

R para iniciantes

Aula I

Programação

Carlos Henrique Tonhatti

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Objetivo da disciplina

- Apresentar a linguagem R.
- Introduzir noções básicas de programação.
- Dar autonomia ao aluno para solucionar problemas programando em R.

Estrutura da disciplina

- Aulas presencias no período da manhã.
- Plantões facultativos no período da tarde.
- Um tutorial e um exercício prático para cada aula.
- Tarefas para serem entregues.

Tutoriais, exercícios e tarefas

- Material base <http://ecologia.ib.usp.br/bie5782/doku.php>
- Material dessa disciplina
https://tomatebio.github.io/R_iniciante/

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Programar é preciso

Uma das coisas mais importantes que você pode fazer é dedicar um tempo para aprender uma linguagem de programação de verdade.

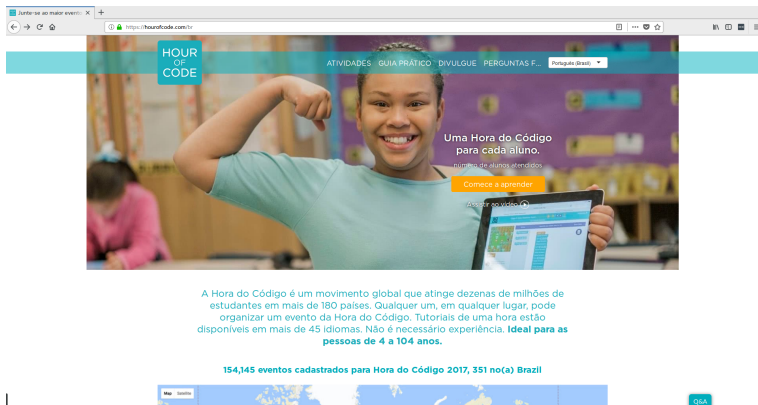
Aprender a programar é como aprender outro idioma: exige tempo e treinamento, e não há resultados práticos imediatos.

Mas se você supera essa primeira subida íngreme da curva de aprendizado, os ganhos como cientista são enormes.

Programar não vai apenas livrar você da camisa de força dos pacotes estatísticos, mas também irá aguçar suas habilidades analíticas e ampliar os horizontes de modelagem ecológica e estatística.

Tradução um tanto livre de Gotelli & Ellison, 2004. A Primer of Ecological Statistics. Sunderland, Sinauer.

Todos podem programar



<https://hourofcode.com/br>

Alguns termos

Programa Sequência de instruções escritas para realizar uma tarefa específica;

Linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador;

Código fonte é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica;

“Máquina” Equipamento eletrônico: computador, servidor, celular, urna eletrônica, etc.

Linguagens de programação

Linguagem (Código) de máquina

Conjunto de instruções em código binário que são compreendidos pela CPU. Dependente do tipo de máquina.

```
8B542408 83FA0077 06B80000 0000C383  
FA027706 B8010000 00C353BB 01000000
```

Linguagem de baixo nível

Linguagem de programação dependente do tipo de máquina de fácil tradução para a máquina. Ex. *assembly*

fib:

```
mov edx, [esp+8]  
cmp edx, 0  
ja @f  
mov eax, 0  
ret
```

Linguagens de programação

Linguagens de alto nível

Linguagem de programação independente do tipo de máquina e de fácil utilização pelo ser humano. Ex. Pascal, C, Algol, Fortran, java, R, Python, Perl, etc

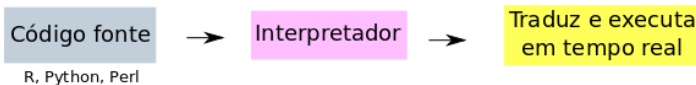
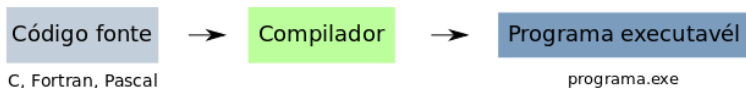
```
unsigned int fib(unsigned int n)
{
    if (n <= 0)
        return 0;
    else if (n <= 2)
        return 1;
    else {
        int a,b,c;
        a = 1;
        b = 1;
        ...
    }
}
```

Tradução das linguagens

- Linguagens de alto ou baixo nível precisam ser traduzidas para código de máquina para que o computador possa executar.
- O modo de tradução depende do tipo de linguagem.
 - Compilação.
 - Interpretação.



Compilação × interpretação



A forma de tradução é uma característica da linguagem.

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Definição

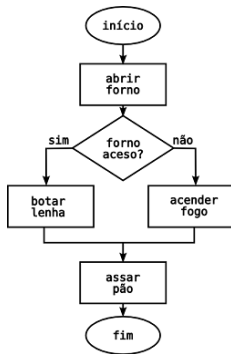
Algoritmo

Uma sequência finita de instruções bem definidas e não ambíguas, e cada uma das quais pode ser executada mecanicamente num período de tempo finito e com uma quantidade de esforço finita.

- Não representa necessariamente um programa de computador;
- Diferentes algoritmos podem realizar a mesma tarefa usando um conjunto diferenciado de instruções em mais ou menos tempo, espaço ou esforço que outros.

Representação visual

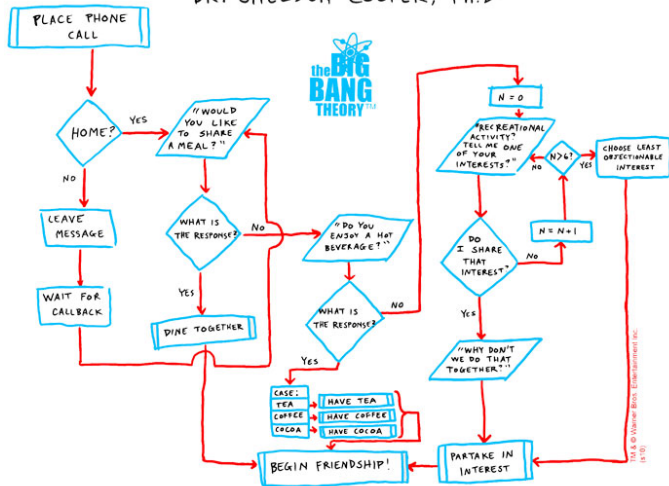
Um algoritmo (ou mesmo um programa inteiro) pode ser representado na forma de um fluxograma.



Exemplo

THE FRIENDSHIP ALGORITHM

DR. SHELTON COOPER, Ph.D



Representação por texto

Um algoritmo (ou mesmo um programa inteiro) pode ser escrita de forma informal na forma de “pseudocódigo”

INÍCIO

 Abrir o Forno

 SE(Forno== acesso)

 Botar lenha

CASO CONTRÁRIO

 Acender fogo

 Assar pão

FIM

Vantagem de não ser dependente de uma linguagem. Pode ser traduzida para qualquer linguagem de programação.

Recomendações

Quando for começar um novo trabalho:

- Escreva todos os passos em “pseudocódigo” ou desenhe um fluxograma. Isso vai ajudar a entender todos os passos,
- Simule os passos no papel. Veja se não ficou nada ambíguo,
- Implemente em uma linguagem de programação por partes e teste cada parte.
- Use comentários!

Para ver como se programa em outras linguagens

http://rosettacode.org/wiki/Rosetta_Code

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R**
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - **Características**
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

O que é R

- Ambiente de programação: coerente e integrado.
- Conjunto de ferramentas para manipulação de dados, cálculos e apresentação gráfica.
- Linguagem S (Bell Labs).

Características do R

- Código aberto: projeto colaborativo.
- Flexibilidade.
- Disponível para Windows, Mac e Linux.
- É orientado a objetos.
- Interface com outras linguagens.

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> □
```


Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R**
 - Características
 - Estrutura do R**
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Avaliador de expressões

O R avalia as expressões digitadas:

Ler → *Analisar* (*sintaxe*) → *Avaliar*

O R lê o que foi escrito pelo usuário, analisa o texto como uma expressão e tenta avaliar esta expressão. Normalmente retorna uma resposta na tela.

Exemplos

```
> 12,3
```

```
Erro:  ', ' inesperado in "12,"
```

```
> oi
```

```
Erro:  objeto 'oi' não encontrado
```

```
> 1
```

```
[1] 1
```

```
> a=1
```

```
>
```

Sintaxe básica

Sintaxe básica de atribuição

Mais usado.

objeto <- expressão

Não significa igualdade.

objeto = expressão

Pouco usado.

expressão ->objeto

```
> a<-4
```

```
> a=4
```

```
> 4->a
```

Digite o nome para exibir.

```
> a
```

```
[1] 4
```

Sintaxe básica

Sintaxe básica das funções

`função(argumento1=valor, argumento2= valor,...)`

```
> plot(x=area, y=peso)
> plot(y=peso, x=area)
> plot(area, peso)
```

É diferente de:

```
> plot(peso, area)
```

O R interpreta os argumentos pelo nome ou pela posição dentro da expressão. Não é preciso colocar o nome do argumento se a posição for mantida.

Não esqueça dos parênteses

Mesmo sem argumentos é necessário digitar os parênteses.

```
> citation()
```

To cite R in publications use:

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

```
> citation
function (package = "base", lib.loc = NULL, auto = NULL)
{
  if (inherits(auto, "packageDescription")) {
    :
  }
}
```

Sabendo os argumentos de uma função

Retorna os argumentos de uma função

`args(Nome da função)`

```
> args(ls)
function (name, pos = -1L, envir = as.environment(pos),
all.names = FALSE, pattern)
NULL
```

Cartão de referência para imprimir que contém as funções mais usadas.
Disponível em

<http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

Ajuda sobre funções

Felizmente existem várias formas de buscar ajuda sobre as funções.

Funções de ajuda

Abre texto de ajuda.

`help(Nome da função)`

`?Nome da função`

Abre a ajuda no navegador.

`help.start()`

Busca dentro dos arquivos de ajuda.

`help.search("palavra-chave")`

`? ? "palavra-chave"`

`apropos("palavra-chave")`

Ajuda online

Google Não funciona muito bem, “R” é uma palavra fácil de confundir. Sugestão:

```
filetype:R “palavra-chave” -rebol
```

Rseek Buscador específico para R. Funciona muito bem.

www.rseek.org

Site do R Os manuais oficiais do R, material produzido por usuários, listas de email.

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

Outros endereços

R-bloggers Notícias, Tutoriais enviados por usuários.

<http://www.r-bloggers.com>

Quick-R Tutoriais e exemplos curtos.

<http://www.statmethods.net/>

Apostila online Apostila usada no curso da Ecologia da USP.

<http://ecologia.ib.usp.br/bie5782>

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R**
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos**
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Linguagem Orientada a Objetos

Objeto

*It is a location in memory having a **value** and referenced by an **identifier**. An object can be a variable, function, or data structure.*

[http://en.wikipedia.org/wiki/Object_\(computer_science\)](http://en.wikipedia.org/wiki/Object_(computer_science))

```
> a<-42
```

Objetos pertencem a classes

Classe

*It is an extensible **template for creating objects**, providing initial values for state (member variables) and implementations of **behavior** (member functions, methods)*

[http://en.wikipedia.org/wiki/Class_\(computer_science\)](http://en.wikipedia.org/wiki/Class_(computer_science))

Principais classes de objetos no R

- Vetores.
- Fatores.
- Matriz.
- Lista.
- *Data frame*.

Vetor

Vector

É um conjunto de elementos de uma mesma classe (números, caracteres, lógicos, etc) organizados em uma dimensão. Não corresponde aos vetores da álgebra linear.

```
a<-4  
> Animal<-c("Cachorro", "Gato", "Mosquito")  
> Peso<-c(1500,1000,1)  
> Pet<-c(TRUE,TRUE,FALSE)
```

Fatores

Factor

Semelhante aos vetores de R. Os elementos são divididos em níveis *levels*. Usado para dados categóricos.

```
> doce <-factor(c("light","light","diet","diet"))
> doce
[1] light light diet diet
Levels:  diet light
```


Fatores

Níveis de um fator

Retorna os níveis de um fator.

`levels(Nome do fator)`

```
> levels(doce)
[1] "diet" "light"
```

Adicionando um nível que não há na amostra

```
> levels(doce) <- c("diet", "light", "açúcar")
> doce
[1] light light diet diet
Levels:  diet light açúcar
```

Matriz

Matrix

É um vetor com dois atributos adicionais: Número de linhas e colunas.
Corresponde a matriz da Álgebra linear.

$$A_{2,3} = \begin{pmatrix} 1 & 4 & 5 \\ 3 & 6 & 7 \end{pmatrix}$$

```
> A <-matrix(c(1,4,5,3,6,7),nrow=2,ncol=3)
> A <-matrix(c(1,4,5,3,6,7),nrow=2)
> A
```

	[,1]	[,2]	[,3]
[1,]	1	5	6
[2,]	4	3	7

Lista

List

É uma coleção de objetos de diferentes classes. Pode conter outras listas.

```
> lista.qualquer<-list(doce,A)
```

```
> lista.qualquer
```

```
[[1]]
```

```
[1] light light diet diet
```

```
Levels:  diet light açúcar
```

```
[[2]]
```

```
      [,1] [,2] [,3]
```

```
[1,]    1    5    6
```

```
[2,]    4    3    7
```

Data frame

Data frame

É um tipo especial de lista no qual os componentes tem o mesmo tamanho. Semelhante às planilhas de dados do excel.

```
> Animal<-c("Cachorro", "Gato", "Mosquito")
> Peso <-c(1500,1000,1)
> Pet <-c(TRUE,TRUE,FALSE)
> Bichos <-data.frame(Animal,Peso,Pet)
> Bichos
```

	Animal	Peso	Pet
1	Cahorro	1500	TRUE
2	Gato	1000	TRUE
3	Mosquito	1	FALSE

Sobre as classes

Identificando a classe de um objeto

Retorna qual a classe de um objeto.

`class(Nome do objeto)`

Retorna Verdadeiro ou Falso para uma determinada classe.

`is.vector(Nome do objeto)`

`is.factor(Nome do objeto)`

`:`

Retorna a estrutura de um objeto.

`str(Nome do objeto)`

```
> class(Bichos)
[1] "data.frame"
> is.vector(Bichos)
[1] FALSE
> is.data.frame(Bichos)
[1] TRUE

> str(Bichos)
'data.frame':  3 obs.  of 3 variables:
 $ Animal:  Factor w/ 3 levels "Cahorro","Gato",...:  1 2 3
 $ Peso  :  num 1500 1000 1
 $ Pet   :  logi TRUE TRUE FALSE
```

Listar e Remover

Listar e remover

Lista todos os objetos criados.

`ls()`

Remove um objeto. Sem recuperação.

`rm(Nome do Objeto)`

```
> ls()
```

```
> rm(Bichos)
```

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R**
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes**
- 4 Sessão de trabalho no R
- 5 Recomendações

Pacotes

Todas as funções do R são organizadas em pacotes. O pacote base é instalado junto com o R e traz as funções mais básicas.

Pacotes adicionais estão disponíveis na Internet. Cada pacote adiciona novas funções e métodos e, normalmente são feitos para uma tarefa específica.

Achando um pacote

Há mais de 12000 pacotes no site oficial do R.
<https://cloud.r-project.org/>

Índice alfabético

Available CRAN Packages By Name

[A](#)[B](#)[C](#)[D](#)[E](#)[F](#)[G](#)[H](#)[I](#)[J](#)[K](#)[L](#)[M](#)[N](#)[O](#)[P](#)[Q](#)[R](#)[S](#)[T](#)[U](#)[V](#)[W](#)[X](#)[Y](#)[Z](#)

[A3](#)
[abc](#)
[abcdeFBA](#)
[ABCExtremes](#)
[ABCoptim](#)
A3: Accurate, Adaptable, and Accessible Error Metrics for
Tools for Approximate Bayesian Computation (ABC)
ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balan
ABC Extremes
Implementation of Artificial Bee Colony (ABC) Optimizatio

Organizados por tarefa

CRAN Task Views

[Bayesian](#)
[ChemPhys](#)
[ClinicalTrials](#)
[Cluster](#)
[DifferentialEquations](#)
[Distributions](#)
[Econometrics](#)
Bayesian Inference
Chemometrics and Computational Physics
Clinical Trial Design, Monitoring, and Analysis
Cluster Analysis & Finite Mixture Models
Differential Equations
Probability Distributions
Computational Econometrics

Instalando um novo pacote

Instalar um novo pacote

```
install.packages("Nome do pacote")
```

É necessário carregar o pacote antes de usar as funções.

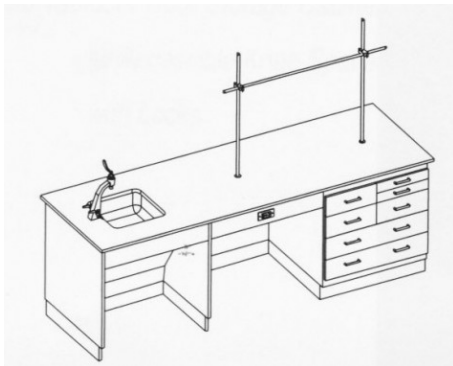
Carregando um pacote

```
library(Nome do pacote)  
require(Nome do pacote)
```

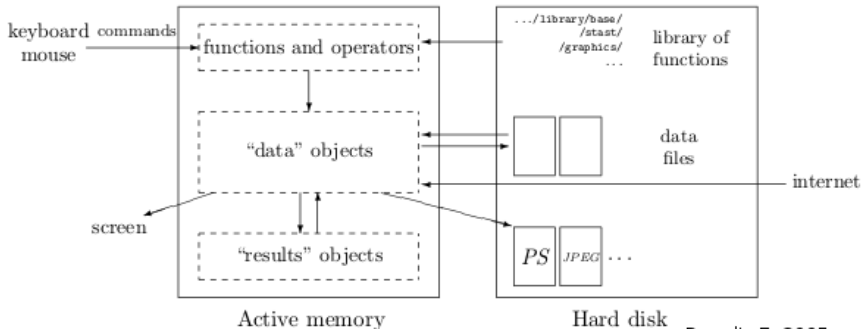
Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Área de trabalho



Workspace



Paradis E. 2005

Iniciar uma sessão de trabalho

Existem algumas formas diferentes de iniciar uma sessão de trabalho no R:

- Clicando no ícone na Área de trabalho (Win/Mac).
- Digitando “R” no terminal (Linux/Mac).
- Abrindo arquivos com extensões associadas ao R (.r, .RData, etc).

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> 
```

Encerrando uma sessão de trabalho

Encerra uma sessão

```
q()
```

```
> q()
```

```
Save workspace image? [y/n/c]:
```

Ao encerrar uma sessão o R pergunta se você pretende salvar a área de trabalho. Se responder “y” o R salva uma imagem de toda a área de trabalho num arquivo oculto *.RData* dentro do diretório de trabalho.

Trabalhando com a imagem da área de trabalho

Salvando a área de trabalho

Salva a área de trabalho no arquivo oculto *.Rdata* sem encerrar a sessão.

```
save.image()
```

Salva a área de trabalho com o nome que você escolher sem encerrar a sessão.

```
save.image(file="meuworkspace.RData")
```

Carrega um arquivo de área de trabalho.

```
load(file="meuworkspace.RData")
```

É recomendável salvar o trabalho regularmente.

Diretório de trabalho

Os arquivos *.RData* são salvos no diretório de trabalho do R. Isso pode gerar problemas pois as novas análises podem sobrescrever as antigas. Por isso:

- Crie um diretório diferente para cada projeto.
- Coloque todos os seus arquivos de dados, planilhas etc nesse diretório.
- Direcione o R para usar o diretório que você criou como diretório de trabalho.

Diretório de trabalho

Diretório de trabalho

Retorna qual diretório de trabalho atual.

`getwd()`

Altera o diretório de trabalho.

`setwd()`

Sumário

- 1 Apresentação da disciplina
- 2 Programação
 - Algoritmos
- 3 Linguagem R
 - Características
 - Estrutura do R
 - Linguagem Orientada a Objetos
 - Pacotes
- 4 Sessão de trabalho no R
- 5 Recomendações

Recomendações para trabalhar com o R

- Use um diretório para cada projeto.
- Salve a área de trabalho regularmente, `save.image()`.
- Escreva seus comandos em um arquivo de texto.
- Escreva de forma organizada.
- Salve este arquivo no diretório de trabalho com extensão `.r` ou `.R`.
- Envie os comandos ao R:
 - Usando o comando `source("arquivo.r")`. Executa todos os comandos de uma vez.
 - Na interface R-GUI do Windows, use o editor de scripts e `ctrl+R` para enviar linhas ou blocos.
 - **Use editores próprios para programação em R.**

Editores próprios para R

Facilitam o trabalho de programação no R. Combinam ferramentas para escrever executar o código de modo mais eficiente. Experimente alguns até achar um que agrade.

Tinn-R Editor de códigos brasileiro.

`http:`

`//nbcgib.uesc.br/lec/software/editores/tinn-r/pt`

ESS *Emacs Speaks Statistics* Extensão do Emacs para R.

`http://ess.r-project.org/`

R studio É uma das melhores opções para começar.

`http://www.rstudio.com/`

Para a próxima aula

- Realizar “Uma hora de código”
- Instalar o R e o R-studio.
- Ler capítulo 1 e 2 da apostila.
- Fazer os tutoriais “Introdução ao R” que estão em
`http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:02_tutoriais:start`