

# BIE5782

Aula 1 :

APRESENTAÇÃO DA DISCIPLINA  
E DA LINGUAGEM R

# ROTEIRO

1. Motivações para aprender R, filosofia do R
2. Estrutura e funcionamento da disciplina
3. Estrutura básica do R:
  - Sintaxe dos comandos
  - Obtendo ajuda
  - Orientação a objeto
  - Organizando e gravando seu trabalho no R
  - Manipulações básicas de objetos
  - Criação de objetos simples e suas classes

# O que é o R?

- Um ambiente de programação
- Ferramentas para manipulação de dados, cálculos e gráficos
- Linguagem S: flexibilidade e alto padrão
- Código aberto: projeto colaborativo
- Voltado para **desenvolvimento e implementação pelo usuário**

# Um ambiente pouco amigável ...

```
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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.

[Área de trabalho anterior carregada]

> █
```

# .. ou apenas pouco familiar ?

The image displays a collage of SPSS-related software windows. At the top, the 'SYSTAT Output Organizer' window shows a file tree with 'SYSTAT Output' and 'File C:\DOCL' containing a 'Plot TEST1'. Below it, the 'SYSTAT Graph' window displays a scatter plot of 'TEST2' with a y-axis ranging from 80 to 100. The plot shows several data points. To the right of the graph are controls for '3-D Rotation' and 'X-Power' and 'Y-Power', both set to 1.000.

In the bottom left, the 'SPSS Data Editor' window shows a data table with columns 'Name', 'Type', 'Width', 'Decimals', 'Label', 'Values', 'Missing', 'Columns', and 'Align'. The 'Variable View' tab is active, showing a list of variables: 'status', 'q1a', 'q1b', 'q1c', 'q1d', 'q1e', and 'q1f'. A 'Syntax Editor' window is overlaid on the data editor, containing the following code:

```
end if.  
loop.  
select if status=1.  
freq /var=q1a.
```

Below the data editor, a 'STATISTICA Basic' window is visible, showing a flowchart of the software's architecture. The flowchart includes sections for 'INPUT:', 'STATISTICA BASIC:', and 'OUTPUT:'. The 'STATISTICA BASIC:' section contains a list of commands and their corresponding outputs:

- DATAFILES: DATAFILES
- SCROLLSHEETS: SCROLLSHEETS
- GRAPHS: GRAPHS
- DISK FILES: DISK FILES
- INTERACTIVE INPUT: INTERACTIVE INPUT

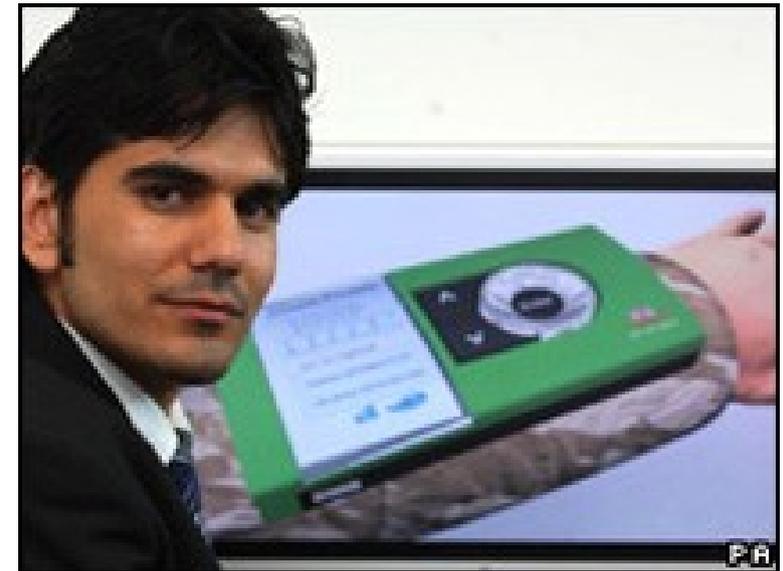
The 'OUTPUT:' section includes: DATA FILES, SCROLLSHEETS, GRAPHS, DISK FILES, and TEXT & GRAPHICS REPORTS. A 'PRINTER' icon is also present, indicating that reports can be printed.

# A Camisa de Força das Interfaces

*O aparelho deve traduzir frases como "não atire", "não se aproxime" e "a ajuda vai chegar logo".*

*A idéia foi de Amin Ismail, um ex-estudante iraquiano que fugiu de seu país em 2002 e vive na Grã-Bretanha.*

Ismail disse que o projeto é "uma maneira de agradecer aos soldados britânicos por seu trabalho honroso"



# Golden Hammer



Abraham Maslow

*"It is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail."*

[http://en.wikipedia.org/wiki/Law\\_of\\_the\\_instrument](http://en.wikipedia.org/wiki/Law_of_the_instrument)

**Martelinho de Ouro**

ESPECIALIZADO EM RECUPERAÇÃO DE VEÍCULOS DANIFICADOS POR CHUVA DE GRANIZO

9954-0021 / 9263-4005 JUNIOR



# Programar é preciso!

*“to turn raw data into published research papers often requires a little programming, which means that most scientists write software.”*

Nick Barnes, Nature News 2010

Published online 13 October 2010 | *Nature* 467, 753 (2010) | doi:10.1038/467753a

Column: World View

## Publish your computer code: it is good enough

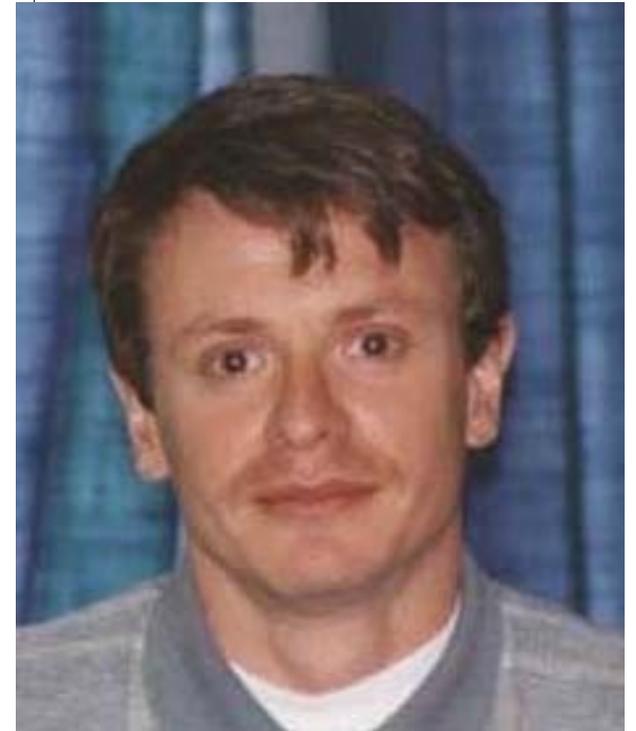


Freely provided working code — whatever its quality — improves programming and enables others to engage with your research, says Nick Barnes.

Nick Barnes

# Flexibilidade

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

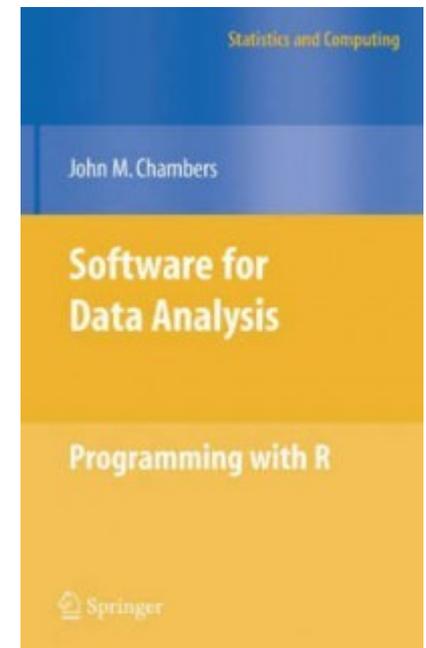


Paulo Justiniano Ribeiro

# Filosofia do R: Programar com dados

*“programming ought to be regarded as an integral part of effective and responsible data analysis”*

John M. Chambers, Programming with R, 2008



# Controle

" One of the objectives of statistical analysis is to distil a long and complicated set of data into a small number of meaningful descriptive statistics."

" Many of the modern computer statistical packages, however, do exactly the opposite of this." [Their] "copious output has several major shortcomings: it is open to uncritical acceptance; it can lead to over interpretation of data; and it encourages the bad habit of data trawling."

**"R, by other hand, tells you nothing unless you explicitly ask for it."**

Crawley, M. The R Book.

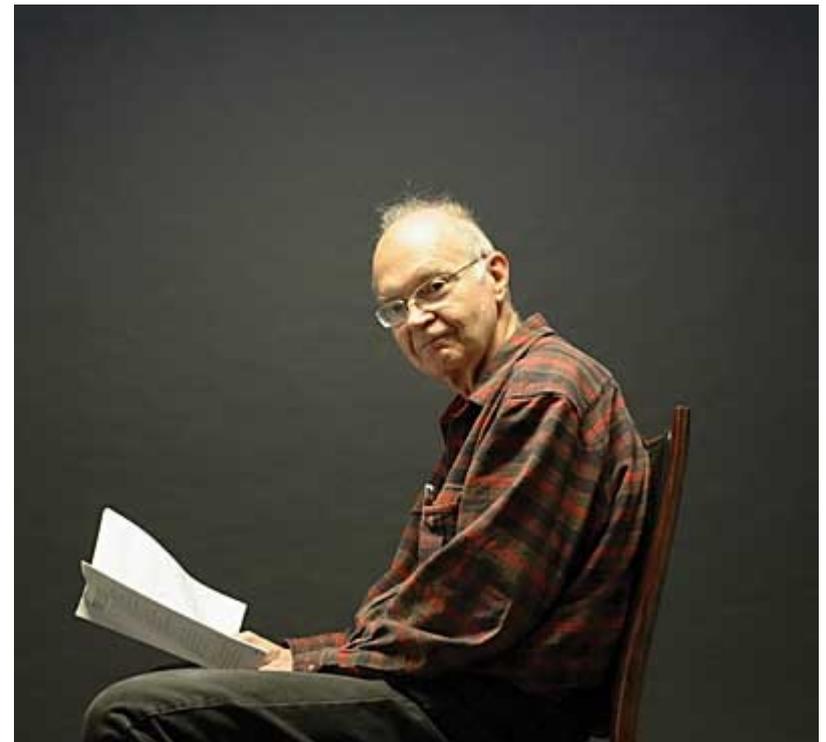


Mick Crawley

# Reproducible research ...

*” Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.”*

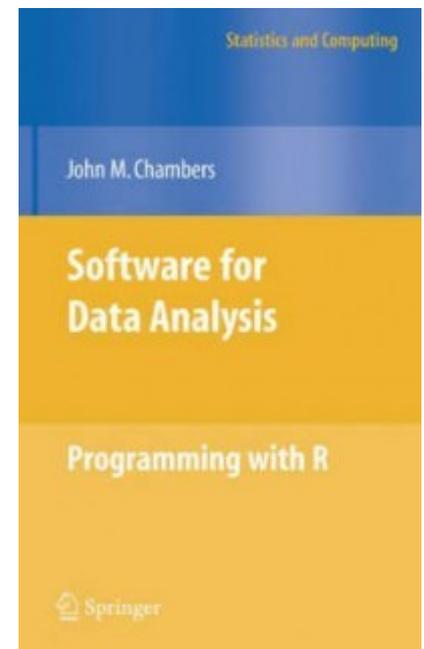
*Donald Knuth – “Literate Programming” The Computer Journal, 1984.*



... é programar com dados ...

*“to enable the best and most thorough exploration of data possible, [...] but always explaining the nature of the method applied, in an open and understandable format, supported by as much evidence of its quality as can be produced.”*

John M. Chambers, Programming with R, 2008



# E só é possível com código aberto



## **COPYLEFT:**

Código disponível para ser avaliado, modificado e redistribuído.

1. Transparência
2. Flexibilidade
3. Colaboração

# Por que uma disciplina de R?

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

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

**"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.

E dizemos mais:

**Programar com seus dados  
muda sua maneira de  
entender e fazer pesquisa!**

# Mas não se iluda:

"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."

# Uso da Linguagem R em Ciências Biológicas e Ambientais



Alexandre Adalardo de Oliveira (IB-USP)  
João Batista Ferreira ( ESALQ-USP)  
Rodrigo Santinello Pereira (FFCLRP-USP)  
Paulo Inácio Prado (IB-USP)

& MUITOS monitores!

# BIE5782

## Estratégia de Ensino

- Manhãs: revisão dos exercícios e tutoriais e aula teóricas introdutórias
- Tardes: tutoriais com a presença da equipe
- No dia seguinte: estudo (individual ou grupo):
  - Leia a apostila
  - Conclua os tutoriais
  - Faça o exercícios
  - Busque e dê auxílio no fórum

# **BIE5782**

## Avaliações

- Exercícios:
  - Postar no wiki até a aula seguinte.
- Trabalho final:
  - Criar uma função em R e sua página de ajuda.
  - Postar proposta no wiki até 28/3
  - Postar trabalho no wiki até 23/04

# BIE5782

## Wiki

Apresentação [Uso da Linguagem R para Análises de dados Ecológicos] - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

Apresentação [Uso da Lingua... X] N BIE5782 - Alunos 2012 | Page 2 X +

ecologia.ib.usp.br/bie5782/doku.php

Mais visitados PG-Eco LET IB-Webmail USP Sistemas USP Agenda disciplinas Google SibiNet R Project LAGE FB RBZ Fluxbox-wiki CET - Trânsito Agora

### [[Apresentação]]

## USO DA LINGUAGEM R PARA ANÁLISES DE DADOS ECOLÓGICOS

Visitou: • Página dos alunos e trabalho final • Apresentação

Mostrar código fonte Revisões anteriores Alterações recentes Índice Entrar

Pesquisar

#### Índice



#### Curso 2012

- Introdução
- Curso 2012
- Atividades Preparatórias

## Apresentação

Tabela de conteúdos

"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

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

# BIE5782

## Fórum

BIE5782 - Mozilla Firefox

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

Apresentação [Uso da Língua...] N BIE5782

bie5782.138098.n3.nabble.com

open source

Mais visitados PG-Eco LET IB-Webmail Sistemas USP Agenda disciplinas Google SibiNet R Project LAGE FB RBZ Fluxbox-wiki CET - Trânsito Agora

### BIE5782

USO DA LINGUAGEM R PARA ANÁLISE DE DADOS ECOLÓGICOS

Fórum de discussão da disciplina [BIE5782](#), do [Curso de Pós-Graduação em Ecologia](#) da [Universidade de São Paulo](#)

**INSTRUÇÕES:**

- Antes de criar um novo tópico, verifique se não existe nenhum similar.
- Ao criar um tópico, dê a ele um título informativo.
- Sempre que o tópico for sobre problemas ou dúvidas sobre exercícios, tutoriais ou uso de funções, coloque na sua mensagem seu script, e se possível o resultado, mesmo que seja uma mensagem de erro.

New Topic People Options

1 2 3 4 ... 8

Topics (265)	Replies	Last Post	Views
<a href="#">Alunos 2012</a> by Paulo Prado	☆ 52	<a href="#">Mar 18</a> by Izadora	773
<a href="#">No create page button for own wiki</a> by Michel	☆ 2	<a href="#">Mar 16</a> by adalardo	47
<a href="#">Login problems on my own wiki page</a> by Michel	☆ 7	<a href="#">Mar 15</a> by adalardo	66
<a href="#">Dúvida cronograma</a> by Fernanda	☆ 1	<a href="#">Mar 12</a> by Paulo Prado	62
<a href="#">Aninhamento NODF</a> by Roberto Schmidt	☆ 0	<a href="#">Nov 20, 2011</a> by Roberto Schmidt	24
<a href="#">Problema no Encoding [ Urgente ]</a> by Rzhou	☆ 1	<a href="#">Nov 15, 2011</a> by adalardo	14
<a href="#">Criando Graficos de Indices</a> by Rodrigo	☆ 0	<a href="#">Nov 04, 2011</a> by Rodrigo	11

<http://bie5782.138098.n3.nabble.com/>

# Estrutura Básica da Linguagem R

```
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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.

[Área de trabalho anterior carregada]

> |
```



# Sintaxe Básica das Funções

Isto é uma EXPRESSÃO



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

```
> plot(x=area, y=riqueza, log="xy")
```

```
> plot(area, riqueza)
```

```
> plot(area, riqueza, log="xy")
```

# help , help.start e args

## Ajuda sobre funções

### Texto de Ajuda

```
> help(mean)  
> ?mean
```

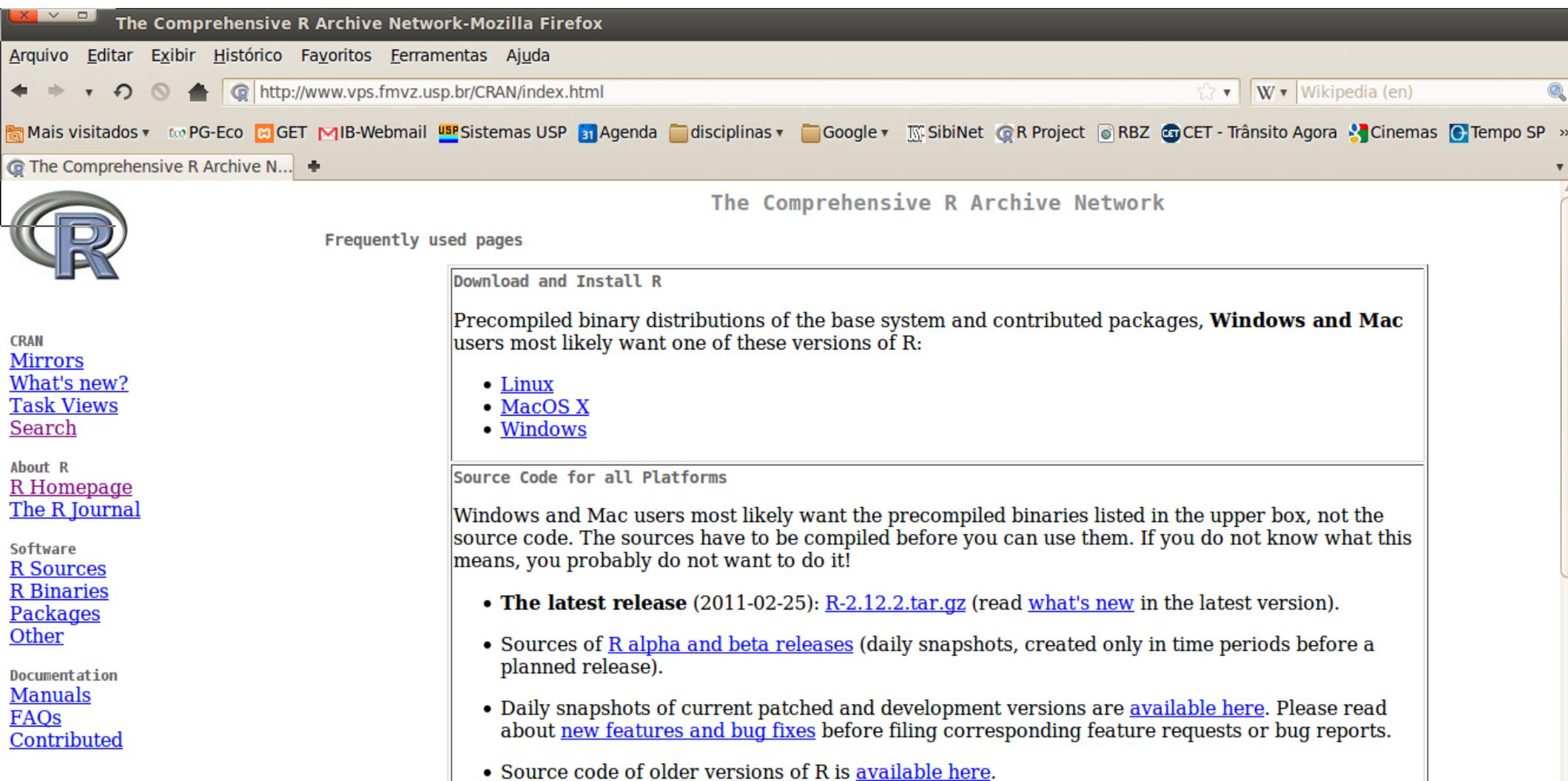
### Hipertexto de Ajuda

```
> help.start()
```

### Argumentos de uma função

```
> args(lm)  
function (formula, data, subset, weights, na.action,  
method = "qr", model = TRUE, x = FALSE, y = FALSE,  
qr = TRUE, singular.ok = TRUE, contrasts = NULL,  
offset, ...)
```

# De onde vem as funções?



The Comprehensive R Archive Network

Frequently used pages



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:

- [Linux](#)
- [MacOS X](#)
- [Windows](#)

### Source Code for all Platforms

Windows and Mac users most likely want 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** (2011-02-25): [R-2.12.2.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](#).

# A metáfora da marcenaria



# install.packages

pacotes no CRAN

```
> install.packages("vegan")
Installing package(s) into
'/home/paulo/R/x86_64-pc-linux-gnu-
library/2.12'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use
in this session ---

Loading Tcl/Tk interface ... done
tentando a URL
'http://www.vps.fmvz.usp.br/CRAN/src/contrib/vegan_1.1
7-8.tar.gz'
Content type 'application/x-gzip' length 1438458 bytes
(1.4 Mb)
URL aberta
=====
downloaded 1.4 Mb

* installing *source* package 'vegan' ...
```



# **require , library**

Carrega pacotes instalados

## **Um pacote da distribuição padrão**

```
> require(MASS)
```

```
Carregando pacotes exigidos: MASS
```

```
> library(MASS)
```

```
Carregando pacotes exigidos: MASS
```

## **Pacote adicionado com `install.packages`**

```
> require(vegan)
```

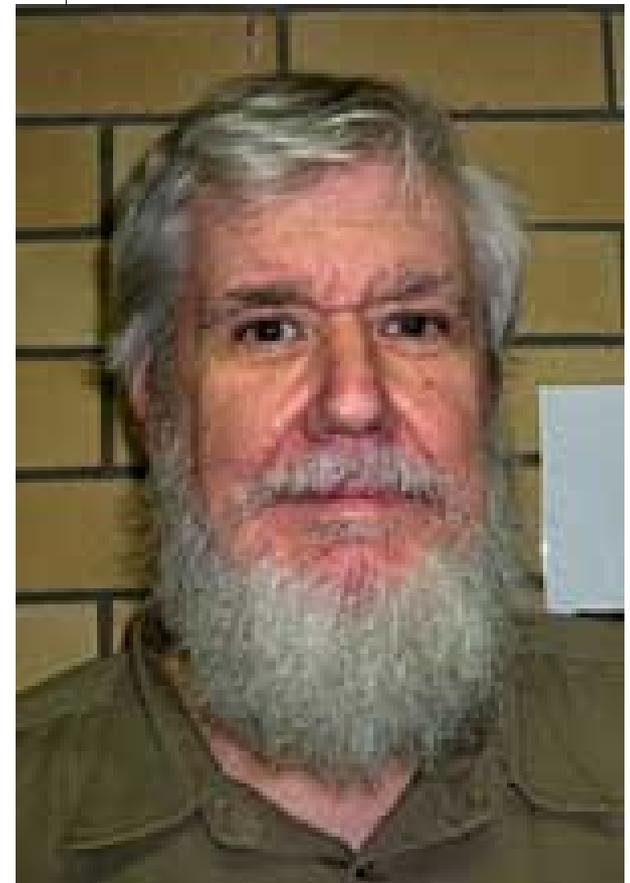
```
Carregando pacotes exigidos: vegan
```

```
This is vegan 1.17-8
```



A chave para entender o R é que trata-se de uma linguagem.

**Uma linguagem para manipular objetos.**



Bill Venables

# Operadores `<-`, `=`, `->`

## Objetos e Atribuição

**objeto** `<-` expressão

```
> area <- c(303, 379, 961, 295, 332, 47, 122, 11, 53, 2749)
```

**objeto** `=` expressão

```
> area = c(303, 379, 961, 295, 332, 47, 122, 11, 53, 2749)
```

**Pouco Usado:**

**expressão** `->` objeto

```
> c(303, 379, 961, 295, 332, 47, 122, 11, 53, 2749) -> area
```

# Digite o nome do objeto para exibí-lo

```
> area
```

```
[1] 303 379 961 295 332 47 122 11 53 2749
```

```
> riqueza
```

```
[1] 3 10 20 7 8 4 8 3 5 23
```

# Um erro comum

```
> help
function (topic, offline = FALSE, package = NULL,
lib.loc = NULL,
    verbose = getOption("verbose"), try.all.packages =
getOption("help.try.all.packages"),
    chmhelp = getOption("chmhelp"), htmlhelp =
getOption("htmlhelp"),
    pager = getOption("pager"))
{
  if (!missing(package))
    if (is.name(y <- substitute(package)))
      package <- as.character(y)
```



Funções são objetos, como tudo mais no R. Se você digita uma função sem os parênteses, o R entende que vc quer exibir seu conteúdo, no caso seu código.

# Linguagem Orientada a Objetos

```
> area <- c(303, 379, 961, 295, 332, 47, 122, 11, 53, 2749)
```

```
> area
```

```
[1] 303 379 961 295 332 47 122 11 53 2749
```

```
> summary(area)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
11.00	70.25	299.00	525.20	367.20	2749.00

# Linguagem Orientada a Objetos

```
> modelo1 <- lm(riqueza~area)
> summary(modelo1)
```

```
Call:
lm(formula = riqueza ~ area)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-4.4614	-2.1245	-0.5101	1.4862	7.6863

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	5.226994	1.352158	3.866	0.00477	**
area	0.007374	0.001431	5.155	0.00087	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 3.555 on 8 degrees of freedom
```

```
Multiple R-squared: 0.7686, Adjusted R-squared: 0.7397
```

```
F-statistic: 26.57 on 1 and 8 DF, p-value: 0.00086930
```

# ls e rm: lista e remove objetos

```
> ls()
[1] "area"      "modelo1"   "modelo2"   "previsto"  "riqueza"   "varea"

> rm(varea)

> ls()
[1] "area"      "modelo1"   "modelo2"   "previsto"  "riqueza"
```

# Como o R trabalha?

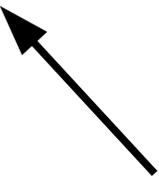
- Os objetos criados são mantidos na memória, até que sejam salvos.
- Por *default*, todos os objetos são salvos em um único **arquivo oculto** no diretório de trabalho, chamado **.RData**.
- Ao sair do programa com a função **q**, o R pergunta se deve salvar as alterações.
- É recomendável salvar seu trabalho regularmente, com o comando **save.image**

# q :

## Encerra a sessão do R

Sai do programa mas salva a área de trabalho em  
.RData (se a resposta for “y” !)

```
> q()  
Save workspace image? [y/n/c]: y
```



Se a resposta for “n” você perde  
todas as alterações !

**save.image** , **save** e **load**:

Salva e carrega a área de trabalho

Salva a área de trabalho no arquivo default (.RData)

```
> save.image()
```

Salva a área de trabalho em outro arquivo

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

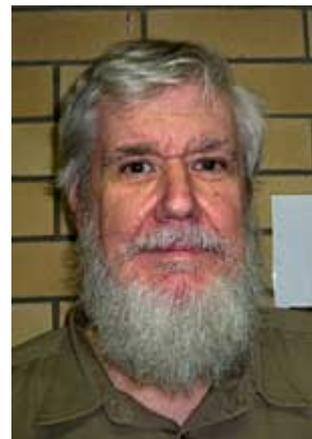
Carrega um arquivo de área de trabalho

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

# Como Trabalhar com o R?

## Dica 1: Separe cada projeto em um diretório

- R works best if you have **a dedicated folder for each separate project** – the working folder. Put all data files, &c, in the working folder (or in subfolders of it)
- **Start R in the working folder**: three ways
  - make an R shortcut pointing to the folder and double-click
  - double-click on the **.RData** file in the folder, when it exists
  - double-click any R shortcut and use **setwd()**
- Work on the project – your objects can be automatically saved in the **.RData** file
- To quit – use **q()**



# getwd e setwd:

Mostra e muda o diretório de trabalho

## Qual o diretório de trabalho atual?

```
> getwd()  
[1] "/home/paulo/work/Pos_grad/Eco_USP/cursoR/2011"
```

## Muda o diretório de trabalho

```
> setwd("/home/paulo/work/treinos_R/")
```

## Verificando ...

```
> getwd()  
[1] "/home/paulo/work/treinos_R"
```

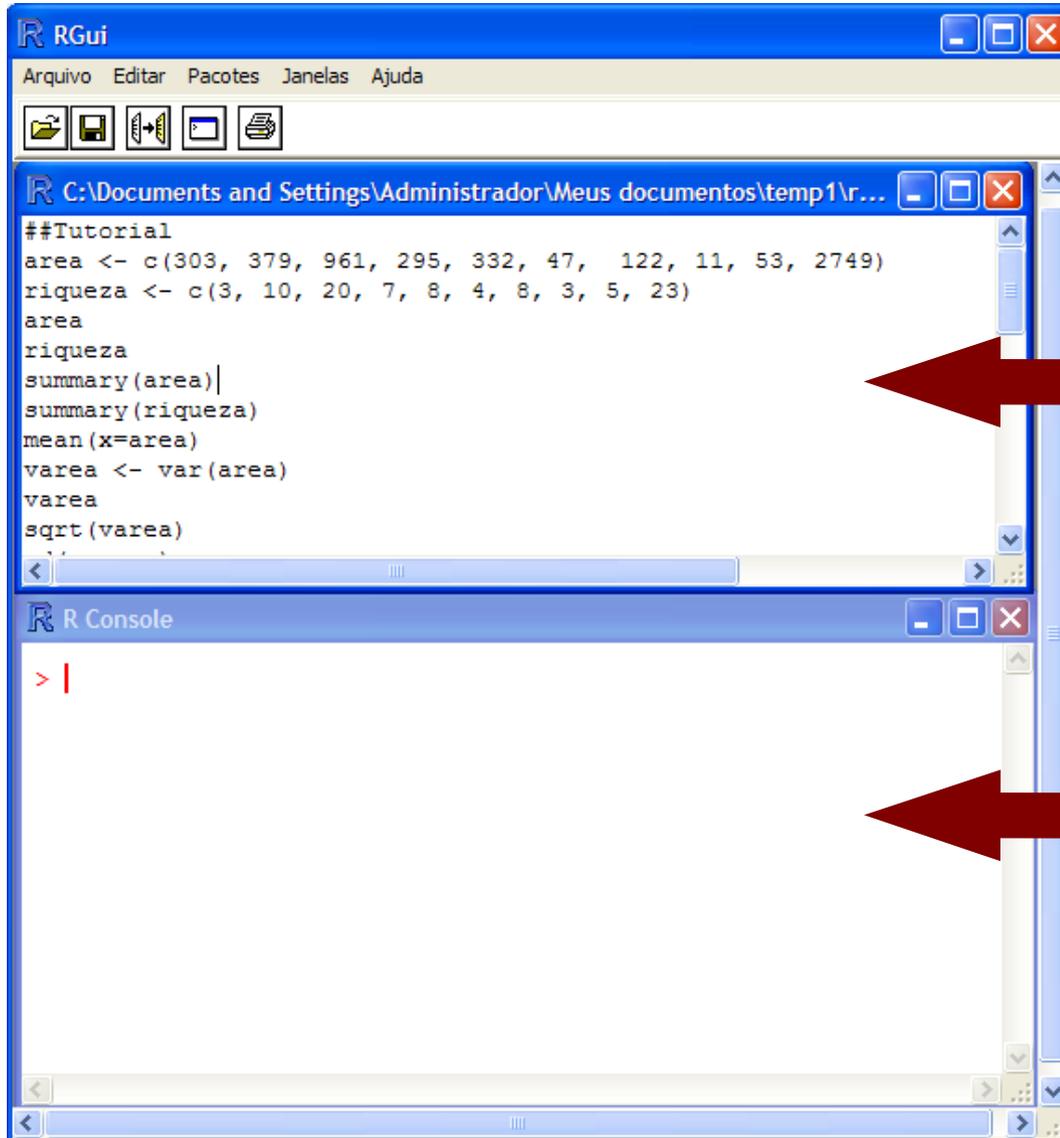
# Como Trabalhar com o R?

## Dica 2: salve seus comandos em um arquivo

- Use um editor de textos para escrever seus comandos em um arquivo.
- Salve este arquivo em seu diretório de trabalho com extensão `.r` ou `.R`
- Envie seus comandos ao R. Três maneiras:
  - Com o comando **source** ("arquivo.R"): executa todos os comandos.
  - Na interface R-GUI do Windows, use o editor de scripts e **ctrl-R** para enviar linhas ou blocos para.
  - Use editores próprios para programação em R (e.g., Emacs com módulo ESS, Tinn-R).

# Como Trabalhar com o R?

Dica 2: salve seus comandos em um arquivo



Arquivo de código  
(*script*)

Janela do R  
(R- console)

# Editores e IDE's

- Editor no R-GUI (Windows e Mac)
- Tinn-R (Windows): <http://sciviews.org/Tinn-R/>
- NotePad++ & NPPtoR (Windows):  
<http://npptor.sourceforge.net/>
- TexMate (Mac): <http://macromates.com/>
- Rgedit (Linux) : <http://sourceforge.net/projects/rgedit/>
- Emacs + ESS : <http://ess.r-project.org/>
- R Studio: [www.rstudio.org](http://www.rstudio.org)
- E muito mais, veja no wiki!

# Classes de Objetos

```
> class(modelo1)
[1] "lm"
```

```
> names(modelo1)
[1] "coefficients" "residuals"      "effects"        "rank"
[5] "fitted.values" "assign"         "qr"            "df.residual"
[9] "xlevels"      "call"          "terms"         "model"
```

O objeto `modelo1` pertence à classe `lm` (linear models), que tem resultados de uma regressão linear.

Esta classe de objeto contém outros objetos, como os vetores de valores previstos e de resíduos, a fórmula do modelo ajustado e muito mais.

# Classes de Objetos

```
> x.1
[1] 1 2 3 4 5
> class(x.1)
[1] "integer"
> x.2
[1] 0.33333333 0.16666667 0.08333333
> class(x.2)
[1] "numeric"
> x.3
[1] "a" "b" "c" "d" "e"
> class(x.3)
[1] "character"
> x.4
[1] TRUE TRUE FALSE FALSE FALSE
> class(x.4)
[1] "logical"
```

# **is.[classe]** e **as.[classe]**: Verificação e Coerção de Classes

```
> x.1
[1] 1 2 3 4 5
> x.1t <- as.character(x.1)
> x.1t
[1] "1" "2" "3" "4" "5"
> x.5 <- c("TRUE", "TRUE", "TRUE", "FALSE")
> class(x.5)
[1] "character"
> is.logical(x.5)
[1] FALSE
> x.5 <- as.logical(x.5)
> class(x.5)
[1] "logical"
```

# C

## Criação de Vetores

```
> a = c(3.4, pi, exp(-1))  
> a  
[1] 3.40000000 3.1415927 0.3678794
```

- Um objeto da classe `vector` no R é um conjunto de elementos de uma mesma classe (números, caracteres, lógicos, etc).
- Um objeto vetor no R não corresponde aos vetores de álgebra matricial (para isso há a classe `matrix`).
- A função `c` combina elementos em um vetor.

# factor

## Classe de fatores

```
> sexo <- rep(c("F", "M"), each=9)
> sexo
[1] "F" "M"
[11] "M" "M" "M" "M" "M" "M" "M" "M" "M"
> class(sexo)
[1] "character"
> sexo <- factor(sexo)
> sexo
 [1] F F F F F F F F F F M M M M M M M M M M
Levels: F M
> class(sexo)
[1] "factor"
```

# table

## Contagens de Observações

```
> sexo
 [1] F F F F F F F F F M M M M M M M M M
Levels: F M
> table(sexo)
sexo
F M
9 9
> dieta
 [1] L L L D D D N N N L L L D D D N N N
Levels: N D L
> table(sexo,dieta)
      dieta
sexo  N  D  L
  F   3  3  3
  M   3  3  3
```

# levels

## Níveis de um fator

```
> politicos <- factor(rep("corrupto",10),
                      levels=c("corrupto","honesto"))
> levels(politicos)
[1] "corrupto" "honesto"
> politicos
 [1] corrupto corrupto corrupto corrupto corrupto
 [6] corrupto corrupto corrupto corrupto corrupto
Levels: corrupto honesto
> table(politicos)
politicos
corrupto  honesto
      10         0
```



# FIM DA PRIMEIRA AULA

Hoje à tarde:

Aula 2 – Funções matemáticas

Plantão tutoriais e exercícios – Lab. Bioquímica I

Postar no wiki até quarta:

Listas 1 e 2 de Exercícios:

[http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01\\_curso\\_atual:exercicios1](http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:exercicios1)

[http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01\\_curso\\_atual:exercicios2](http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:exercicios2)