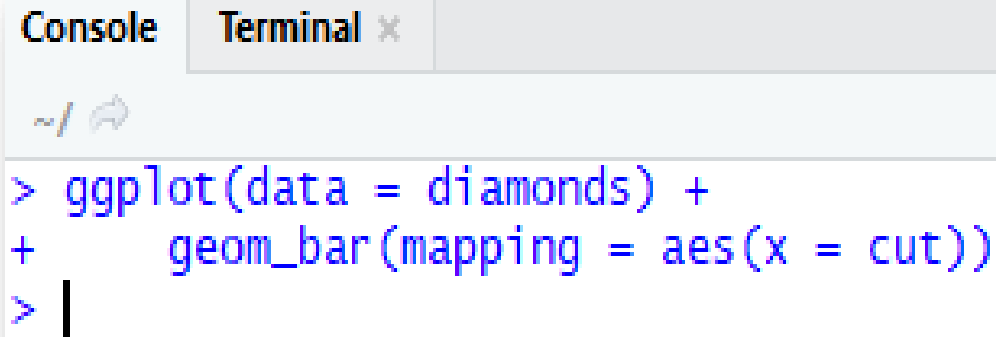


```
Console Terminal x
~/
> ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue")|
```

# Transformações estatísticas com ggplot

---

Vamos usar gráfico de barras usando a função `geom_bar()` exibe o número total de diamantes no Diamonds conjunto de dados, agrupados por cut. O Diamonds conjunto de dados vem em ggplot2 e contém informações sobre ~ 54.000 diamantes, incluindo o price, carat, color, clarity, e cut de cada diamante. O gráfico mostra que mais diamantes estão disponíveis com cortes de alta qualidade do que com cortes de baixa qualidade.



```
Console Terminal x
~/
> ggplot(data = diamonds) +
+   geom_bar(mapping = aes(x = cut))
> |
```



# Onde avançar com R ???

---



coursera



UDACITY



# Referências bibliográficas

---

- **R for Data Science-** Hadley Wickham, Garrett Grolemund
- de Aquino, Jakson Alves. "R para cientistas sociais." (2014).
- Landeiro, Victor Lemes. "Introdução ao uso do programa R." *Manaus: Instituto Nacional de Pesquisas da Amazônia* (2011).
- **OLIVEIRA, Paulo Felipe de; GUERRA, Saulo; MCDONNEL, Robert. Ciência de Dados com R – Introdução. Brasília: Editora IBPAD, 2018**