R e pacote EcoVirtual

Para os roteiros deste wiki indicados com o ícone você deve ter o ambiente R instalado em seu computador, com o pacote **Ecovirtual**. Siga as instruções abaixo para instalar.

Instalação rápida

1. Instale o R. Instruções aqui.
2. No R instale o pacote devtools com o comando:

```
install.packages("devtools")
```

3. Em seguida digite os comandos para instalar o Ecovirtual:

```
library(devtools)
install_github("EcoVirtual", "ecovirt")
install_github("EcoVirtualPlugin", "ecovirt")
```

4. Siga para a seção Ecovirtual no Rcmdr

Instalação longa

Siga estas instruões se teve problemas com a instalação rápida.

Os roteiros com o pacote EcoVirtual foram elaborados para serem utilizados no ambiente de programação . Ele não é um software do tipo aplicativo, a preocupação não é com amigabilidade, mas com flexibilidade e capacidade de manipulação de dados e realização de análises.

Os roteiros podem ser seguidos utilizando-se apenas a interface básica do , com linhas de comando, copiando-se o código e executando as linhas diretamente no programa. Como nossa intenção não é ensinar o uso do [1], preparamos um pacote com as ferramentas do EcoVirtual baseado em uma interface gráfica de usuário (GUI) chamada Rcommander.

O Rcommander é uma interface gráfica com menu interativo, criada por John Fox, que permite a inclusão de novas ferramentas através de plugins. As ferramentas do EcoVirtual estão separadas em dois pacotes: (1) EcoVirtual com as funções básicas e (2) RcmdrPlugin.EcoVirtual que cria o menu de interface no Rcommander. Para rodar os roteiros usando a interface gráfica do Rcommander é necessário instalar:

1. O ambiente de programação R: Instalando o R
2. O pacote Rcommander dentro do R: Instalando o Rcommander
3. Os pacotes do EcoVirtual disponíveis nesse wiki: Instalando o EcoVirtual.

Arquivos de instalação

Comece baixando os arquivos de instalação do **EcoVirtual** abaixo:

Ruindous

- ecovirtual_0.05.zip}}preservefilenames:autofilled:EcoVirtual_0.05.zip
- RcmdrPlugin.EcoVirtual_0.05.zip}}preservefilenames::RcmdrPlugin.EcoVirtual_0.05.zip

Linux e Mac

- ecovirtual_0.05.tar.gz}}preservefilenames:autofilled:EcoVirtual_0.05.tar.gz
- rcmdrplugin.ecovirtual_0.05.tar.gz}}preservefilenames:autofilled:RcmdrPlugin.EcoVirtual_0.05.tar.gz

——

Alexandre Adalardo 2013/11/13 19:34

Instalando o R

Na página principal do clique no link para o CRAN (o que é isto? veja no wiki!), e escolha o repositório mais próximo de você para baixar arquivos de instalação.

Siga as instruções das FAQS para baixar e instalar a versão mais recente do para seu sistema operacional (Windows/ Mac/ Linux).

Vá à seção de pacotes adicionais do , que te enviará para uma lista e breve descrição dos pacotes disponíveis para baixar. Nela você pode ter uma idéia da quantidade e diversidade de aplicações que a comunidade de usuários do já desenvolveu. Uma visão temática destes pacotes é fornecida nas "Task Views", cujo link está na página de pacotes.

Veja a Task view "Environmetrics" , que descreve as aplicações disponíveis para ecologia. Escolha um dos pacotes e instale em seu computador.

Instalando R no Linux

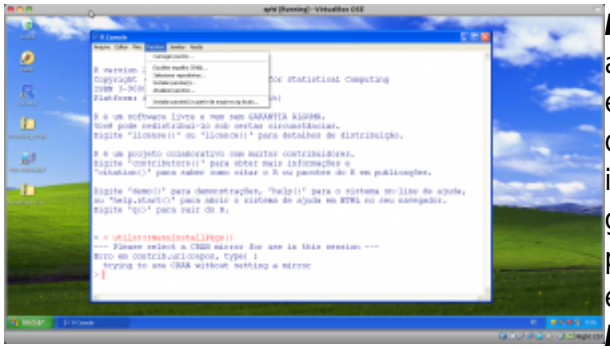
Para as distribuições linux que usam o repositórios Debian, pode-se baixar diretamente o pacote pelo programa de gerenciamento de pacotes (Synaptic no Ubuntu e Debian) e selecionar dentro o pacote ***r-base*** *ou r-base-core*. Após marcar para a instalação, deve aceite a instalação dos pacotes adicionais, isso irá instalar automaticamente a distribuição básica do R.

No **Ubuntu** há a alternativa de instalar pelo aplicativo ***Ubuntu Software Center*** que vem na

distribuição base. No programa busque *R core* e antes de instalar olhe as informaçôes adicionais onde há opções de intalação de alguns módulos importantes. Para aqueles que querem instalar diretamente a partir do pacote do CRAN, há um bom tutorial para instalação do R disponível em: R no linux

Instalando o Rcommander

ruWindows

Para instalar o R commander no Windows abra o R e no menu clique em ***Pacotes***, em seguida em  ***Instalar Pacotes***. Na janela de repositórios que se abrirá escolha um localidade mais próxima de onde se encontra. Em seguida aparecerá a lista de pacotes disponíveis no repositório, selecione ***Rcmdr*** e solicite da instalação. Após a instalação, para abrir a interface gráfica do Rcmdr é necessário carregar o pacote que pode ser feito no mesmo menu ***Pacotes*** selecionando em seguida ***Carregar pacotes...***. Na janela selecione o ***Rcmdr***.

ATENÇÃO

Quando o Rcmdr estiver carregando uma janela abrirá solicitando a instalação de pacotes adicionais. Você não precisa deles!! **NÃO INSTALE ESSES PACOTES**, o processo é demorado e dependendo da velocidade de conexão da internet pode demorar horas!

Linux

Para os sistemas operacionais Unix-like pode-se instalar o pacote diretamente no console do R digitando:

```
install.packages("Rcmdr")
## para iniciar o Rcmdr:
library(Rcmdr)
```

Rcmdr no Mac-OS

Para Mac-Os, veja as instruções no link: Instalando o Rcmdr no Mac

Abaixo um resumo do que devem fazer, qualquer problema acessem a página acima que é a página oficial do Rcmdr, escrita pelo seu idealizador John Fox.

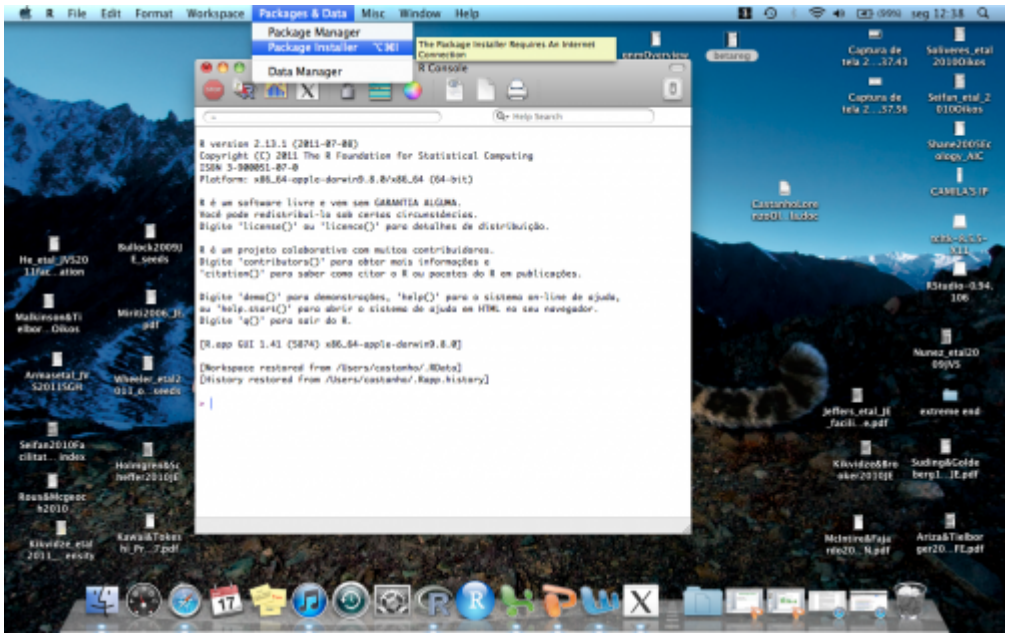
Em geral, para a correta instalação do Rcmdr no Mac é necessário que seja instalado antes um

aplicativo que permite a execução da interface gráfica *tcltk*.

- 1. Baixe o aplicativo do linque abaixo:

http://socserv.mcmaster.ca/jfox/Courses/soc3h6/tcltk-8.5.5-x11.dmg

- 2. Instale o arquivo clicando duas vezes nele. Caso abra uma janela mostrando um outro arquivo com o mesmo nome, repita a operação.
- 3. Siga as instruções de instalação e verifique que o aplicativo foi corretamente instalado.
- 4. Abra o R e entre no menu ***Packages & Data***, em seguida em ***Packages-Installer*** e selecione a opção de repositórios ***Cran Source***, em seguida procure o Rcmdr e solicite a instalação.



Instalando o EcoVirtual

Instale apenas um vez

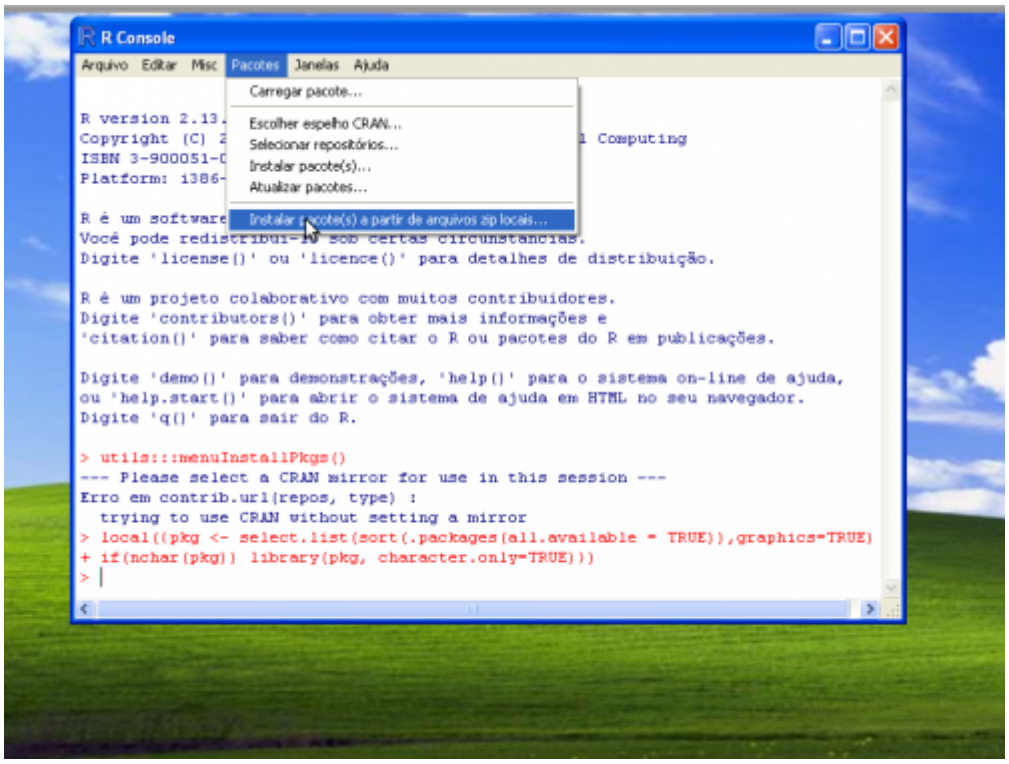
- **Atenção, se ainda não instalou o pacote Rcmdr retorne a sessão Instalando o Rcommander**
- Não é necessário instalar o EcoVirtual todas as vezes que for usá-lo, apenas se houver uma versão mais atual!
- Para reabrir o EcoVirtual instalado, apenas reinicie o Rcmdr e ative o plugin como indicado no item EcoVirtual no Rcmdr desta página.

Siga as instruções abaixo, dependendo do seu sistema operacional:

ruWindows

- 1. Baixe os arquivos de instalação do **Ecovirtual** aqui.
- 2. No menu do R → ***Pacotes***, vá em ***Instalar pacote(s) a partir de arquivo .zip local***. Selecione o arquivo EcoVirtual_x.xxx.zip[2)] e instale.

- 3. Em seguida, No menu do R → ***Pacotes***, vá em ***Instalar pacote(s) a partir de arquivo .zip local***. Selecione o arquivo RcmdrPlugin.ecovirtal_x.xxx.zip.

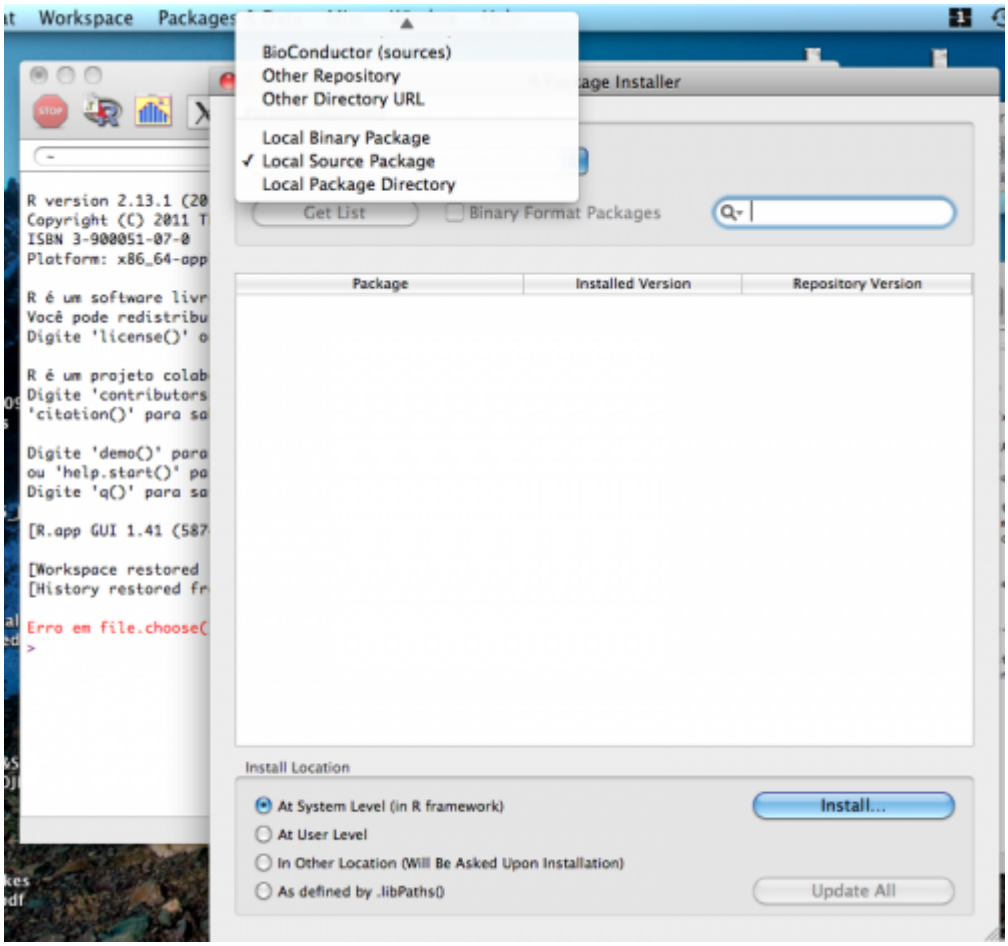


Linux e Mac

Ambos os sistemas usam a base Unix, portanto os arquivos dos pacotes são os mesmos. Baixe os dois arquivos a seguir no seu computador e siga as instruções específicas para o seu sistema.

Mac

- 1. Baixe os arquivos de instalação do **Ecovirtual** aqui.
- 2. Vá ao menu ***Packages & Data*** , em seguida em ***Packages Installer***. Na janela que se abre, mude a seleção para ***Local Source Packages***



- 3. Na nova janela que se abre, navegue até o diretório em que salvou os arquivos e selecione *ecovirtual_x.xxx.tar.gz*[3)]
- 4. Clique em ***Install...***
- 5. Repita o mesmo procedimento para o arquivo *rcmdrplugin.ecovirtual_x.xxx.tar.gz*

Linux

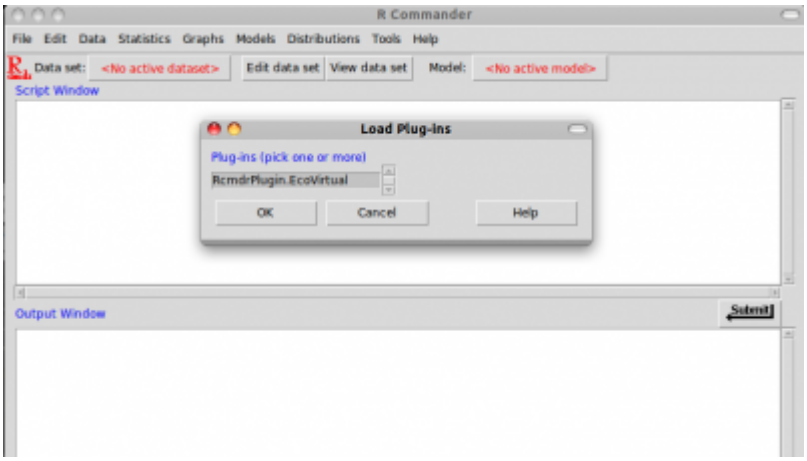
- 1. Baixe os arquivos de instalação do **Ecovirtual** aqui.
- 2. Execute o R
- 3. Na linha de comando do R digite o código abaixo, modificando `/caminho` para o caminho para o diretório onde você gravou o arquivo de instalação:

```
install.packages("/caminho/ecovirtual_x.xxx.tar.gz" ,repos=NULL)

install.packages("/caminho/rcmdrplugin.ecovirtual_x.xxx.tar.gz" ,repos=NULL)
```

Onde `x.xxx` é a versão dos arquivos de instalação.

EcoVirtual no Rcmdr



- Após instalar os pacotes do **EcoVirtual** abrir o Rcmdr. Caso o Rcmdr já estaja aberto no seu computador é necessário fechá-lo e abri-lo apenas após a instalação dos pacotes do EcoVirtual.
- Na janela do Rcmdr vá ao menu ***Tools*** e depois clique na opção ***Load Rcmdr Plugin(s)…***. Se a instalação do EcoVirtual foi bem sucedida aparecerá o Plugin para ser ativado.
- O Rcmdr irá solicitar que seja reinicializado para ativar o menu do EcoVirtual.
    - Explore a nova opção menu ***EcoVirtual*** que aparecerá na barra superior do Rcmdr.

Problemas & Soluções

Caso a opção de instalação do EcoVirtual-Plugin não esteja disponível no Rcmdr, houve algum problema na instalação dos pacotes. Por ordem de prioridade, tente as seguintes soluções:

- reiniciar o R e o Rcmdr e verifique se a opção do plugin aparece
- Retorne ao item anterior e reinstale ambos os pacotes do EcoVirtual, verificando as mensagens que aparecem no R verificando se não há nenhuma mensagem de erro na instalação
- cheque se realmente as instalações foram feitas com sucesso digitando no prompt do R:

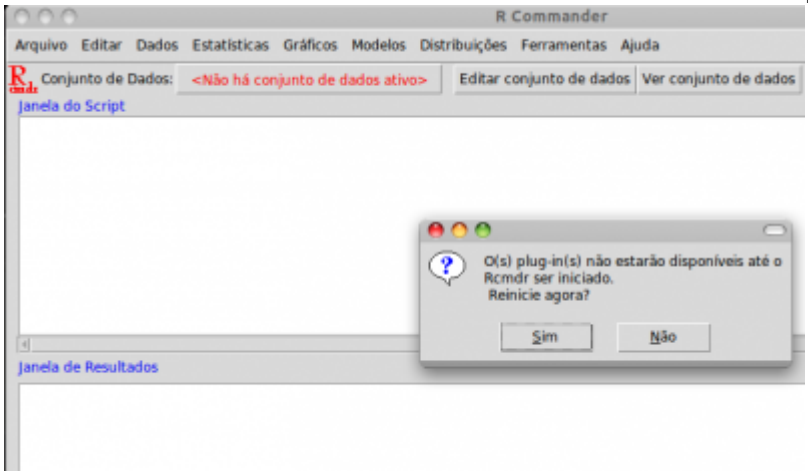
```
library()
```

Os pacotes do EcoVirtual devem estar entre os pacotes listados!

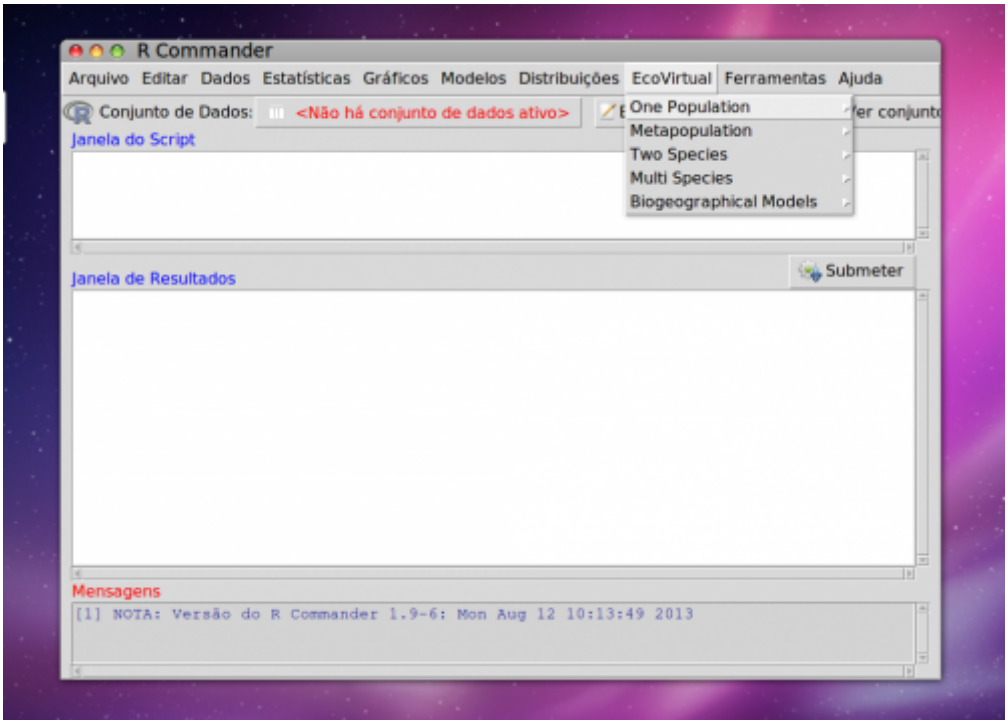
- entre em contato com o monitor Renato
- entre em contato com o professor Alexandre

Última fase

DIGA SIM! Clique em **sim** quando o Rcmdr pergunt



A janela do Rcmdr deverá reabrir, agora com o menu **EcoVirtual** ativado:



R para iniciantes

Boa parte dos roteiros deste site usam o ambiente de programação estatística R. Você não precisa saber programar em R para seguir os roteiros, mas é recomendável que se familiarize com a filosofia do R, e o modo de executar comandos.

Este guia foi preparado por Marcel Caritá Vaz e Vitor Rios. É uma introdução voltada para quem nunca teve contato com o ambiente R.

Prazer, meu nome é Gui, R Gui

Em nossas práticas iremos utilizar um programa que interage com você por meio de linhas de comando, o que significa que ele tem uma linguagem própria que vocês podem não conhecer. À primeira vista vai parecer assustador, mas vocês verão que não é tão ruim assim. Este roteiro atuará como seu guia e tradutor. Sigam-no e ninguém sairá ferido. *Bon voyage!*

Primeiro vamos abrir o tal programa e nos familiarizar um pouco com ele.

Se você usa Windows ou Mac procure no desktop, dock menu ou menu iniciar do seu computador um *R* azul e estiloso como o da figura acima. Para Linux, recomendamos que você use o Rstudio[4)].

Abram o programa e vocês verão uma tela árida e sem vida (Lembrem-se de que a Terra era assim também há alguns bilhões de anos atrás e veja no que ela se tornou). Note que há algumas poucas opções de comandos no menu superior. Esse é o GUI (Graphic User Interface) que apesar de básico irá ajudá-lo em algumas tarefas mais simples.

Sem desanimar, vamos utilizar o *R* para fazer a coisa mais simples que ele sabe: operações aritméticas. Note que os comandos que devem ser inseridos no programa estarão sempre no formato de texto *code*, com um fundo azulado.

Durante esse roteiro terá duas opções: (1) copiar os códigos que aparecem no roteiro em um fundo azul claro e colar no console do R ou (2) digitar diretamente no console do *R* os comando que estão no roteiro. Não esqueça de que devem dar um *ENTER* depois de escrever cada linha de comando. Veja o exemplo de linhas de comando abaixo e teste-as no *R*.

```
2+2
5-1
2*3
4/5
```

Notem que há um sinal de > em vermelho no canto esquerdo. Isso mostra o início da linha de comando. As linhas de comando, ou as ordens dadas ao programa, vão sempre ficar em vermelho, enquanto que as respostas que ele nos der vão sempre ficar em azul. Repare também que sempre, antes do resultado, aparece um **[1]**. Isso nada mais é do que um contador, quer ver? Vamos pedir agora uma seqüência de 1 a 100.

```
1:100
```

Note que o contador mostra a posição do primeiro valor de cada linha de resultado.

Um pouco mais de R

Continuando, uma coisa que o *R* gosta bastante de fazer é criar objetos. Para tanto basta escolher um nome e atribuir-lhe algum valor ou conjunto de valores ou mesmo palavras. Se você quiser que seu objeto tenha mais de um elemento basta escrever **c(elemento 1,elemento 2,etc)**. Vamos tentar?

```
valor <- 5
```

```
pares <- c(0,2,4,6,8)
cores <- c("azul","vermelho","amarelo")
```

Quando usamos o símbolo “=” ou “<” seguido de “-” estamos criando objetos com um nome que aparece a esquerda e que contém alguns elementos (o que vem a direita do símbolo). Quando criamos um objeto, ele fica guardado na memória do *R* até que você feche o programa. Porém, os objetos criados ficam ocultos. Para ver a lista de arquivos ocultos basta dar o seguinte comando:

```
ls()
```

Já para ver o conteúdo de cada objeto basta chamá-lo pelo nome:

```
valor
pares
cores
```

Muito bem. Espero que até aqui ninguém tenha morrido **ainda**. Agora vamos fazer um dos truques mais bacanas do *R*: que comece a jogatina! Você sabia que podemos jogar dados com esse programa? Quer apostar?

```
moeda <- c("cara","coroa")
moeda
sample(moeda,1)
sample(moeda,1)
sample(moeda,1)
sample(moeda,1)
```

A cada vez que você roda *sample(dado,1)* o *R* faz um sorteio. Vamos jogar uns dados agora.

```
dado <- c(1:6)
dado
sample(dado,1)
sample(dado,1)
sample(dado,1)
```

Gostaram? Mas não vão ficar viciados, hein?!

Finalmente, vamos às brincadeiras de gente grande. Em algumas de nossas práticas vamos trabalhar com matrizes. Felizmente é muito fácil transformar um conjunto de dados em uma matriz utilizando o comando *matrix()*.

```
1:100
matrix(1:100,nrow=20,ncol=5)
```

Note que os argumentos *nrow* define o número de linhas (20) e *ncol* o de colunas (5).

Mas podemos ir além e criar um objeto tridimensional do tipo array:

```
array(1:100,dim=c(5,5,4))
```

Note que o argumento agora é *dim* deve conter os valores de cada dimensão c('linha', 'colunas', 'matriz'). O resultado aparece como fatias de um bolo, onde cada camada é uma matriz. Conseguiu visualizar?

Outra coisa bacana é que podemos criar objetos vazios. (*NA* quer dizer não se aplica, ou seja, aquela posição não está sendo ocupada por nenhum valor.)

```
a=rep(NA,10)
a

b=matrix(NA,nrow=3,ncol=3)
b

c=array(NA,dim=c(3,3,3))
c
```

Mas qual a graça de se fazer isso? A graça é que depois podemos preencher os lugares vazios com os valores que quisermos.

```
a[1]=2
a
a[1]

b[3,3]="oi"
b
b[2,2]="oi"
b
b[1,1]="oi"
b
b[3,3]

c[3,3,1]="oi"
c
c[3,3,2]="ai"
c
c[3,3,3]="ui"
c
c[3,3,3]
```

Pronto, agora sem querer querendo vocês já sabem indexar! O que é isso? É simplesmente dizer a posição de um objeto que você quer. Se for um objeto simples do tipo linear, como é o caso de *a*, basta escrever o nome do objeto e em seguida um valor dentro dos colchetes (**nome[posição]**). Se for uma matriz como o *b*, você tem que dizer **nome[número da linha, número da coluna]**. Finalmente, se for um objeto tridimensional como o *c* você deve dizer **nome[número da linha, número da coluna, número da fatia do bolo]**.

Gráficos

Outro forte do *R* é a criação de gráficos. Veja alguns exemplos extremamente simples:

```
plot(1:10,10:1)
plot(1:10,10:1,pch=2)
plot(1:10,10:1,pch=2,col=2)
plot(1:10,10:1,pch=2,xlab="eixo x")
plot(1:10,10:1,pch=2,xlab="eixo x",ylab="eixo y")


hist(rnorm(1000),col=33)
arrows(2,100,2,70,col=2)

boxplot(iris$Sepal.Length~iris$Species)
```

Funções no R

Agora, o golpe de misericórdia: como criar uma função. Criar uma função nada mais é do que programar o computador para fazer uma tarefa em seu lugar. Soou interessante? Pois é, mas isso exige um pouco mais do nosso esforço. Vamos ver um exemplo bem simples. Vou ensinar o programa a calcular a média de um conjunto de valores.

```
media=function(objeto){sum(objeto)/length(objeto)}
media(c(1,2,3))

alturas=c(1.4,1.7,2.0,1.6,1.8)
media(alturas)
```

Note que entre parênteses estão os valores ou o objeto dos quais você quer a média. Entre chaves está o comando que calcula a média. Note também que para calcular a média usei duas funções pré-existentes no *R*: *sum* e *length*. Na verdade também já existe uma função pronta que calcula a média: *mean*.

```
sum(alturas)    # soma dos elementos do objeto "alturas"
length(alturas) # número de elementos do objeto "alturas"
mean(alturas)   # média dos elementos do objeto "alturas"
```

O "#" em cada linha de comando significa que o *R* deve desconsiderar o texto. Isso é bom: quer dizer que o programa não se mete na conversa dos outros.

Bom, isso é tudo, ou quase tudo, o que precisávamos saber sobre a lógica do *R* para fazermos a prática. Não se preocupem com detalhes sobre a linguagem. Contem com seu guia: o roteiro a seguir. E se ele falhar, por favor não hesitem em clamar pela ajuda dos professores e monitores. Boa sorte.

Para você que quer mais!

Se você se interessou pelo programa não se sinta mal. Ele realmente é incrível, pois te permite fazer quase tudo que quiser. Visite o site da disciplina do R, oferecida para os alunos de pós graduação da Ecologia da USP. O começo, como em toda linguagem, é uma subida árdua. Mas com o tempo a coisa chega a ser até prazerosa. *Marcel*

Carregando Arquivos no R

O R também permite ler arquivo de dados, para que você não tenha de digitar tudo à mão sempre que for fazer suas análises.

Geralmente estamos acostumados a salvar nossas planilhas de dados no Excel (formato .xls e .xlsx), e fazer as análises e transformações de dados diretamente na planilha, porém este não é o melhor procedimento para o uso do R. A melhor prática é salvar os dados originais em um arquivo, e fazer as análises diretamente no R, salvando os comandos utilizados para a análise em outro arquivo, com a extensão .R (que chamamos de "script"), e os resultados da análise em um terceiro arquivo. Parece mais prático ter tudo em um arquivo só, mas a filosofia do R é a reciclagem dos comandos: se você salvar os comandos separado dos dados originais, pode usar esse arquivo para analisar outros dados semelhantes, sem correr risco de sobrescrever ou misturar os dados de dois experimentos diferentes. Pense quantas vezes você já teve de fazer uma ANOVA no Statistica ou SPSS ou transformar seus dados? Usar um script para automatizar sua análise salva tempo, dor de cabeça e lesões por esforço repetitivo.

Definindo o diretório de trabalho

Tudo que você fizer no R será salvo ou carregado a partir do seu diretório de trabalho. Para isso, é necessário indicar ao R em qual diretório você está trabalhando, usando os comandos

```
getwd()
```

e

```
setwd()
```

. Primeiro, verifique em que diretório o r está digitando getwd(). por padrão, no windows o diretório é a sua pasta documentos. para alterar isso, utilize o comando setwd()

```
setwd("C:/meu_drive/meu_diretorio")
```

Note que o endereço do diretório tem de estar entre aspas, e as barras ("\") tem de ser trocadas por barras invertidas("/"). Após usar o comando acima, confira se funcionou usando o getwd(). você também pode alterar o diretório utilizando o menu Arquivo → Mudar dir

Formatação dos arquivos

O R é capaz de ler arquivos de texto em uma variedade de formatos. Sugerimos que você digite seus dados em um planilha no Excel, OpenOffice ou LibreOffice, e salve os dados em um formato especial chamado ".csv" (No Excel, vá em Arquivo → Salvar Como → outros formatos, e escolha CSV (separado por virgulas)) Isso transforma sua planilha em um arquivo de texto, em que cada célula é separada das outras por uma virgula.

Muita calma nessa hora!

O R segue o padrão americano de formatação de números, portanto os

> decimais devem ser separados por pontos e não por vírgulas. Revise bem seus dados antes de começar suas análises! Sugerimos definir o seu editor de planilhas para sempre usar o padrão americano, para evitar confusões. Busque na ajuda do seu editor como fazer isso (o procedimento é diferente para cada programa)

Mas porque tanta firula quanto a salvar os dados em um formato especial? Bem, primeiro de tudo, o formato .csv nada mais é que um arquivo de texto puro, que pode ser lido em qualquer computador, independente de ser Mac, Windows ou Linux, e de ter o MS Office instalado ou não. Além disso, as empresas mudam frequente os formatos dos seus arquivos, gerando incompatibilidade e forçando você a obter a versão mais recente do programa (duvida? tente abrir uma planilha do Excel 2013 em um PC com Excel 2003). Salvar em um formato padronizado e aberto garante que daqui a dez anos você ainda vai poder ter acesso a seus dados e colaborar com pessoas que usem outros programas.

Com seus dados devidamente arrumados e salvos, podemos prosseguir para a leitura dos arquivos no R

Lendo os arquivos

Agora basta dizer ao R qual arquivo você quer que ele leia, e como esse arquivo está formatado. para isso usamos a função `read.table()` e guardamos o resultado em uma tabela

```
tabela = <- read.table(file="DadosDoMeuDoutorado.csv",header=T,sep=",")
```

O que isso quer dizer? esse comando diz ao R e os dados do meu doutorado estão no arquivo "DadosDoMeuDoutorado.csv", que possui uma linha de cabeçalho com o nome das minhas colunas (`"header=T"`), e que minhas colunas são separadas por vígulas (`sep=","`). Se seu arquivo nao tiver cabeçalho, use `header=F`, e se suas colunas estiverem separador por outro símbolo, por exemplo ponto-e-vírgula, coloque o símbolo dentro das aspas do parâmetro `sep` (`sep=";"`).

Após rodas essa linha de comando, meus dados estão guardados dentro do objeto `tabela`, e agora posso realizar minhas análises sem me preocupar em sobrescrever os dados originais, que estão são e salvos no arquivo DadosDoMeuDoutorado.csv, no meu diretório de trabalho (por falar nisso, você já fez o backup dos seus dados hoje?)

Para se aprofundar

- Impatient R: introdução muito didática e bem humorada.
- R em 4668 palavras: uma introdução mais extensa ao R, que inclui os principais procedimentos estatísticos. Boa apresentação também da filosofia de programação com dados e código aberto do R.
- Disciplina de R da PG-Ecologia USP: tutoriais, exercícios e muitos links para outros recursos.
- Learn R: excelente curso, do USGS. Tutoriais, exercícios e vídeo-aulas.

Maxima

O Maxima é um *CAS* (Computer Algebra System) semelhante a sistemas como o *Mathematica*, *Maple* e outros que manipulam expressões simbólicas, com a vantagem de ser um programa de código aberto. Ele tem procedimentos para operar com números e também com expressões matemáticas.

É uma ferramenta essencial para obter resolver problemas matemáticos. Entretanto, para utilizar as ferramentas deve entender o quê está fazendo, porquê está fazendo e para quê! Abaixo o linque do site para baixar o Maxima:

- Maxima download

O Máxima já vem com uma interface gráfica para usuário chamada wxMaxima. Para usuários de sistemas Linux (Ubuntu e outros) o wxMaxima está disponível diretamente nos repositórios da distribuição.

Atenção: Se você tiver problemas para executar o Maxima em Windows, talvez seja necessário configurar o seu firewall.

Maxima em 10 passos

Esse tutorial[5)] contém os 10 primeiros passos para iniciantes no Maxima poderem acompanhar alguns roteiros deste sítio. Para quem está acostumado com as linhas de comando de programação em R, fica muito mais fácil usar o Maxima [6)]

Caso queira rodar esse tutorial a partir do arquivo do código do WxMaxima, carregue o arquivo do tutorial no seu computador.

Maxima como calculadora

- 1. Vamos começar com alguns exemplos de cálculos simples! Abaixo está

uma célula de entrada com uma simples adição. Coloque o cursor sobre ele e pressionar SHIFT-ENTER

para avaliá-lo.

```
1 + 1;
```

Se você não obter quaisquer erros, o Maxima está configurado corretamente. Se você fez um erro, você deve verificar a configuração ou wxMaxima visite o website wxMaxima (http://wxmaxima.sourceforge.net/) para instruções sobre como configurar e wxMaxima Maxima corretamente!

- 2. Assumindo que você tenha resolvido seus problemas, vamos fazer mais algumas

cálculos (mais uma vez - colocar o cursor na célula de entrada abaixo e pressionar SHIFT-ENTER para avaliar o código)!

```
 5!;
% * 10;
%o1 * 100;
1 / 3;
1.0 / 3.0;
```

Na célula de entrada acima, nós mandamos cinco linhas de comandos, para o Maxima. Cada linha deve terminar com um "; " ou um "\$". No caso da linha termina com um "; ", Maxima irá mostrar o resultado da linha, enquanto os resultados das linhas terminando com o "\$" serão suprimidos. O "\$" vem a calhar quando se faz mais cálculos. Note também que o resultado de "03/01" e "1.0/3.0" diferem. Isso porque Maxima, ao contrário de programas de matriz numérica (Matlab, etc) tenta manter cálculos precisos - expressões como 1 / 3 ou sqrt (2) não são avaliadas a menos que explicitamente solicitado. Em "1.0/3.0" foi utilizado a aproximação de "Números de Ponto flutuante" (Floating_point) para o Maxima avaliar a expressão.

- 3. Podemos, no entanto, solicitar ao Maxima uma aproximação de ponto flutuante de uma expressão. Rode o script abaixo e observe os resultados.

```
sqrt(2 * %pi);
float(%);
```

Na linha "float (%);" foi utilizado o símbolo "% ". Este símbolo contém sempre o resultado da última linha avaliados. Símbolos numerados com "o" (output) como "% o1", "% o2" armazenam os resultados que aparecem quando as células de entrada (input: %i1,%i2,...) são avaliadas.

- 4. Nós também podemos armazenar, não só números, mas as expressões inteiras, em variáveis. Use "variable_name: valor;" forma para armazenar o valor em "variable_name". Execute a célula abaixo e observe.

```
radius: 10 $
height: 100 $
area: %pi * radius^2;
volume: area * height;
```

Vamos avaliar o último resultado númericamente:

```
float(%);
```

Operando funções

- 5. Até agora usamos o Maxima apenas como uma calculadora normal. Vamos agora fazer algo que não é possível em uma calculadora simples, definir e operar funções:

```
f(x) := x^2 + a$
f(5);
f(5), a = -5;

h(x):=  1 / (x^2 + a);
h(5);
h(5), a=-20;
float(%);

f1(x):= exp(x);
g1(x):= -x^2;
f1(g1(x));
```

- 6. Agora que você já aprendeu o básico, é hora para alguns exemplos gerais de matemática! Lembre-se: se você quiser saber mais sobre uma função específica, clique sobre ele e pressione F1. Resolvendo equações usando o “solve”:

```
solve(a*x^2 + b*x + c = 0, x);
```

- 7. O Maxima, além de muitas outras coisas, faz gráficos. Rode o código abaixo!

```
wxplot2d([sin(x), cos(x)], [x,0, 2*%pi]);
wxplot3d( exp(-x^2 - y^2), [x,-2,2],[y,-2,2]);
```

Enfim, derivadas

- 8. Vamos tentar uma diferenciação usando a função “diff” .

```
f(x) := x^2 $
diff(f(x), x);
g(y) := sin(y)$
g(f(x));
diff( g(f(x)) , x);
```

Sim, o Maxima sabe a “Regra de Cadeia”!

- 9. Agora que já explorou o Maxima, vamos obter algumas derivadas. A sintaxe é bastante simples: primeiro forneça a função e em seguida defina qual a variável a ser diferenciada.

```
diff(x^2, x);
diff(log(x),x);
diff(sin(x)+ cos(x),x);
diff(sin(x)/x^2, x);
```

- 10. Muitas vezes o resultado da expressão matemática da derivada parece diferente do que

calculou.

Pode ser que não seja um erro, apenas uma forma diferente de expressar a mesma função. Para simplificar expressões matemáticas no Maxima podemos usar a função ratsimp.

```
ratsimp(%);
```

Maxima: more steps

$\lim\limits_{maxima\to \infty}= \infty$

Maxima: more steps

Gostou!

- Para uma consulta rápida e novas dicas veja o site A 10 minute tutorial for solving math problems with Maxima
- Caso queira se aprofundar um pouco mais sobre o Maxima, o guia introdutório da Bruna Santos da Universidade do Porto é um ótimo começo: Maxima Bruna Santos 2009

maxima, derivada

1) veja o curso Uso da Linguagem R

2) onde x.xxx é o número da versão

3) onde x.xxx é a versão do pacote

4) também disponível para outros sistemas operacionais

5) tutorial baseado na documentação oficial do programa, modificado por — *Alexandre Adalardo de Oliveira 2012/05/02 17:47*

6) há uma opção de pacote no R (Ryacas) que faz a interface com um outro sistema de manipulação simbólica de expressões matemáticas: o yacas

From:
http://ecovirtual.ib.usp.br/ -

Permanent link:
http://ecovirtual.ib.usp.br/doku.php?id=en:ecovirt:roteiro:soft:roteiros

Last update: **2017/08/17 14:26**