

Disciplina

# Análise de Dados de Ciência do Solo no R – Aula 4

## INTRODUÇÃO AO R

Alan Rodrigo Panosso  
Departamento de Ciências Exatas

[alan.panosso@unesp.br](mailto:alan.panosso@unesp.br)



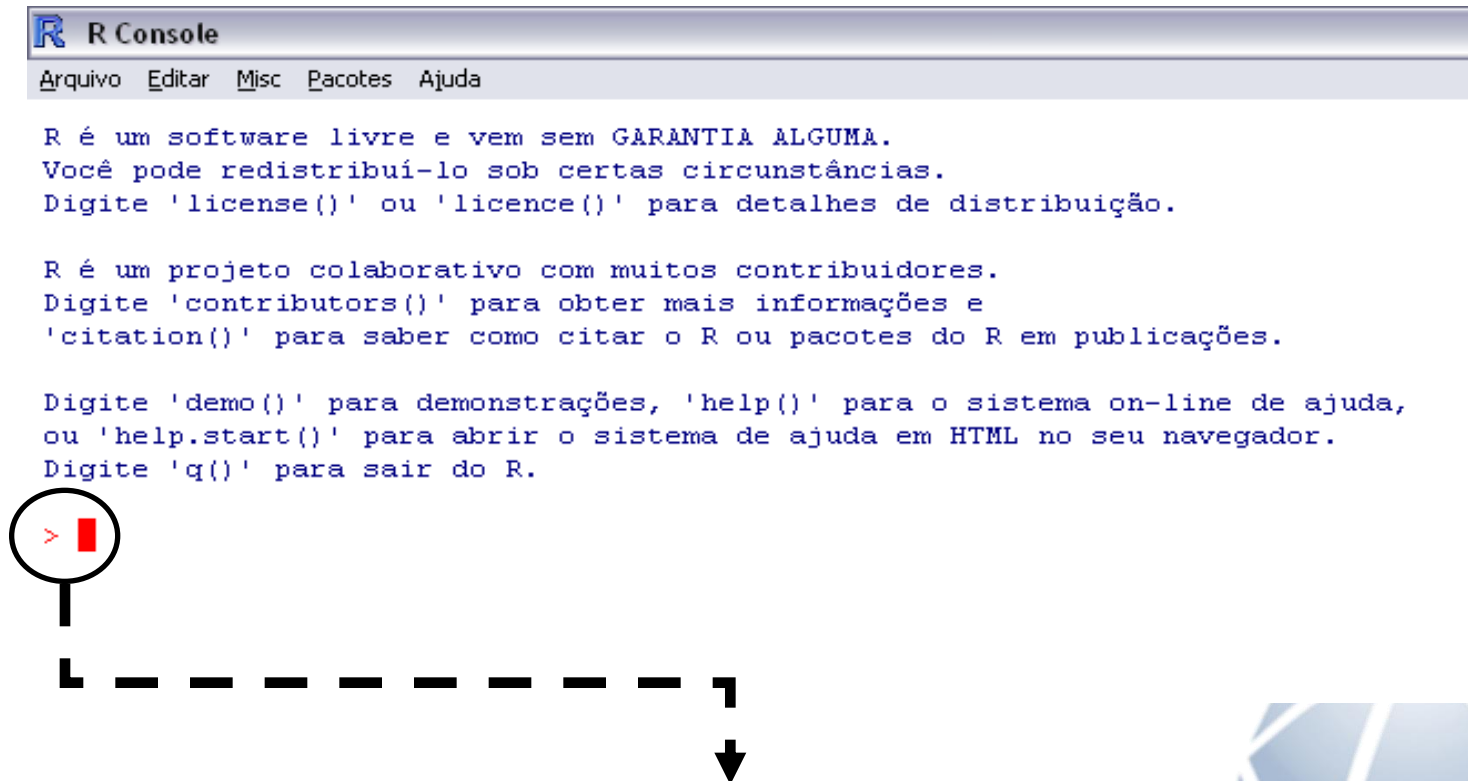
O que é o R

O R é uma linguagem de  
programação interpretativa

Todos os comandos digitados  
em seu ***prompt*** são  
***executados diretamente***  
pelo programa, sem a necessidade  
de uma compilação prévia.

# O que é o R

## Tela inicial do sistema R versão xxxxx:



Este símbolo (>) é o *prompt* do R e ele indica que o sistema está pronto para receber seus comandos. Exemplo `citation()`

O R é um software de ***domínio público***, utilizado para ***análise de dados*** em geral

# O que é o R

A sintaxe do R é simples e intuitiva, por exemplo, em uma análise de regressão linear, poder-se-á utilizar a função `lm(y ~ x)` o que significa "ajuste um modelo linear contendo y como variável dependente e x como variável independente". Se digitarmos apenas o nome da função, o R mostrará o seu conteúdo, a sua programação, exemplo, função `aov`, análise de variância.

Você pode redistribuí-lo sob certas circunstâncias.

Digite `'license()'` ou `'licence()'` para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.

Digite `'contributors()'` para obter mais informações e

`'citation()'` para saber como citar o R ou pacotes do R em publicações.

Digite `'demo()'` para demonstrações, `'help()'` para o sistema on-line de ajuda,

ou `'help.start()'` para abrir o sistema de ajuda em HTML no seu navegador.

Digite `'q()'` para sair do R.

[Área de trabalho anterior carregada]

```
> x<-c(2,4,6,8,10)
> y<-c(3,6,10,15,17)
> lm(y~x)
```

Call:

```
lm(formula = y ~ x)
```

Coefficients:

(Intercept)	x
-0.90	1.85

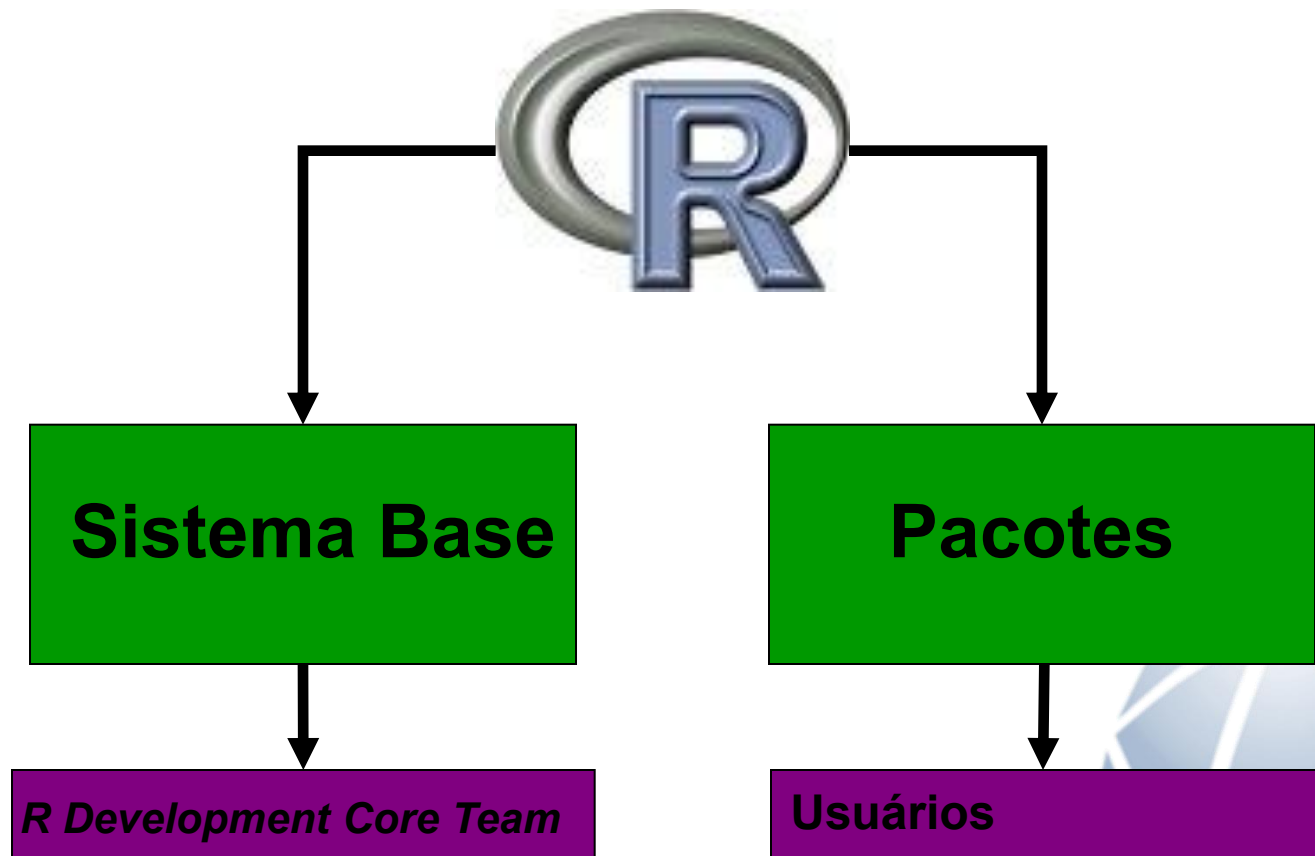
```
> |
```

```
> aov
function (formula, data = NULL, projections = FALSE, qr = TRUE,
  contrasts = NULL, ...)
{
  Terms <- if (missing(data))
    terms(formula, "Error")
  else terms(formula, "Error", data = data)
  indError <- attr(Terms, "specials")$Error
  if (length(indError) > 1L)
    stop(sprintf(ngettext(length(indError), "there are %d Error terms: only $",
      "there are %d Error terms: only 1 is allowed"), length(indError)),
      domain = NA)
  lmcall <- Call <- match.call()
  lmcall[[1L]] <- as.name("lm")
  lmcall$singular.ok <- TRUE
  if (projections)
    qr <- lmcall$qr <- TRUE
  lmcall$projections <- NULL
  if (is.null(indError)) {
    fit <- eval(lmcall, parent.frame())
    if (projections)
      fit$projections <- proj(fit)
    class(fit) <- if (inherits(fit, "mlm"))
      c("maov", "aov", oldClass(fit))
    else c("aov", oldClass(fit))
    fit$call <- Call
    return(fit)
  }
  else {
    if (pmatch("weights", names(match.call()), 0L))
      stop("weights are not supported in a multistratum aov() fit")
    opcons <- options("contrasts")
    options(contrasts = c("contr.helmert", "contr.poly"))
    on.exit(options(opcons))
    allTerms <- Terms
    errorterm <- attr(Terms, "variables")[[1 + indError]]
    eTerm <- deparse(errorterm[[2L]], width.cutoff = 500L,
      backtick = TRUE)
  }
}
```

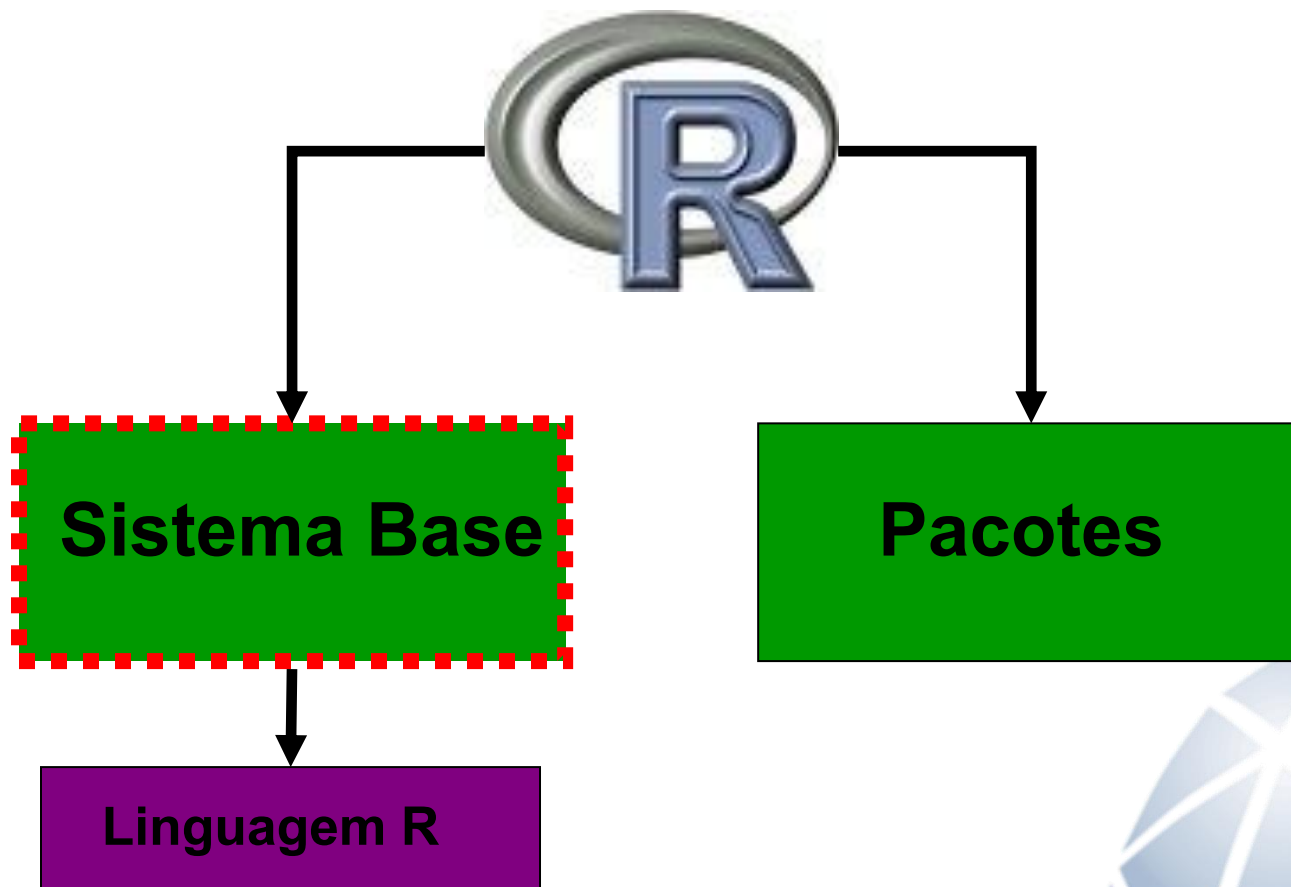
Apresenta  
***código fonte aberto***, ou  
seja, pode ser modificado  
ou implementado por  
usuários



# O que é o R



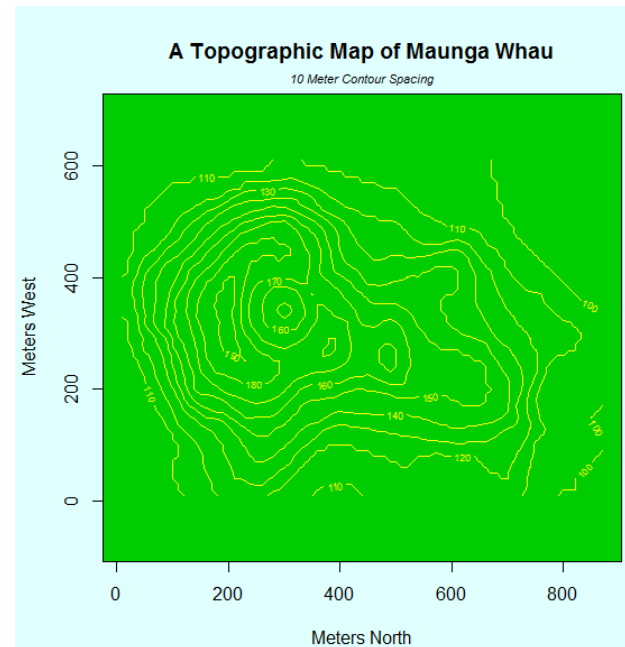
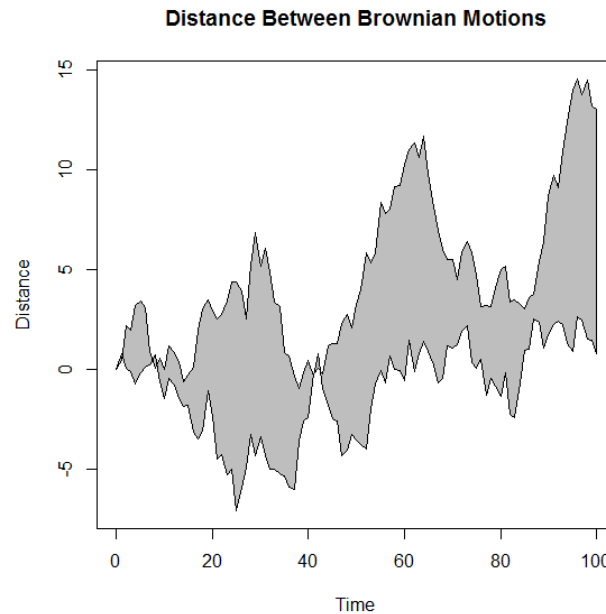
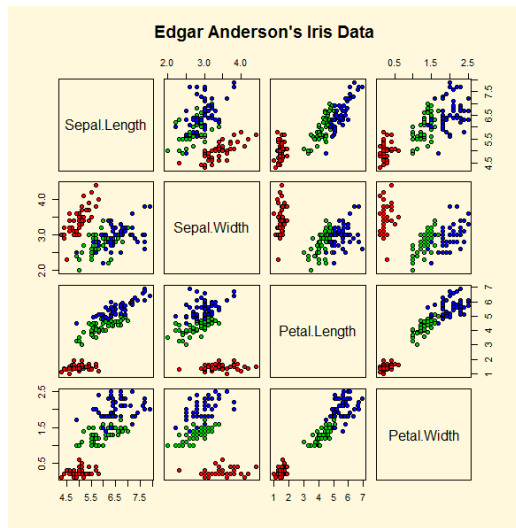
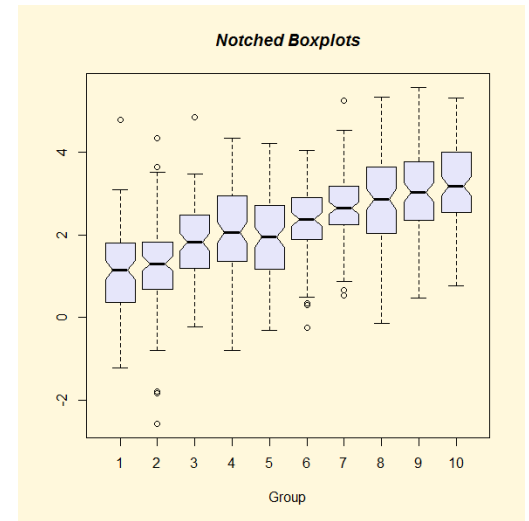
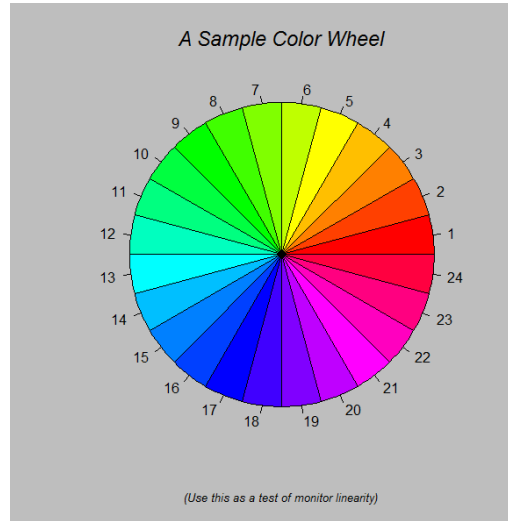
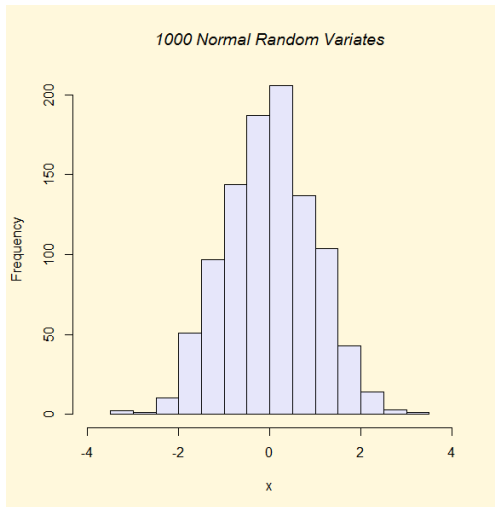
# O que é o R



*"R é uma **linguagem**  
**de alto nível** e um **ambiente**  
**para análise de dados**  
e **gráficos**"*

Crawley, 2007

# O que é o R

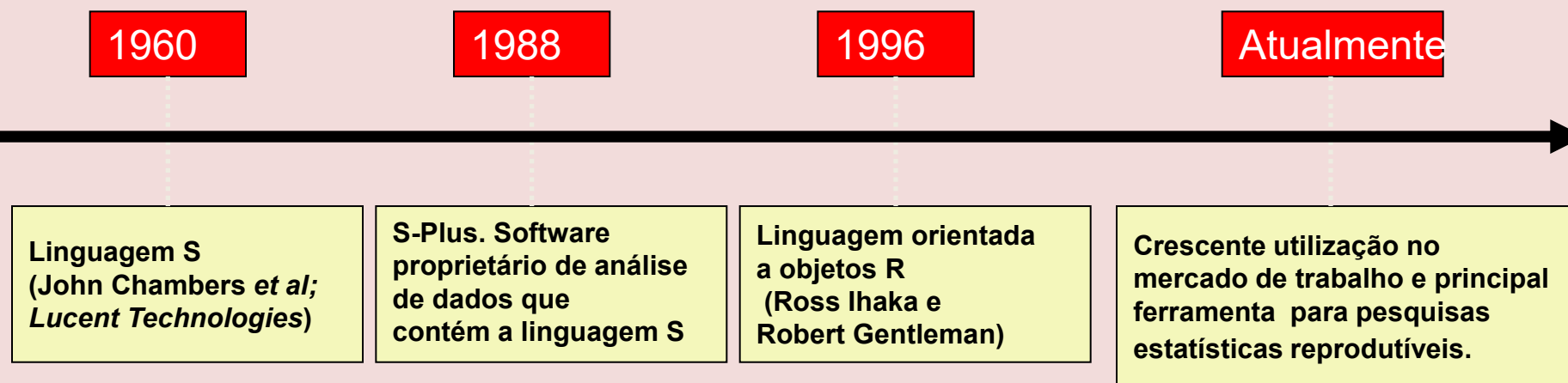


demo(graphics)

## *Adequação geral ao fluxo de trabalho:*

1. Adequado a trabalhos que necessitam de tradicionais linguagens de programação como **C**, **Fortran** e **Pascal**.
2. Similar a softwares matemáticos como **MATHEMATICA** e **MAPLE**, com várias bibliotecas para análises simbólicas.
3. Desempenho de softwares estatísticos com **SPlus** e **SAS** com várias bibliotecas para análise de dados.

## Confiabilidade



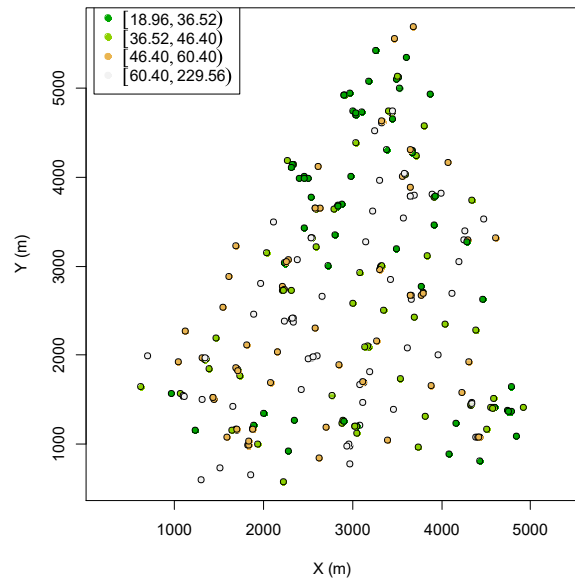
- O sistema R é um ambiente que incorpora uma implementação da linguagem de programação S, que é poderosa, flexível e possui excelentes facilidades gráficas (R Development Core Team);
- R é um projeto *open-source* e está disponível na internet sobre a *General Public License* ([www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html) e [www.fsf.org](http://www.fsf.org));
- Home-page oficial: [www.r-project.org](http://www.r-project.org)

## *Longevidade*

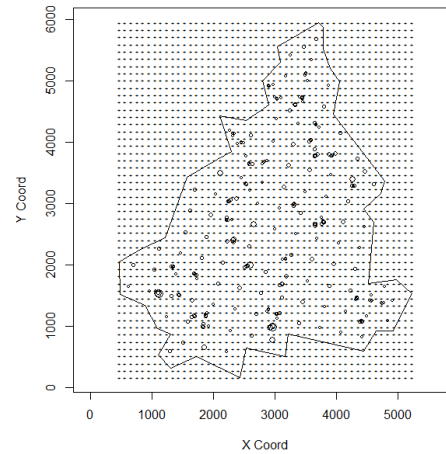
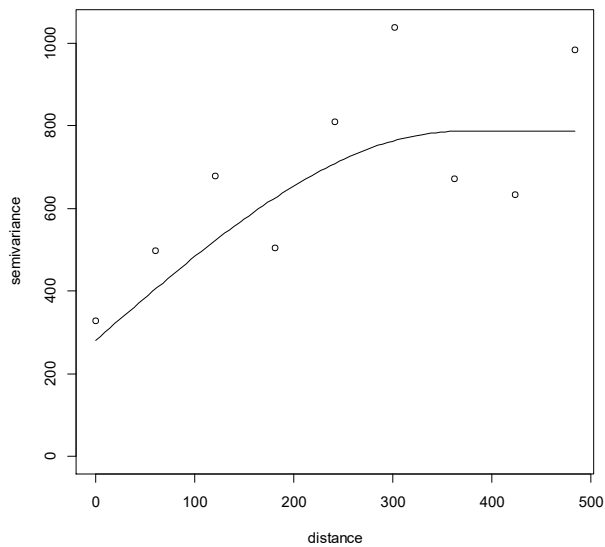
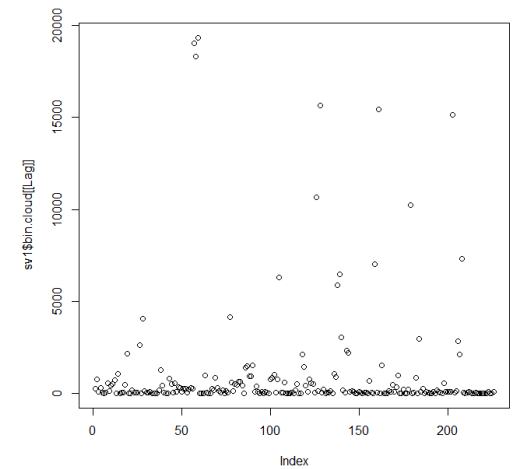
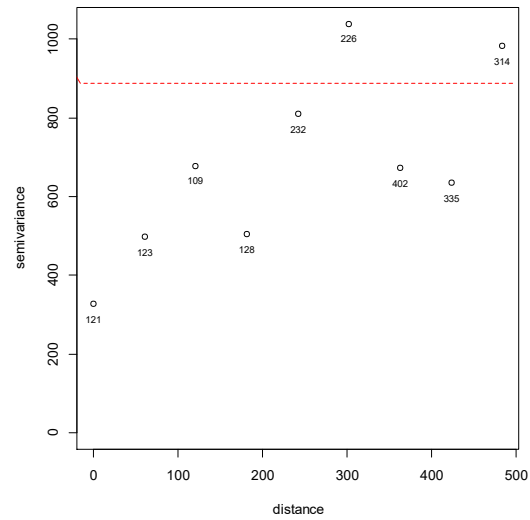
- A **Abrangência** e **disponibilidade** das mais variadas aplicações no campo da análise de dados, das simples às sofisticadas técnicas.
- Mais e mais autores tem utilizado o software em suas pesquisas, a **compreensão da literatura** é essencial.
- **Grandes pesquisadores** em estatística já migraram para o R, **observe em sua área**.
- **Contribuição**: Esses mesmo pesquisadores, são aqueles que dispensam seu tempo e esforços para o desenvolvimentos de pacotes específicos para as mais variadas aplicações.

# Por que utilizar o R

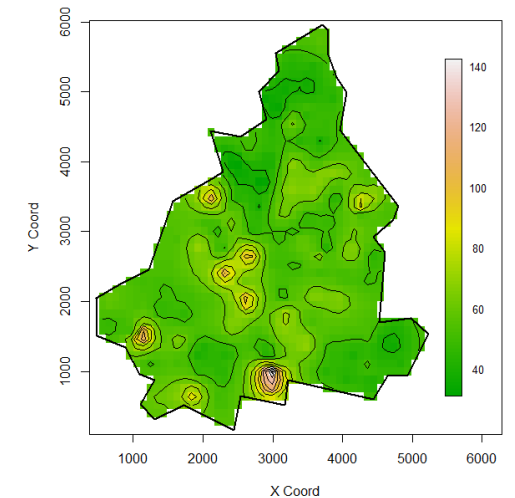
Pb



Semivariograma Pb



Krigagem ordinária Pb

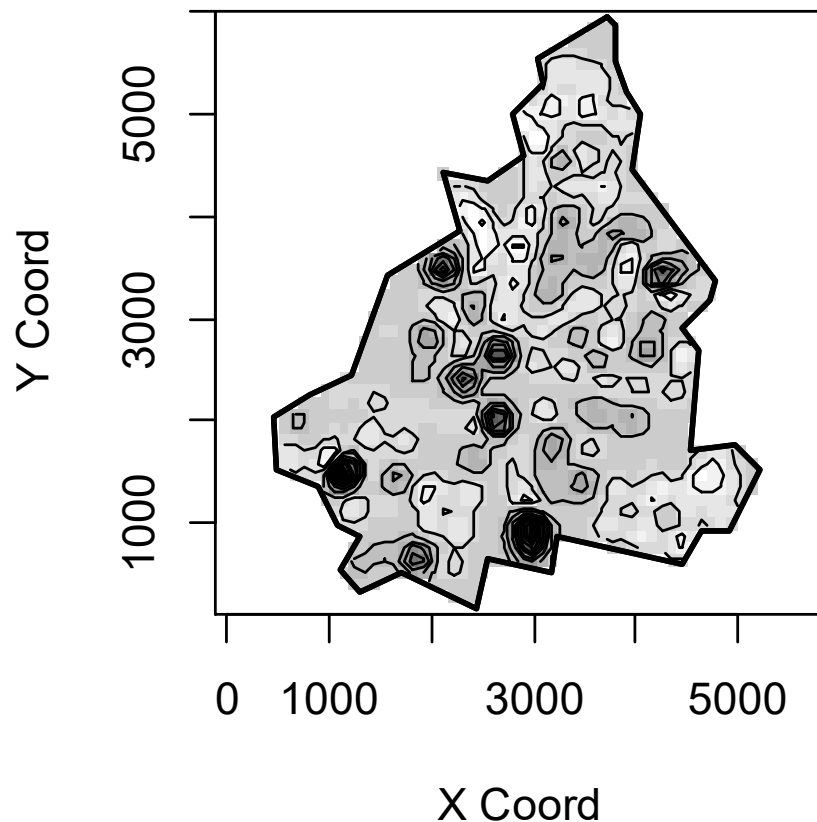


geor – Pacote Para análises geoestatísticas

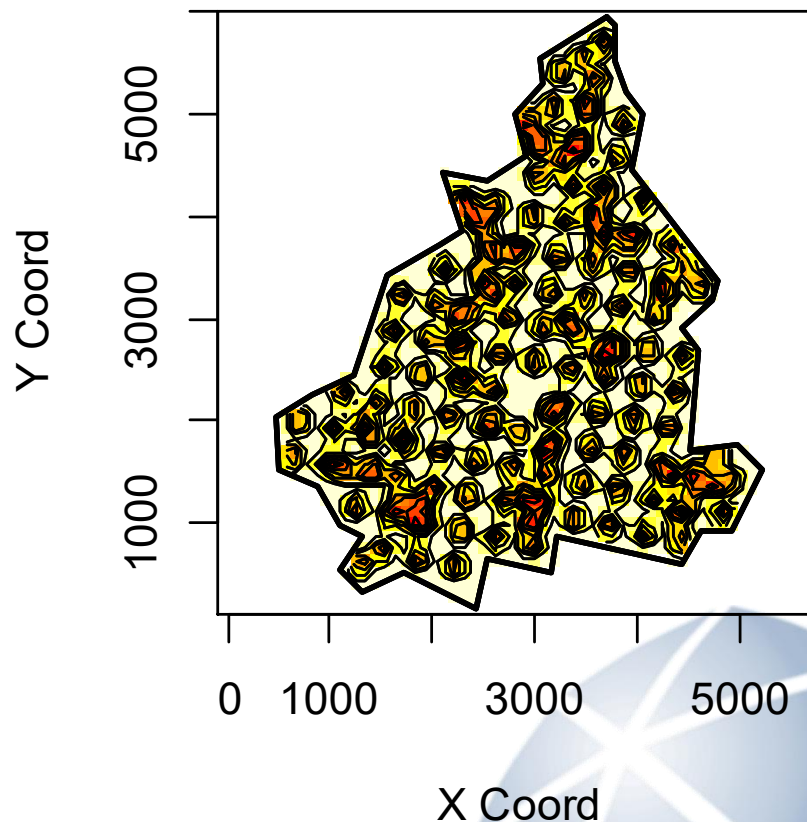


# Por que utilizar o R

## Krigagem ordinária Pb



## Krigagem erro da estimativa Pb



geoR – Pacote Para análises geoestatísticas, desenvolvido pelo Prof. Paulo Justiniano Ribeiro Júnior (UFPR)

## ***Reprodutibilidade***

- Princípio fundamental do método científico.
- A reprodução de resultados em diferentes configurações de softwares e hardwares, torna-se impossível sem a utilização de **programas de código aberto**.
- Políticas de **disponibilização de partes do código** utilizado na pesquisa.
- Que atualmente têm sido adotadas por periódicos como **Biostatistics**, **Science** e **Geoscientific Model Development**.

## ***Liberdade***

O R é distribuído sobre os termos de licença pública GNU.

Existem razões ideológicas para apoiar um programa por uma fundação acadêmica pública, que distribui o seu trabalho grátis para todo o mundo.

O termo código fonte aberto significa que existem vários cientistas trabalhando para a melhoria contínua do programa.

# Instalação

O sistema base e os pacotes são distribuídos via *Comprehensive R Archive Network* (CRAN) acessível sobre

<http://cran.r-project.org>

O analista de dados precisa baixar a distribuição binária pré-compilada e instalá-la localmente. Para os usuários do *Windows* segue o link

<http://cran.r-project.org/bin/windows/base/R-3.1.1-win.exe>

O usuário irá baixar o arquivo correspondente (atualmente chamado **R-3.1.0-win.exe**), execute-o localmente e siga as instruções dada pelo instalador.

Após a instalação procure o ícone do R criado pelo instalador e clique nele



# Instalação

```
R Console
Arquivo  Editar  Misc  Pacotes  Ajuda

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

> █
```

```
> 25+35
[1] 60
```

Poderemos nos deparar com o símbolo ( + ) na parte esquerda do **prompt**, ao invés do ( > ). Isto indica que o sistema está esperando pelo término do comando

```
> 25+(35+53
+
```

Neste caso podemos completar o comando digitando ), e o R retornará o resultado **113**. No caso de erros, pressione a tecla **ESC** para ( > ) retornar na linha do **prompt**.

```
+ )
[1] 113
```

Para sair do R, use a função:  
q() ou a opção de *Menu: arquivo / sair*

# Funções matemáticas (calculadora)

**OBS.: Para adicionar um comentário no R basta colocar o símbolo #**

## **Cálculos apenas com os sinais de operação aritmética:**

```
> 1 + 7 #cálculo da soma destes dois números  
> 1 - 2 * 10 #prioridade da multiplicação  
> 3 ** 7 #potências utilizando ** ou ^  
> 10/3 #divisão
```

## **Cálculos com funções encontradas em calculadoras científicas:**

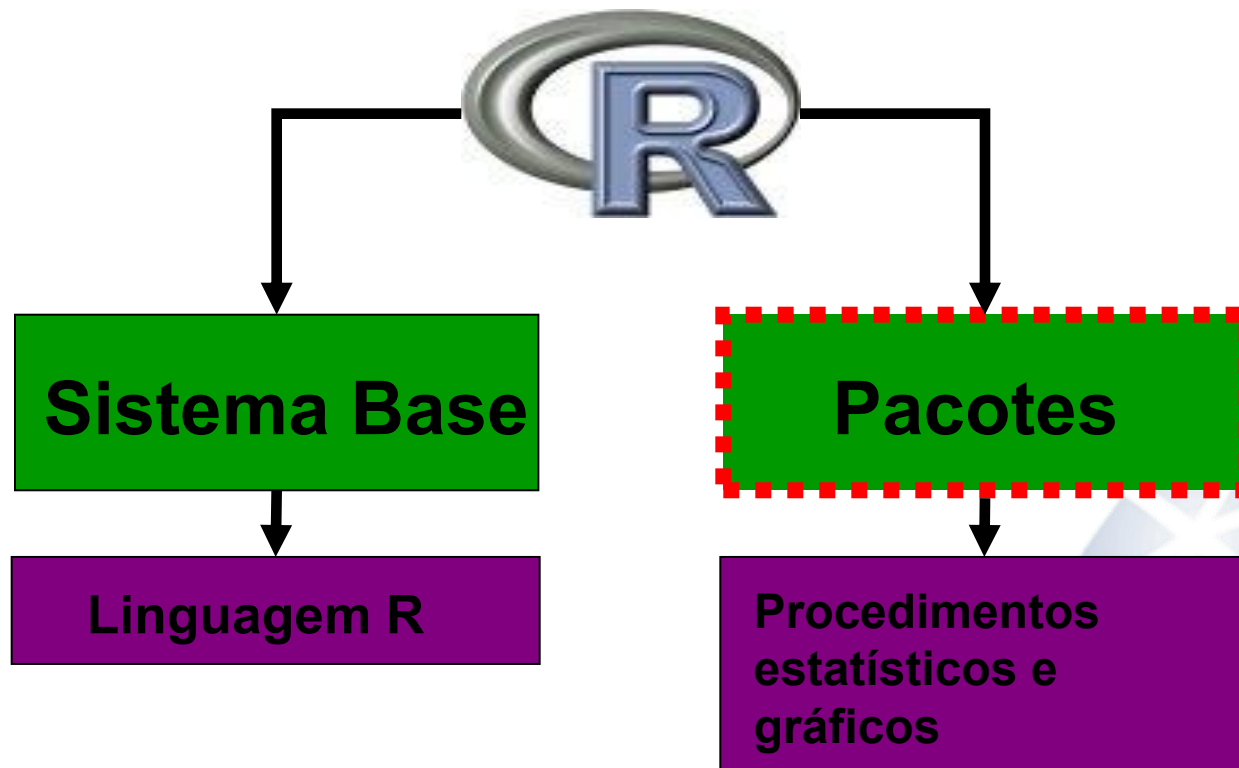
```
> sqrt(9) #raiz quadrada  
> sin(pi) #seno de Pi radianos é zero  
> factorial(4) #4!=4*3*2*1  
> exp(5) #exponencial de 5, e^5
```

## **Outros operadores disponíveis:**

```
%% # módulo  
%/% #divisão inteira
```

# Pacotes

"Um pacote é uma coleção de funções, exemplos e documentação. A funcionalidade de um pacote é frequentemente focada em uma metodologia estatística especial" (Everitt & Hothorn).



# Pacotes

Os procedimentos básicos para a instalação de pacotes são:

## I) Download a partir do *CRAN* e instalação manual

- a) Entrar no site: <http://www.r-project.org>
- b) Selecione a opção *CRAN*
- c) Selecione o *CRAN* do Brasil
- d) No menu da esquerda em *Software*, selecione *Packages*
- e) Selecione a opção *Table of available packages, sorted by name* (ou *Table of available packages, sorted by date of publication*)
- f) Clique no pacote desejado e na próxima página, selecione para o download a versão específica do pacote, de acordo com o seu Sistema Operacional.
- g) No R, clicar em *Pacotes / Instalar Pacote(s) a partir de arquivos zip locais* e selecione o pacote para instalação.



# Pacotes

Os procedimentos básicos para a instalação de pacotes são:

## II) Instalação a partir do *Menu* do R

- a) No R, clicar em ***Pacotes / Instalar Pacote(s)***.
- b) Selecione o ***CRAN*** do Brasil.
- c) Posteriormente aparecerá uma janela com todos os pacotes disponíveis em ordem alfabética.
- d) Selecione o pacote desejado e clique em ***OK***, o R fará o *download* e a instalação automaticamente.

# Pacotes

Os procedimentos básicos para a instalação de pacotes são:

## III) Instalação por meio de linha de comando

Essa é a opção mais simples e rápida para a instalação de pacotes no R, portanto a mais utilizada entre os usuários.

**Sintaxe:** `> install.packages("nome_do_pacote")`

Após a execução do comando é solicitado o *CRAN* onde ele será acessado.

Exemplo:

```
install.packages("agricolae")
```

# Pacotes

```
#listando os pacotes disponíveis no seu computador
.packages(all.available = TRUE)

#listando os pacotes carregados no seu ambiente R
.packages()

#instalando um pacote, especificamente, o "sandwich"
install.packages("sandwich")

#carregando o pacote instalado no seu ambiente R
require("sandwich")

#ou
library("sandwich")

#veja a lista atualizada de pacotes carregados
.packages()
```

Lista de pacotes oficiais:

Na página <http://cran.r-project.org/> clique no *link* "Packages".

```
#ajuda sobre um comando específico do R
help(mean) #o pacote o qual esta função pertença deve estar carregado
#ou simplesmente
?mean

#busca informação do argumento em todos os pacotes instalados
help.search("mean")
#ou simplesmente
??mean

#obter ajuda sobre um particular pacote (instalado)
help(package = survival)

#mostrar exemplos de um comando específico do R
example(mean)
```

# Help on-line

```
#lista as funções e operações contidas no pacote "base"
do R
ls("package:base")
#carregar documento informativo adicional de um pacote
vignette("sandwich") #nem todos os pacotes possuem este
documento
#iniciar ajuda no browser padrão instalado
help.start()
#busca ajuda sobre frases ou palavras-chave na lista de
arquivos ou documentação do R (http://search.r-project.org)
RSiteSearch("statistical multivariate")
#verificar o status da sessão atual do R
sessionInfo()
```

# Help on-line

Para uma documentação eletrônica mais ampla, acesse:

<http://cran.r-project.org/manuals.html>

Para o principiante é indicado a leitura dos 2 primeiros manuais:

***"An Introduction to R" e "R Data Import/Export"***

Uma outra fonte de documentação disponível eletronicamente

<http://cran.r-project.org/doc/Rnews/>

Para acessar a documentação eletrônica contendo as respostas das questões frequentemente perguntadas

Na página <http://cran.r-project.org/> clique no *link* "FAQs".

# Curiosidades

```
#A free book on probability and statistics with R
install.packages("IPSUR")
library(IPSUR)
read(IPSUR)
```

```
#seguir o twitter do R
install.packages("twitterR")
require("twitterR")
Rtweets()
```

```
#R via interface
install.packages("Rcmdr")
require(Rcmdr)
```

# Curiosidades

```
#A free handbook of statistical analysis using R (Everitt & Hothorn  
require(HSAUR)  
vignette()  
vignette("Ch_multiple_linear_regression")  
vignette("Ch_principal_components_analysis")  
browseURL("http://www.r-project.org")  
browseURL("http://www.feis.unesp.br")
```

**Para usuários Windows recomendo a instalação o Tinn-R ou do RStudio:**

**Baixar o executável do Tinn-R ou Rstudio a partir do seguinte link**

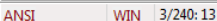
<http://sourceforge.net/projects/tinn-r/>

<http://download1.rstudio.org/RStudio-0.98.1028.exe>

**Seguir a instalação recomendada.**



Tinn-R - [C:\Users\Panosso\Desktop\Exp\_2013\An2013.R] (Not Responding)



# Curiosidades

The screenshot displays the RStudio environment with the following components:

- Script Editor:** Contains R code for data processing and file management.
- Console:** Shows the R version (2.15.3) and workspace information.
- Environment/History:** Displays the current workspace with variables 'P' and 'preco'.
- Files Panel:** Shows the file explorer with a list of files and folders.

**Script Code:**

```
1 #Script para Criação dos arquivos para o pacotezote
2
3 #1-Entrar com os dados
4 setwd("C:/R")#Definição o diretório de trabalho na minha pasta R
5 dados<-read.table("DadosFractal2013.txt",h=TRUE,na.str=".");head(dados)
6
7 #Essa análise indica que existe uma tendência de P em y os semivariograma
8 # cresce indefinidamente, devemos portanto retirar a tendência, trabalhando com os
9 # resíduos
10 # summary(lm(dados$P~dados$x)) #x nao foi significativo
11 pna<-is.na(dados$P)
12
13 dados$P<-ifelse(is.na(dados$P),mean(dados$P,na.rm=T), dados$P) #substituindo NA pela media
14 mod<-lm(dados$P~dados$y) #criando a análise de regressão linear de P vs y
15 dados$P<-as.vector(residuals(mod)) # salvando sobre os valores de P
16 dados$P<-ifelse(pna,NA,dados$P)
17
18 #2-Criação dos DataFrames
19 files<-dir("PacoteIsot_TRD",pattern="^[0-9]*")
20
21 #dir.create("Fractais")
22 length(dados)
```

**Console Output:**

```
R version 2.15.3 (2013-03-01) -- "Security Blanket"
copyright (c) 2013 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-w64-mingw32/x64 (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Workspace loaded from ~/.RData]
```

**Environment/History:**

values	
P	3.3
preco	5.69146

**Files Panel:**

Name	Size	Modified
.RData	80 B	Nov 18, 2013, 11:22 PM
.Rhistry	3.7 KB	Jun 12, 2014, 4:14 PM
Arquivos do Outlook		
Blocos de Anotações do OneNote		
dados0.odt	1.7 KB	Dec 28, 2013, 10:28 PM
dados1.odt	1.7 KB	Dec 29, 2013, 10:56 AM
dados2.odt	1.7 KB	Dec 30, 2013, 9:30 AM
dados3.odt	1.7 KB	Dec 30, 2013, 10:13 AM
dados4.odt	1.7 KB	Dec 30, 2013, 10:17 AM
EndNote		
Enterprise Guide Sample.Ink	1.7 KB	Nov 22, 2010, 10:09 AM
graficoErosividadeAlanMorel.doc	34 KB	Jun 21, 2014, 11:29 AM
GSWin		
IMG_20140528_181057.htm	116.1 KB	May 28, 2014, 10:48 PM
IMG_20140528_181057_arquivos		
IMG_20140_files		
LJ6400Sim		

# Livro Base

"Conhecendo o R. Uma visão estatística."

Autores: Luiz Alexandre Peternelli e Márcio Pupin de Mello

Editora UFV

Publicado em 2007



# Operações com escalares

**#OBS.: A linguagem do R é *case-sensitive***

`x<-5` #o objeto x receberá o valor 5

`5->y` #o valor 5 será armazenado no objeto y

`X=8` #o objeto X receberá o valor 8, não usual

`x=sqrt(100)` #x receberá o valor da operação indicada

`x` #ou

#ao construir funções o comando print deverá ser usado explicitamente

`print(x)`

`ls()` #lista todos os objetos existentes na memória

`lista = ls()` #armazena a lista de objetos

`rm(x,y)` #remove os objetos x e y

`rm(list=ls())` **#remove TUDO do projeto ativo!**  
**CUIDADO!**

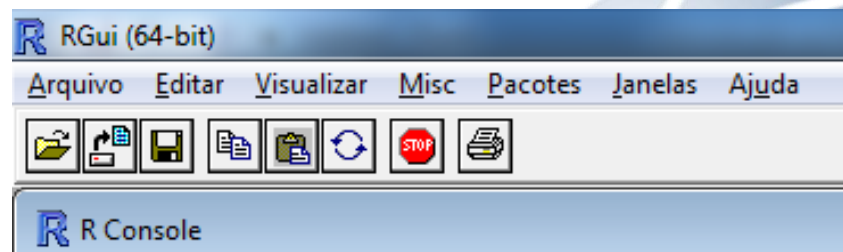
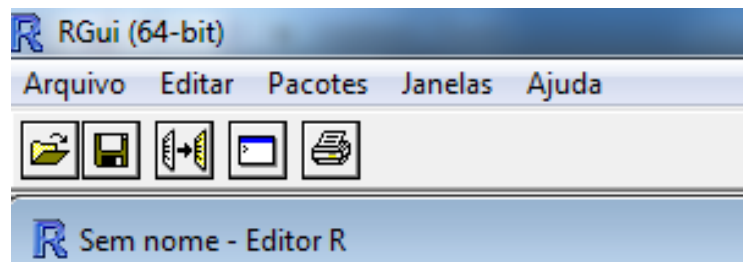
# Iniciando o R

Quando executamos uma análise no R, geralmente, utilizamos várias funções, executando várias linhas de comandos, nesse caso, consideramos útil a utilização de um editor de texto (*script*) para armazenar os diversos comandos, ao invés de executá-los diretamente no *prompt*.

O R possui seu próprio editor, que pode ser acessado pela opção **Arquivo / Novo script**. Será apresentada uma nova janela chamada **Editor R**.

Clique na opção **Janelas** e escolha a melhor forma de apresentação das janelas no programa, por exemplo: **Janelas / Dividir Lado a Lado**.

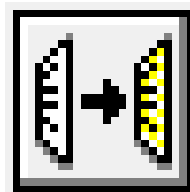
**OBS:** As opções de *Menu* e os *Ícones* apresentados na interface do R serão alterados de acordo com a janela ativa no programa (**Editor R** ou **R Console**).



# Iniciando o R

No Editor R, poderemos digitar e editar os comandos, para executar uma linha do Script, basta deixarmos o cursor nessa linha e pressionar **Ctrl+R** ou **F5**, para executarmos um grupo de linhas basta selecioná-las e pressionar **Ctrl+R** ou **F5**.

Alternativamente a execução de linhas, ou grupos de linhas, também pode ser realizada por meio do ícone abaixo (ou opção *Editar / Executar linha ou seleção*).



# Iniciando o R

## *Símbolos e comandos importantes*

Serão apresentados os principais símbolos e comandos para o início de nossos trabalhos no R.

### *Gerais:*

<i><b>Símbolo</b></i>	<i><b>Significado/ação</b></i>
<b>#</b>	Linha de comentário, não será executado pelo programa.
<b>NA</b>	Dado ausente.
<b>NULL</b>	Nulo, elemento ou componente inexistente.
<b>NaN</b>	Não é um número (exemplo: $0/0$ ).
<b>Inf</b>	Infinito (exemplo: $1/0$ ).
<b>-Inf</b>	Menos infinito (exemplo: $-12/0$ ).
<b>:</b>	criação de sequências (exemplo: $1:50$ ).
<b>~</b>	Representa igualdade na formulação de modelos.
<b>\$</b>	Índice de listas.

# Iniciando o R

## Operadores:

### aritméticos:

- +** Adição.
- Subtração.
- \*** Multiplicação.
- /** Divisão (exemplo: `119/3` R retorna `39.6667`).
- /%** Quociente Inteiro da divisão (exemplo: `119/%3` R retorna `39`).
- %%** Módulo, resto da divisão (exemplo: `119%%3` R retorna `2`).
- ^** Potenciação (exemplo: `2^10` retorna `1024`)
- \*\*** Potenciação (exemplo: `2**10` retorna `1024`).



# Iniciando o R

## Relacionais:

**T** ou **TRUE** Verdadeiro.

**F** ou **FALSE** Falso.

**==** igual a (exemplo:  $225^{0.5} == 15$  ).

**<** menor (exemplo:  $4/8 < 3$ ).

**>** maior (exemplo:  $78/56 > 48/78$ ).

**<=** menor ou igual (exemplo:  $1/8 \leq 0.250$ ).

**>=** maior ou igual (exemplo:  $1/8 \geq 0.125$ ).

**!=** diferente (exemplo:  $0.3 \neq 3/10$ ).

## Lógicos:

**!** Não (!TRUE ou !T)

**&** e (exemplo:  $3 == 30/10 \ \& \ 50/0.1 == 50$ ).

**|** ou (exemplo:  $3 == 30/10 \ | \ 50/0.1 == 50$ ).

# Iniciando o R

## Funções:

No R todas as funções têm a seguinte forma:

```
função(argumento(s)obrigatórios , argumento(s)opcional(is))
```

Sendo que os argumentos opcionais podem apresentar um valor padrão.

Os argumentos estarão sempre entre parênteses, separados por vírgula. Entretanto, algumas função não terão argumentos.

A atribuição de parâmetros aos argumentos será realizada com o sinal de igual "=".

# Iniciando o R

## Algumas funções do R.

### Função

q()  
save.image()  
ls()  
rm(x)  
rm(x,y)  
rm(list=ls(all=TRUE))  
is.na(x)  
sqrt(x)  
log(x,n)  
log(x)  
log10(x)  
exp(x)  
sin(x)  
asin(x)  
abs(x)  
factorial(x)  
floor(x)  
ceiling(x)  
trunc(x)  
round(x, digits=0)  
signif(x, digits=6)  
runif(n)  
c()

### Significado/ação

Sair do programa.  
Salva o trabalho realizado.  
Lista todos os objetos da área de trabalho atual.  
Remove o objeto x.  
Remove os objetos x e y.  
Remove todos os objetos (*R Console – Misc/Remover todos os objetos*).  
Verdadeiro se existir dado(s) ausente(s) no objeto x.  
raiz quadrada de x.  
logaritmo de x na base n.  
logaritmo neperiano de x.  
logaritmo decimal de x.  
antilogaritmo -  $e^x$ .  
seno de x (em radianos).  
arco-seno de x.  
modulo(x).  
 $x!$ .  
maior inteiro  $< x$ .  
menor inteiro  $> x$ .  
inteiro de x, descartando seus decimais.  
arredondando o valor x para um inteiro.  
apresentar 6 dígitos significativos de x.  
gera n números aleatórios entre 0 e 1 a partir de uma distribuição uniforme.  
concatenação, criação de vetores.

# Iniciando o R

```
x = c(1,3,5) #o objeto x recebe a concatenação dos valores numéricos  
1, 3 e 5, ou seja, x é um vetor.
```

```
class(x); mode(x); length(x) #classe, tipo e tamanho do objeto
```

```
is.numeric(x) #x tem atributo numérico?
```

```
# Outros possíveis atributos (tipos). Como, por exemplo:  
vetor, fator, array, matriz, data.frame, ts, lista.
```

```
x=pi
```

```
round(x,3)#arredondando o valor de pi para 3 casas decimais
```

```
trunc(pi)#o valor de pi truncado
```

```
trunc(22.98)
```

```
signif(pi,3)#o valor de pi truncado
```

```
x=0.000256981
```

```
signif(x,3)
```

```
ceiling(pi)# aproxima para o menor inteiro superior ao  
valor especificado (teto)
```

```
floor(pi)# aproxima para o maior inteiro inferior ao  
valor especificado (pisso)
```

# Iniciando o R

```
x = 1:10 #cria a sequência crescente de inteiros de 1 a 10
y = 50:30 #cria a sequência decrescente de inteiros de 50 a 30
#outra forma de criar sequências é através do comando "seq"
seq(1,10,1)# o mesmo que 1:10
seq(1,10,2)
seq(10,1,3) #ordem inversa, produzirá ERRO. Veja!
seq(10,1,-3)
#para criar valores repetidos
rep(1,10)#cria um vetor com o valor 1 repetido 10 vezes
rep(c(1,8),10)
c(rep(0,10),rep(1,5))
a=8
rep(c(1,2),a)#repetição de "c(1,2)" a vezes
rep(4:1,1:4)#exemplo interessante
rep(c(10,20,30),c(3,1,2))#para cada valor, um número de
repetições
```