



# Configuração do ambiente de trabalho

Prof. Dr. Walmes Zeviani

**JUSTIÇA 4.0:** INOVAÇÃO E EFETIVIDADE NA REALIZAÇÃO DA JUSTIÇA PARA TODOS  
PROJETO DE EXECUÇÃO NACIONAL BRA/20/015

# Download e instalação do R

Configurando o ambiente de trabalho



# Download e instalação do R

## Roteiro

Instalar a linguagem R.

- ▶ É a **linguagem de programação**.
- ▶ Faz o processamento/análise de dados.

Downloads do R:  
<http://cran.r-project.org/>.

## Links para downloads

### ▶ Windows

- ▶ Baixar o `*.exe` e instalar com qualquer outro software.
- ▶ Instruções em: [base para Windows](#).
- ▶ Print screens do processo aqui: [r-installation-screenshots](#).

### ▶ Mac OS X

- ▶ Seguir instruções no CRAN: [Mac OS X](#).
- ▶ Tutorial de instalação (en): [install-r-and-rstudio-on-mac](#)

### ▶ GNU Linux

- ▶ Conforme a distribuição, seguir instruções em: [GNU Linux](#).

# IDEs/Editores para R

## Roteiro

Instalar um editor/IDE de preferência.

- ▶ **IDEs**: ambiente integrado de desenvolvimento.
  - ▶ RStudio.
  - ▶ RCode.
  - ▶ Tinn-R.
- ▶ **Editores**: são gerais e podem ser habilitadas para as linguagens usando plugins.
  - ▶ GNU Emacs, Spacemacs ou Doom-Emacs usando ESS.
  - ▶ Vim.
  - ▶ Visual Studio Code (VS Code).

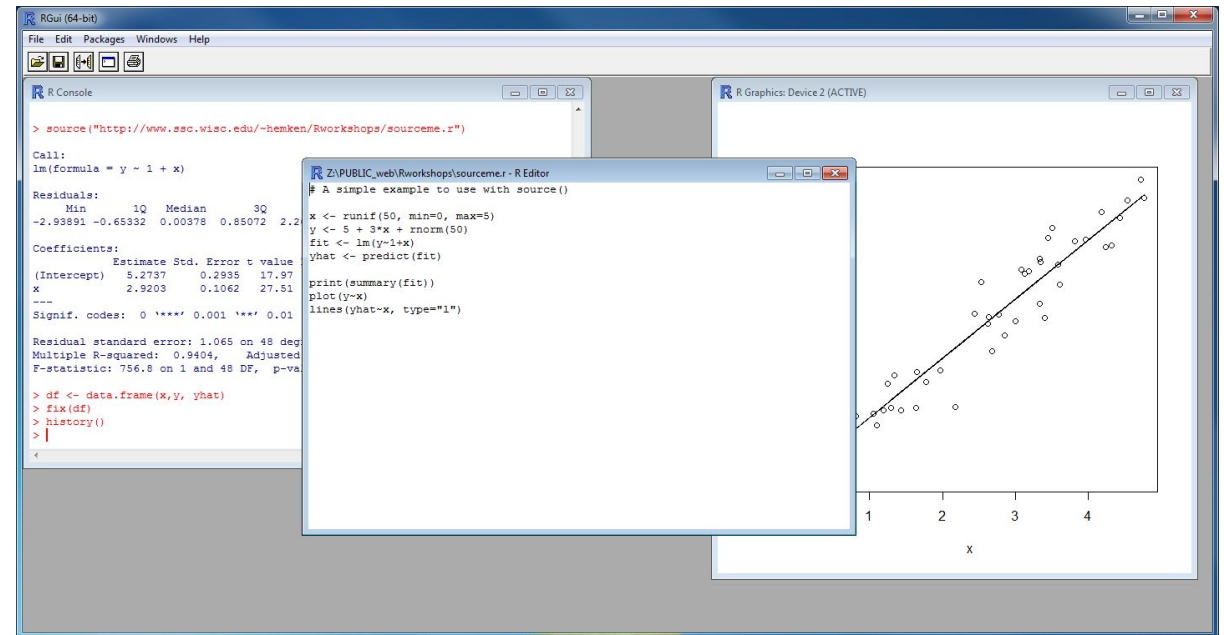
# Instalação do R no Ubuntu GNU Linux

- ▶ Instalação conforme documentação oficial do [CRAN](#).
- ▶ Executar em um Terminal.

```
# Adiciona chave.  
apt-key adv --keyserver keyserver.ubuntu.com \  
    --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9  
  
# Adiciona endereço à lista de repositórios.  
sudo add-apt-repository \  
    "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"  
  
# Atualiza lista de repositórios.  
sudo apt-get update  
  
# Faz a instalação do R.  
sudo apt-get install r-base
```

# R instalado com sucesso

- ▶ R instalado com sucesso!
- ▶ Procure nos aplicativos instalados.
- ▶ Windows: vem com a R Gui.
- ▶ Linux: acesso pelo Terminal.
- ▶ Trabalhar com editor padrão é contra produtivo.



Interface padrão do R para Windows.

# Download e instalação e uma IDE/Editor

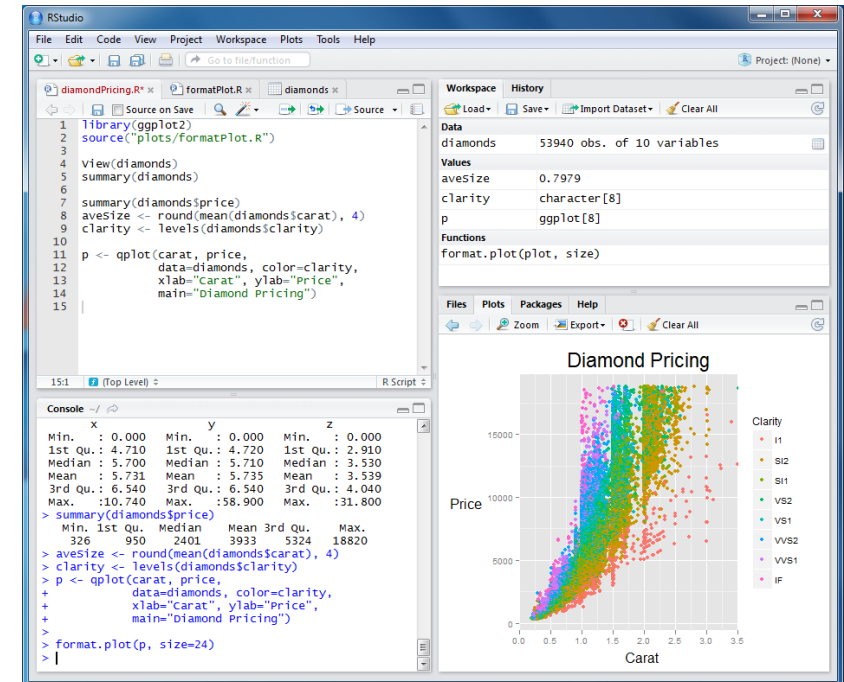
Usando o R em um ambiente mais produtivo



# RStudio IDE

## Detalhes

- ▶ É um ambiente mais popular para trabalhar com a linguagem R.
- ▶ É certamente o mais rápido de aprender.
- ▶ A IDE vem totalmente habilitada para o uso do R em todos os aspectos.
- ▶ Download do RStudio (recomendado):  
<https://www.rstudio.com/products/rstudio/download/>.



Print screen do RStudio.



# RStudio IDE

## Instalação

### ► Windows:

baixar `*.exe` e instalar.

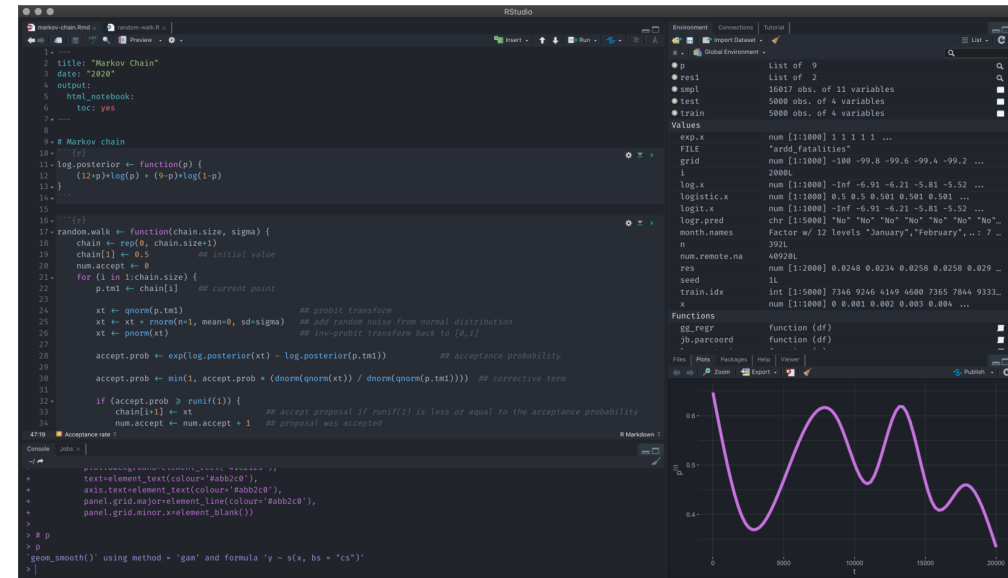
### ► Mac OS X:

descrição em [install-r-and-rstudio-on-mac](#).

### ► GNU Linux:

► Ubuntu: baixar o `*.deb` e instalar:  
`sudo dpkg -i <nome>.deb`.

► Outras distros: instalar via `tar.gz`  
ou `*.rpm`.



Print screen do RStudio em tela toda e ambiente escuro. [Fonte](#).

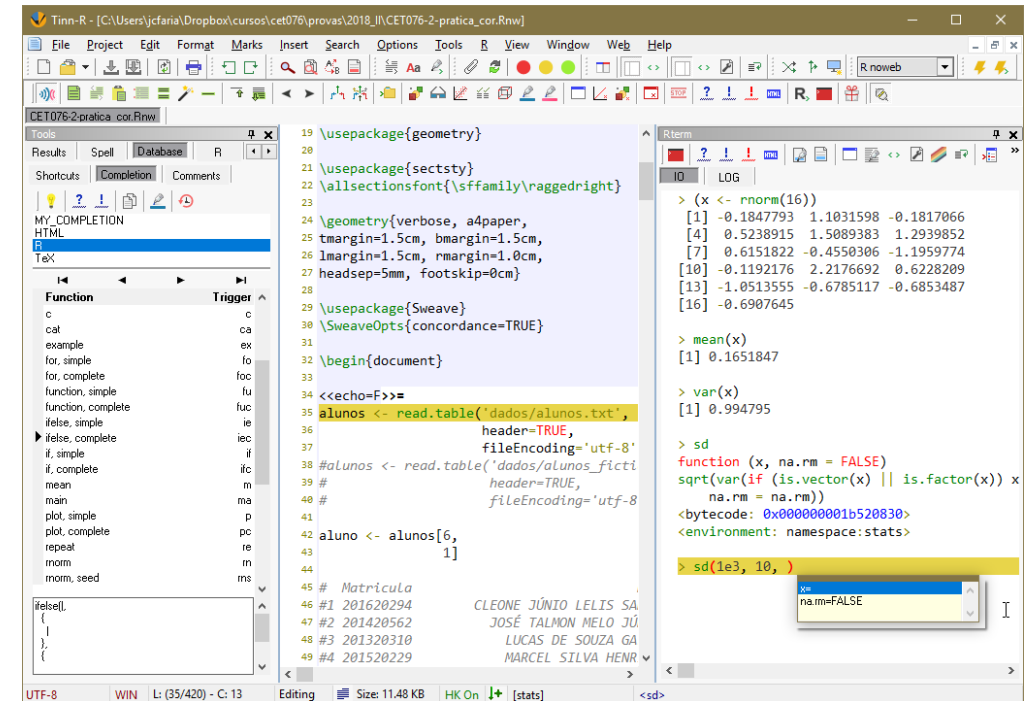
# Outros editores para R

Não custa pensar fora da caixinha



# Tinn-R (IDE)

- ▶ <https://tinn-r.org/pt/>.
- ▶ IDE para Windows.
- ▶ Desenvolvida por brasileiros.
  - ▶ Prof. Dr. José Cláudio Faria e colaboradores no LEC/UESC.
- ▶ Quer saber mais do Tinn-R, assista: <https://www.youtube.com/watch?v=INssoXdkOSI>.



The screenshot displays the Tinn-R IDE interface. The main editor window shows R code for loading packages, setting document options, and reading a data file. The console on the right shows the execution of the code, including the generation of random numbers and the calculation of standard deviation. The left sidebar contains a 'Tools' panel with a 'Completion' tab showing a list of functions and their triggers.

```
19 \usepackage{geometry}
20
21 \usepackage{sectsty}
22 \allsectionsfont{\sffamily\raggedright}
23
24 \geometry{verbose, a4paper,
25   tmargin=1.5cm, bmargin=1.5cm,
26   lmargin=1.5cm, rmargin=1.0cm,
27   headsep=5mm, footskip=0cm}
28
29 \usepackage{Sweave}
30 \SweaveOpts{concordance=TRUE}
31
32 \begin{document}
33
34 <<echo=F>>=
35 alunos <- read.table('dados/alunos.txt',
36   header=TRUE,
37   fileEncoding='utf-8')
38 #alunos <- read.table('dados/alunos_ficti
39 #
40 #   header=TRUE,
41 #   fileEncoding='utf-8')
42
43 aluno <- alunos[6,
44   1]
45
46 # Matricula
47 #1 201620294 CLEONE JÚNIO LELIS SA
48 #2 201420562 JOSÉ TALMON MELO JÚ
49 #3 201320310 LUCAS DE SOUZA GA
50 #4 201520229 MARCEL SILVA HENR
```

```
> (x <- rnorm(16))
[1] -0.1847793  1.1031598 -0.1817066
[4]  0.5238915  1.5089383  1.2939852
[7]  0.6151822 -0.4550306 -1.1959774
[10] -0.1192176  2.2176692  0.6228209
[13] -1.0513555 -0.6785117 -0.6853487
[16] -0.6907645

> mean(x)
[1] 0.1651847

> var(x)
[1] 0.994795

> sd
function (x, na.rm = FALSE)
sqrt(var(if (is.vector(x) || is.factor(x)) x
  na.rm = na.rm))
<bytecode: 0x00000001b520830>
<environment: namespace:stats>

> sd(1e3, 10, )
[1] 31.6227766
```

Print screen Tinn-R. Fonte.

# GNU Emacs

- ▶ <https://www.gnu.org/software/emacs/>.
- ▶ É um dos editores mais maduros e estáveis que existem, continua moderno, em constante evolução e amplo uso.
- ▶ Suporta inúmeras linguagens de programação e frameworks.
- ▶ O primeiro a ter suporte para o R.
- ▶ Download do Emacs + ESS:
  - ▶ Usuários Windows:  
<http://vgoulet.act.ulaval.ca/en/emacs/>.
  - ▶ Usuários Linux instalam por `sudo apt-get install emacs` ou via `*.tar.gz`.

```
blabla.R - Doom Emacs
estbas_pe
> 701_testes_quadrado_file $
> 702_testes_de_correlacao_fil $
> alunos
> config
> datasets
> fonts
> img
> rexams
> scripts
  amostragens-probabilistica $
  bla.R
  consistencia.R
  curl-sed.sh
  distribuicoes.gnumeric
  distribuicoes.ods
  functions.R
  intervalo-de-confianca.R
  rpanel_distributions.R
  video-with-subtitles.html
  video_views.txt
  youtube-dest-ere-captions. $
  youtube-rselenium-subtitle $
  ...

# ATTENTION: Número de visualizações por hora em cada dia.
tb_hora <- tb %>%
  mutate(dia = floor_date(ts, "day"),
         hora = hour(ts)) %>%
  # group_by(tit_wrap, ud, dia, hora) %>%
  group_by(tit_wrap, ud, dia) %>%
  summarise(n = max(views) - min(views))
  # complete(dia, hora = 0:23, fill = list("n" = 0))

ggplot(data = subset(tb_hora, dia > min(dia) & ud == 6),
       mappine = aes(x = dia, y = n)) +
  y = "Número de visualizações",
  color = "Titulo" +
  theme(legend.position = c(0.025, 0.975),
        legend.justification = c(0, 1))
+
Error: Problem with 'filter()' input '..1'.
✖ object 'ud' not found
i Input '..1' is 'ud == max(ud)'.
Run 'rlang::last_error()' to see where the error occurred.
>
floor_date: x=, unit=seconds, week_start=getOption("lubridate.week.start", 7) || date=
```

Print screen do Doom-Emacs com R. [Fonte](#).

- ▶ O pacote **ESS** (Emacs Speaks Statistics) dá suporte para R, S-Plus, SAS, Stata e Julia.
- ▶ Existem variantes como [Spacemacs](#) e [Doom-Emacs](#).

# Vim

- ▶ <https://www.vim.org/>.
- ▶ Assim como Emacs, é um editor maduro e estável, amplamente usado para inúmeros propósitos.
- ▶ Suporte para R com o plugin **Nvim-R**:  
<https://github.com/jalvesaq/Nvim-R>  
<https://nbcgib.uesc.br/lec/software/editores/nvim-r>  
[https://www.vim.org/scripts/script.php?script\\_id=2628](https://www.vim.org/scripts/script.php?script_id=2628).
- ▶ Plugins desenvolvido por brasileiros:
  - ▶ Jakson Alves de Aquino.
  - ▶ José Cláudio Faria.

The screenshot shows the Vim editor interface with a dark theme. The main window displays an R script named `get_links.R` with the following content:

```
4 library(httr)
3 library(magrittr)
2
1 # My API token is saved in an environment file
5 pinsecret <- Sys.getenv('pin_token')
1
2 # GET all my links in JSON
3 pins_all <- GET('https://api.pinboard.in/v1/posts/all',
4               query = list(auth_token = pinsecret,
5                           format = 'json'))
6
7 # Save the two responses to file for the analysis script to use
8 save(pins_all, file = 'pins.RData')
```

On the right side, there is a sidebar titled `.GlobalEnv | Libraries` showing a tree structure of objects in the global environment:

```
16 .GlobalEnv | Libraries
15
14 check_url_response
13 pins
12 | href
11 | description
10 | extended
9 | meta
8 | hash
7 | time
6 | shared
5 | toread
4 | tags
3 pins_all
2 pins_recent
1 responses
17 urls_with_errors
```

Below the sidebar, there is a section titled `Object_Browser` showing the following objects are masked from `'package:base'`:

```
intersect, setdiff, setequal, union
```

The main window also shows the output of the R script, displaying a table of links:

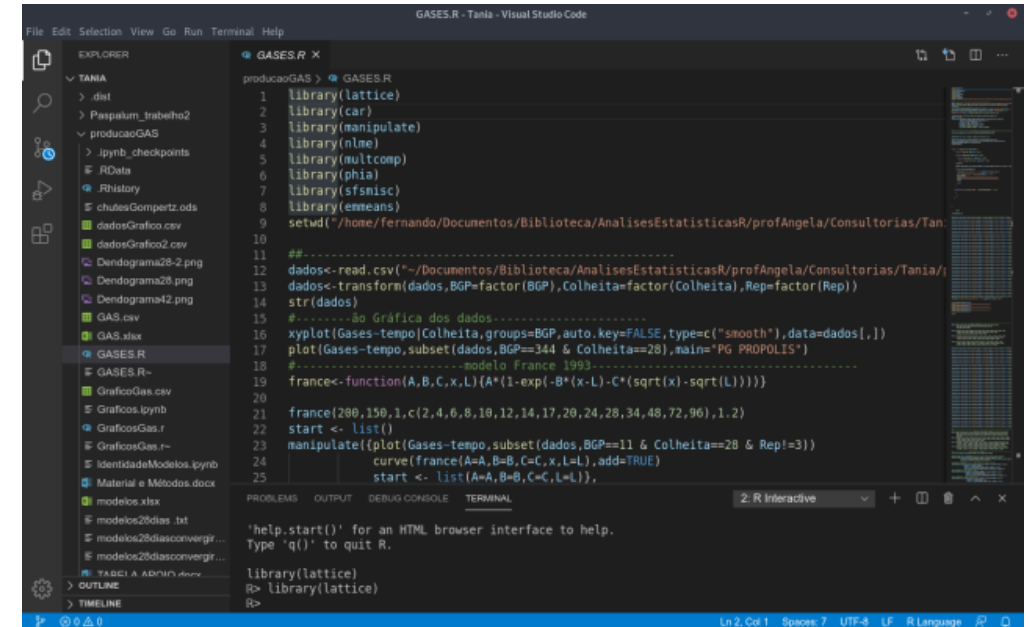
```
> pins %>% select(href, tags) %>% head()
1 href tags
2 http://varianceexplained.org/r/year_data_scientist/ data-science rstats
3 http://martinfowler.com/bliki/BoiledCarrot.html
4 http://blog.revolutionanalytics.com/2016/01/new-data-sources-for-r.html Data rstats api
5 http://daniellakens.blogspot.ca/2016/01/power-analysis-for-default-bayesian-t.html Statistics
6 https://thinkinator.com/2016/01/12/r-users-will-now-inevitably-become-bayesians/ machine-learning
7 https://xianblog.wordpress.com/2016/01/14/precision-in-mcmc/ MCMC
```

The status bar at the bottom shows `N 6603:R` and `1015:2`.

Print screen do Vim com R. [Fonte](#).

# VS Code

- ▶ <https://code.visualstudio.com/>.
- ▶ É um editor muito usado por programadores de aplicações para linguagens JavaScript, Python, etc.
- ▶ Suporte para linguagem R com plugins.
- ▶ Tutorial de configuração para R aqui: [Produção Animal com R](#).



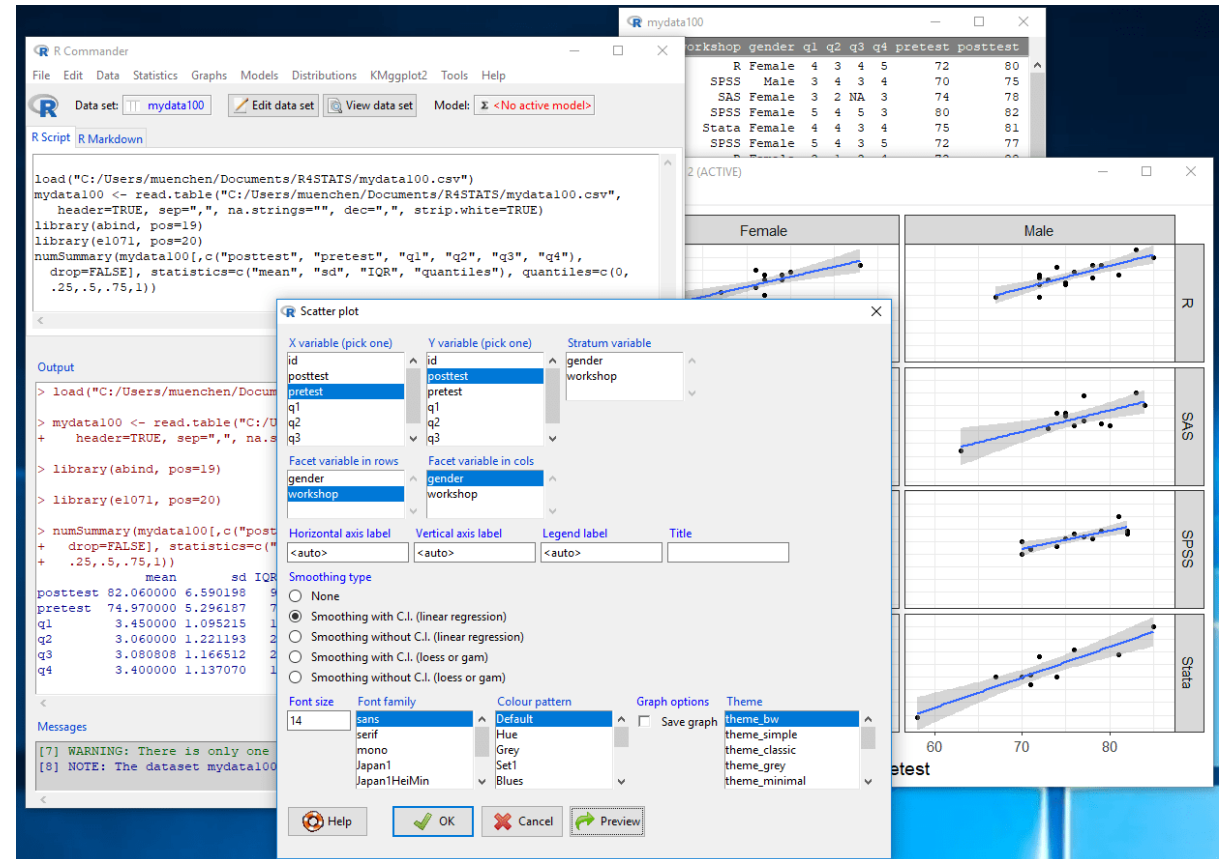
The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left displays a file tree for a project named 'TANIA', including folders like '.dai', 'Pawpawm\_trabeho2', and 'producaoGAS', along with various data files and scripts. The main editor window is open to a file named 'GASES.R', which contains R code for loading libraries, reading data, transforming variables, and creating a plot. The code includes comments in Portuguese. At the bottom, the 'TERMINAL' panel shows the R prompt and the execution of 'library(lattice)'. The status bar at the very bottom indicates 'Ln 2, Col 1', 'Spaces: 7', 'UTF-8', 'LF', and 'R Language'.

```
1 library(lattice)
2 library(car)
3 library(manipulate)
4 library(nlme)
5 library(multcomp)
6 library(phia)
7 library(sfsmisc)
8 library(emmeans)
9 setwd("~/home/fernando/Documentos/Biblioteca/AnalisesEstatisticasR/profAngela/Consultorias/Tania/producaoGAS")
10
11 ##
12 dados<-read.csv("~/Documentos/Biblioteca/AnalisesEstatisticasR/profAngela/Consultorias/Tania/dados")
13 dados<-transform(dados,BGP=factor(BGP),Colheita=factor(Colheita),Rep=factor(Rep))
14 str(dados)
15 #-----do Gráfica dos dados-----
16 xyplot(Gases-tempo|Colheita,groups=BGP,auto.key=FALSE,type=c("smooth"),data=dados[,])
17 plot(Gases-tempo,subset(dados,BGP==344 & Colheita==28),main="PG PROPOLIS")
18 #-----modelo France 1993-----
19 france<-function(A,B,C,x,L){A*(1-exp(-B*(x-L))-C*(sqrt(x)-sqrt(L)))}
20
21 france(200,150,1,c(2,4,6,8,10,12,14,17,20,24,28,34,48,72,96),1.2)
22 start <- list()
23 manipulate({plot(Gases-tempo,subset(dados,BGP==11 & Colheita==28 & Rep==3))
24               curve(france(A=A,B=B,C=C,x=L),add=TRUE)
25               start <- list(A=A,B=B,C=C,L=L)})
```

Print screen do VS code com R. [Fonte](#).

# R Commander

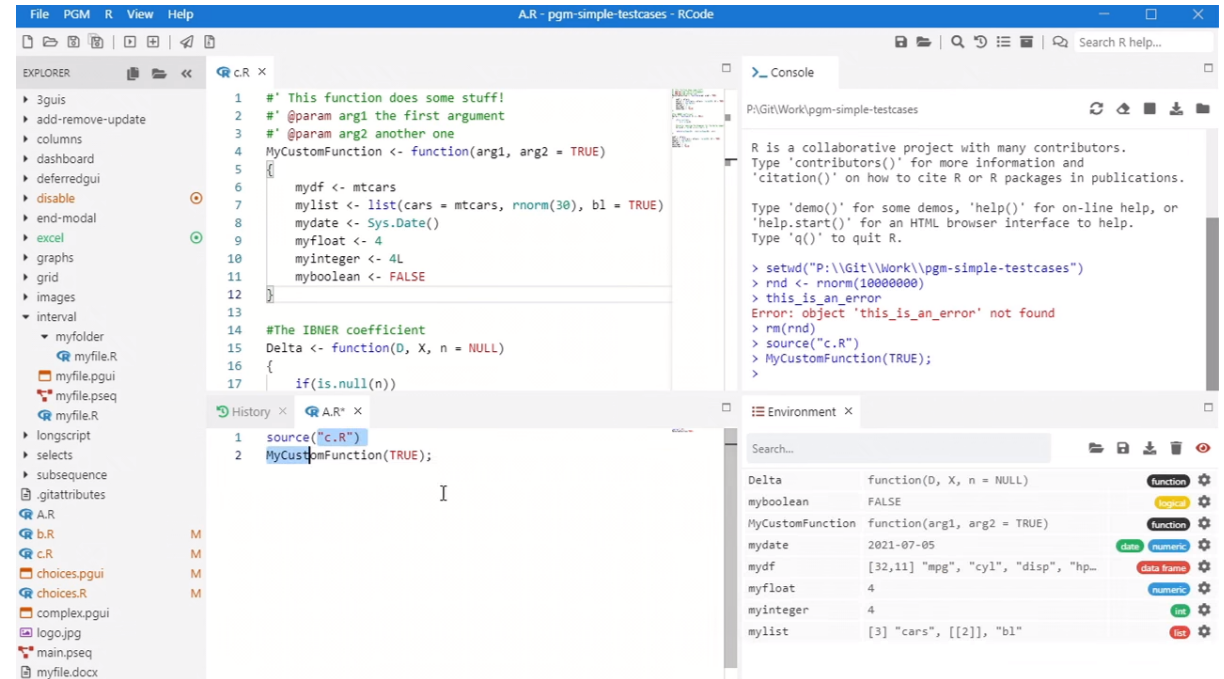
- ▶ É uma interface *point-and-click* para recursos do R.
- ▶ Assemelha-se aos softwares gráficos de Estatística.
- ▶ Menu com entradas para métodos e recursos.
- ▶ Feito para deixar o R mais acessível.
- ▶ Da forma como o **Rcmdr** foi feito, podem ser feitas outras aplicações.



Using the R Commander: A Point-and-Click Interface for R. [Fonte.](#)

# RCode

- ▶ <https://www.rcode.dev/>.
- ▶ É uma IDE para R e Python.
- ▶ Visualmente muito semelhante ao RStudio IDE.

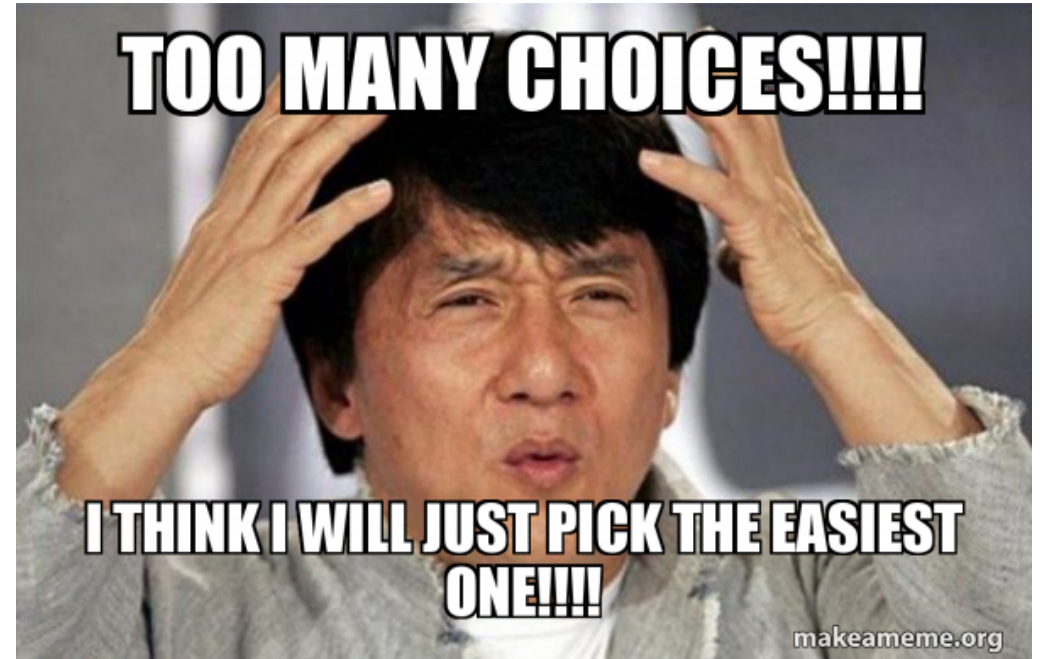


Interface do RCode, IDE para R e Python. [Fonte.](https://www.rcode.dev/)



# Outros editores

- ▶ Mais uma IDEs: [RCode](#).
- ▶ Existem vários outros editores que podem ser habilitados para R.
  - ▶ [Atom](#).
  - ▶ [RGedit](#): Gedit + R.
  - ▶ [RKWard](#) IDE.
  - ▶ [NppToR](#): R in Notepad++.
  - ▶ Etc.
- ▶ Mais editores de código:
  - ▶ [Wikipedia - R programming language editors](#)
  - ▶ [Sciviews - Editors](#)



Fonte.

# Recomendações sobre editores

## IDEs



- ▶ Vêm prontas para utilizar.
- ▶ Curva de aprendizado fácil.
- ▶ Funcionalidades à plena vista.
- ▶ Pouca customização adicional.
- ▶ São limitadas à linguagem específica.
- ▶ Nossa recomendação: **RStudio IDE**.

## Editores



- ▶ Recomendado para experientes.
- ▶ Trabalhar com múltiplas linguagens e de forma similar.
- ▶ Requer habilitar o editor para a linguagem.
- ▶ Possui alto nível de customização.
- ▶ Nossa recomendação: **Doom-Emacs** ou **VS Code**.

