



Universidade do Estado do Rio de Janeiro – UERJ
Instituto de Matemática e Estatística
Departamento de Estatística
Projeto: Software Livre para Alunos de Estatística



Apostila do Curso de Extensão:

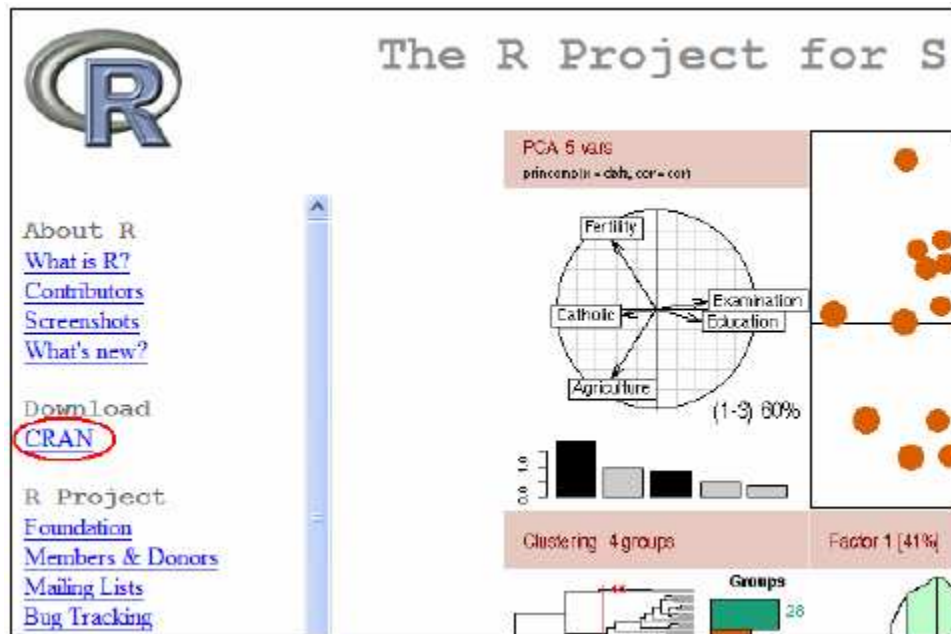
Software Estatístico Livre R


Marcelo Rubens dos Santos do Amaral
Carolina de Vasconcelos Cesário
Guilherme de Almeida Gonçalves
Luciano Teixeira Schultz
Thiago Souza
Kauê Veras

Apoio: Sub-reitoria de Extensão e Cultura – SR-3 - UERJ

- **Download do R**

Primeiramente, deve-se abrir o site www.r-project.org . Em seguida, clicar em “CRAN” e escolher o servidor de seu país e mais próximo da cidade em que está. Em seguida, escolher “Windows”, clicar em “base”, em “Download R 2.9.0 for ‘Windows’” e salvar. Observe:





CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Information on the status of the mirrors can be found [here](#).

<p>About R</p> <p>What is R?</p> <p>Contributors</p> <p>Screenshots</p> <p>What's new?</p> <p>Download</p> <p>CRAN</p> <p>R Project</p> <p>Foundation</p> <p>Members & Donors</p> <p>Mailing Lists</p> <p>Bug Tracking</p> <p>Developer Page</p> <p>Conferences</p> <p>Search</p>	<table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top;"> <p>Argentina</p> <p>Australia</p> <p>Austria</p> <p>Belgium</p> <p>Brazil</p> <p>Canada</p> </td> <td style="vertical-align: top;"> <p>http://mirror.cricyt.edu.ar/r/</p> <p>http://cran.ms.unimelb.edu.au/</p> <p>http://cran.at.r-project.org/</p> <p>http://www.freeststatistics.org/cran/</p> <p>http://cran.br.r-project.org/</p> <p>http://cran.fiocruz.br/</p> <p>http://www.vps.fmvz.usp.br/CRAN/</p> <p>http://brieger.esalq.usp.br/CRAN/</p> <p>http://cran.stat.sfu.ca/</p> </td> <td style="vertical-align: top;"> <p>CONICET, Mendoza</p> <p>University of Melbourne</p> <p>Wirtschaftsuniversitaet Wien</p> <p>K.U.Leuven Association</p> <p>Universidade Federal do Parana</p> <p>Oswaldo Cruz Foundation, Rio de Janeiro</p> <p>University of Sao Paulo, Sao Paulo</p> <p>University of Sao Paulo, Piracicaba</p> <p>Simon Fraser University, Burnaby</p> </td> </tr> </table>	<p>Argentina</p> <p>Australia</p> <p>Austria</p> <p>Belgium</p> <p>Brazil</p> <p>Canada</p>	<p>http://mirror.cricyt.edu.ar/r/</p> <p>http://cran.ms.unimelb.edu.au/</p> <p>http://cran.at.r-project.org/</p> <p>http://www.freeststatistics.org/cran/</p> <p>http://cran.br.r-project.org/</p> <p>http://cran.fiocruz.br/</p> <p>http://www.vps.fmvz.usp.br/CRAN/</p> <p>http://brieger.esalq.usp.br/CRAN/</p> <p>http://cran.stat.sfu.ca/</p>	<p>CONICET, Mendoza</p> <p>University of Melbourne</p> <p>Wirtschaftsuniversitaet Wien</p> <p>K.U.Leuven Association</p> <p>Universidade Federal do Parana</p> <p>Oswaldo Cruz Foundation, Rio de Janeiro</p> <p>University of Sao Paulo, Sao Paulo</p> <p>University of Sao Paulo, Piracicaba</p> <p>Simon Fraser University, Burnaby</p>
<p>Argentina</p> <p>Australia</p> <p>Austria</p> <p>Belgium</p> <p>Brazil</p> <p>Canada</p>	<p>http://mirror.cricyt.edu.ar/r/</p> <p>http://cran.ms.unimelb.edu.au/</p> <p>http://cran.at.r-project.org/</p> <p>http://www.freeststatistics.org/cran/</p> <p>http://cran.br.r-project.org/</p> <p>http://cran.fiocruz.br/</p> <p>http://www.vps.fmvz.usp.br/CRAN/</p> <p>http://brieger.esalq.usp.br/CRAN/</p> <p>http://cran.stat.sfu.ca/</p>	<p>CONICET, Mendoza</p> <p>University of Melbourne</p> <p>Wirtschaftsuniversitaet Wien</p> <p>K.U.Leuven Association</p> <p>Universidade Federal do Parana</p> <p>Oswaldo Cruz Foundation, Rio de Janeiro</p> <p>University of Sao Paulo, Sao Paulo</p> <p>University of Sao Paulo, Piracicaba</p> <p>Simon Fraser University, Burnaby</p>		

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](#)

Subdirectories:

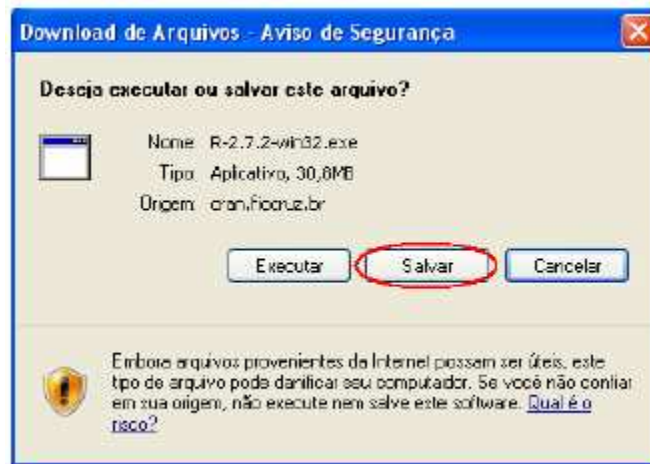
<p>base</p> <p>contrib</p>	<p>Binaries for base distribution (managed by Duncan Murdoch)</p> <p>Binaries of contributed packages (managed by Uwe Ligges)</p>
--	---

R-2.9.0 for Windows

[Download R 2.9.0 for Windows](#) (36 megabytes)

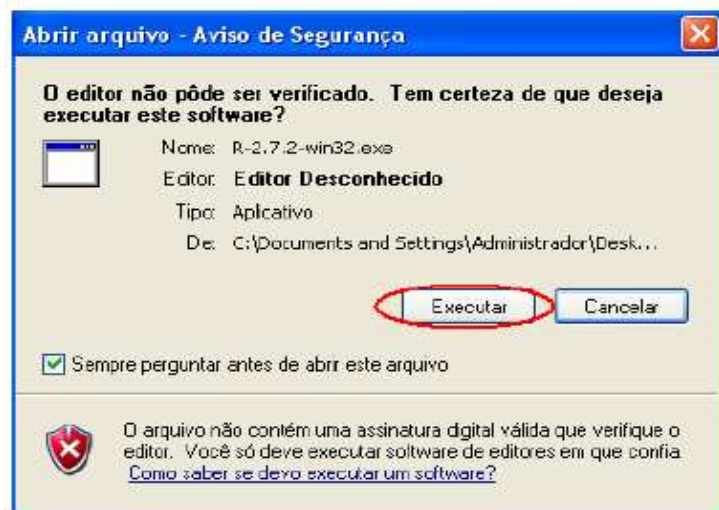
[Installation and other instructions](#)

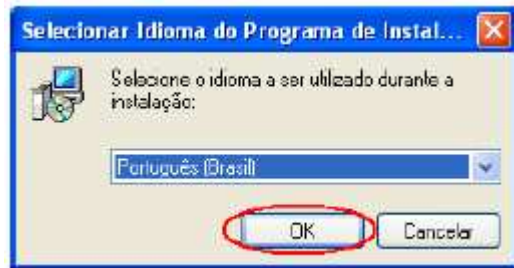
New features in this version: [Windows specific](#), [all platforms](#).



Após a conclusão do download, o R ainda deve ser instalado para que possa começar a ser usado. Para isso, devem-se dar dois cliques com o mouse no arquivo baixado e seguir as instruções de instalação das figuras abaixo:

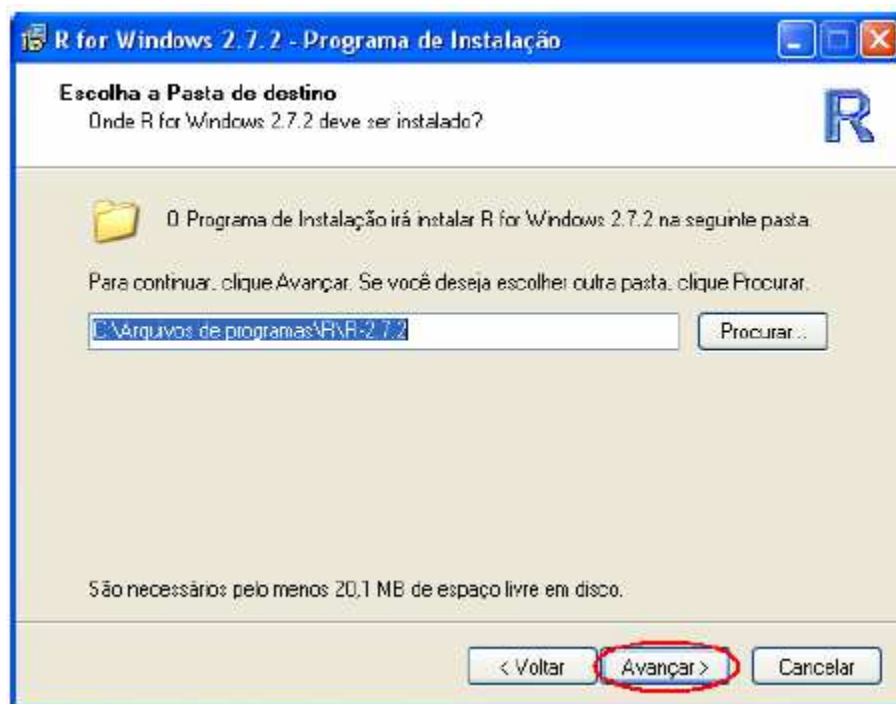
- **Instalando o R:**

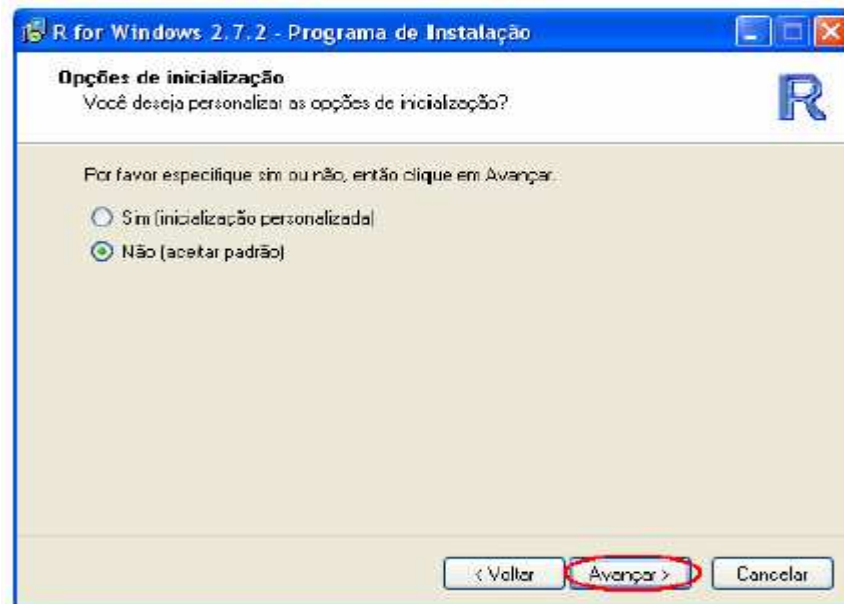
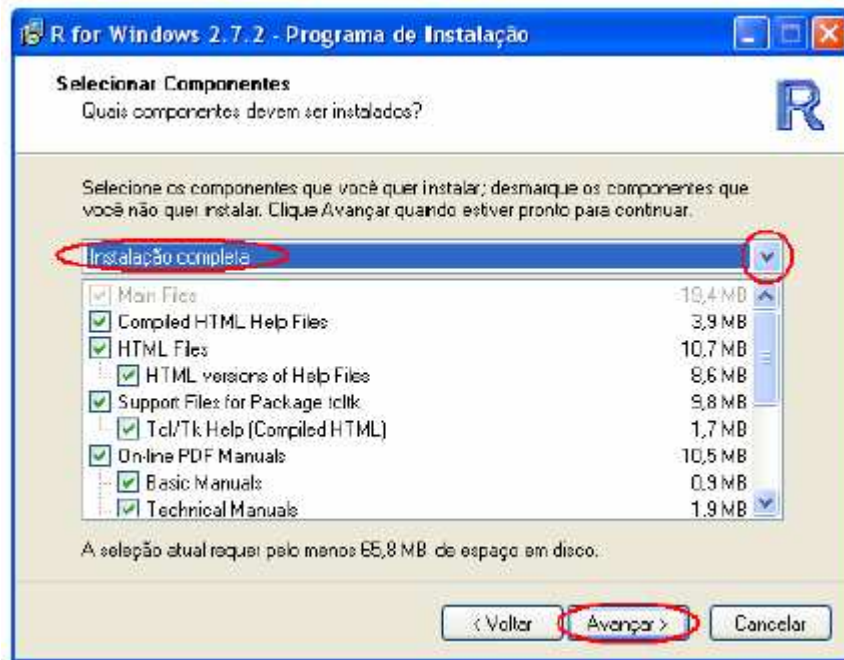


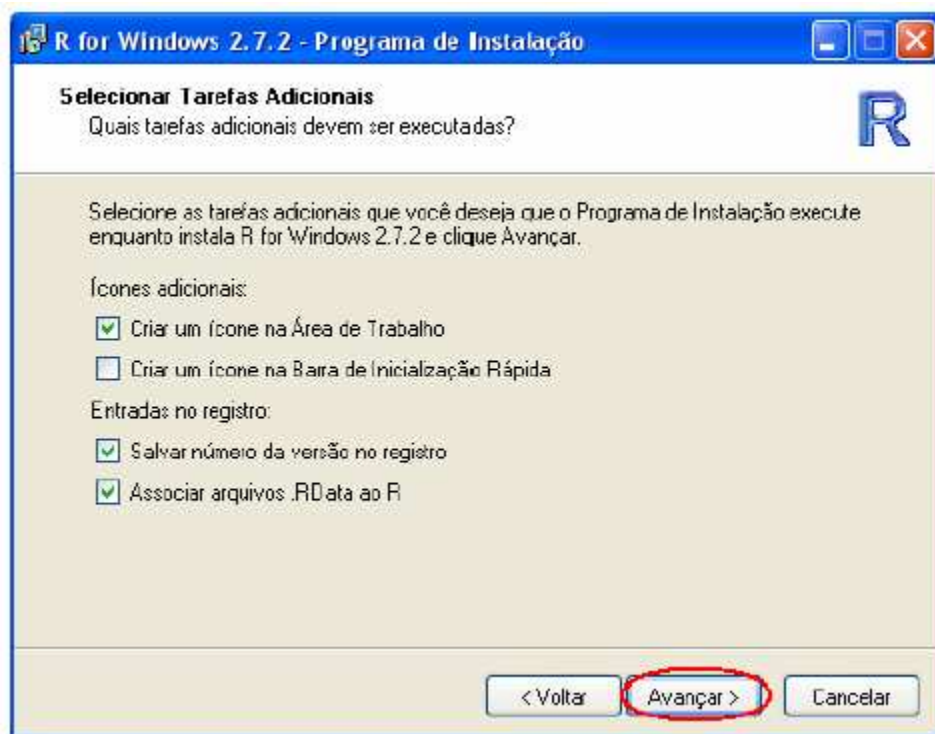




Escolher a pasta onde se quer salvar o programa:







Após o término da instalação, aparecerá uma janela de finalização do instalador. Nela, clicar em "Concluir". A partir daí, o R já pode ser usado.

Observação: As janelas exibidas nesta apostila foram as do Windows XP para as instalações das versões: R 2.7.2, R 2.8.0, R 2.8.1 e R 2.9.0.

- **Pacotes do R**

No R, existe uma grande diversidade de pacotes. Estes pacotes contêm um conjunto de funções que permitem ou facilitam a realização das análises estatísticas, além de possuírem ajuda para suas funções e demonstrações de execução.

Ao instalar o R, apenas alguns pacotes vêm juntos, os quais são essenciais para o funcionamento do programa, os quais denominamos de módulo ou pacote básico, além de muitos poderem servir de base e pré-requisito para o funcionamento de outros pacotes.

Os pacotes extras podem ser encontrados com base no próprio programa ou pelo site, www.r-project.org, ressaltando que em ambos os casos, o usuário deve estar conectado à Internet. Na versão 2.9.1 existem 1.778 pacotes disponíveis para serem instalados.

Instalando pacotes a partir do R:

Após abrir o R, deve-se seguir as instruções abaixo:

Clique em “pacotes” no menu.

Clique em “instalar pacote(s)”. (abrirá uma janela)

Selecione o(s) nome(s) do pacote(s) desejado(s), clique em “OK”.

Aguarde o download ser concluído.

No console do R, aparecerá um texto indicando que o(s) pacote(s) selecionado(s) foram instalado(s).

Muito importante: Para utilizar as funções do pacote, é preciso antes, carregá-lo no próprio R. Este processo será explicado mais à frente nesta apostila.

Instalando pacotes a partir da Internet:

Primeiramente, deve-se abrir o site www.r-project.org e seguir as etapas abaixo, clicando no link destacado:



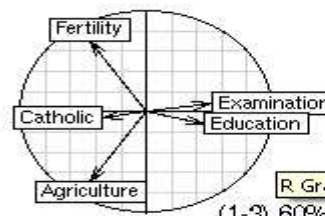
The R Project : Compre

About R
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

Download,
Packages
[CRAN](#)

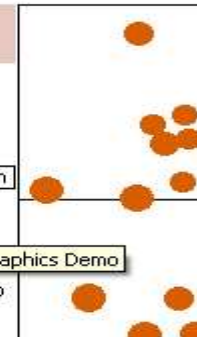


PCA, 5 vars
`princomp(x = data, cor = cor)`



R Graphics Demo

(1-3) 60%



The Comprehensive R Archive Netwo
location close to you. Some statistics c

Argentina

<http://mirror.cricyt.edu.ar/r/>

Australia

<http://cran.ms.unimelb.edu.au/>

Austria

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

Belgium

<http://www.freeststatistics.org/cr>

Brazil

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

<http://cran.fiocruz.br/>

About R
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

Download,
Packages
[CRAN](#)



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** (2009-04-17): [R-2.9.0.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](#).
- Contributed extension [packages](#)

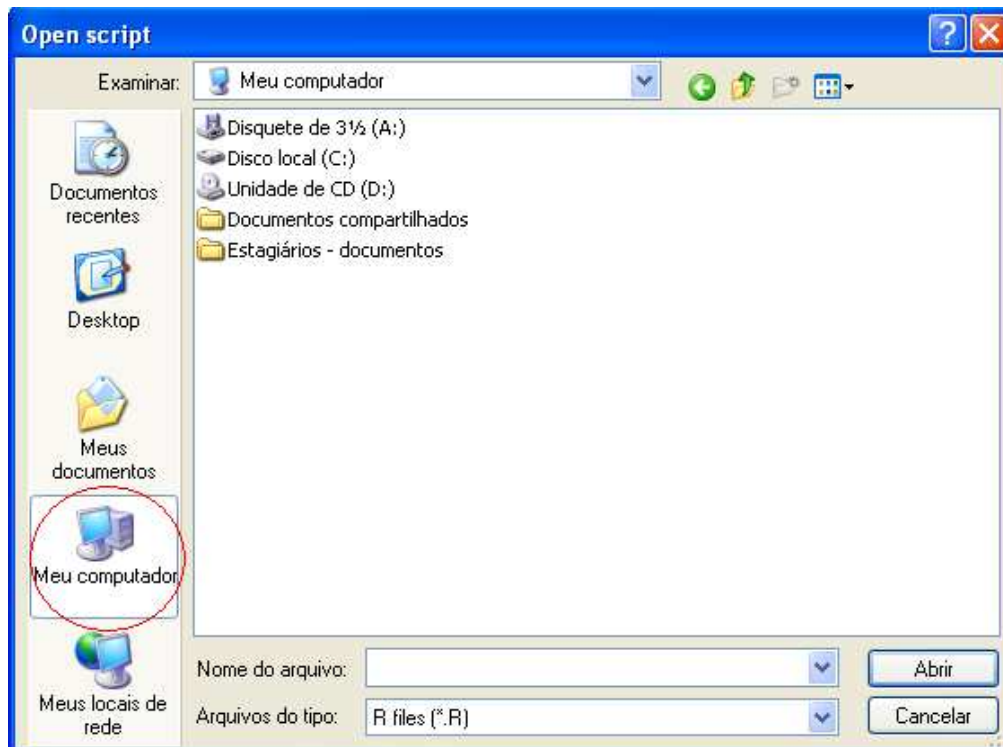
Ao clicar em “packages”, será aberta uma nova página. No final desta página, terão todos os pacotes possíveis a serem baixados. Deve-se então clicar no link do pacote desejado, fazer o download e salvá-lo. Na hipótese de que não haja restrição de espaço no HD, recomendamos que todos os pacotes sejam instalados. Isso irá demanda, na versão 2.9.1, ao todo, 2,81 GB de espaço no HD.

Ao encerrar o download, os arquivos ficam salvos em uma pasta, onde é possível, ver o “help” dos pacotes, ou seja, ver a descrição apresentando a função de cada um deles. É possível ver não só o desempenho estatístico e/ou gráfico que cada um realiza, mas também exemplos com bancos de dados da funcionalidade de cada pacote. Observe as instruções ilustrativas abaixo de como utilizar estes recursos:

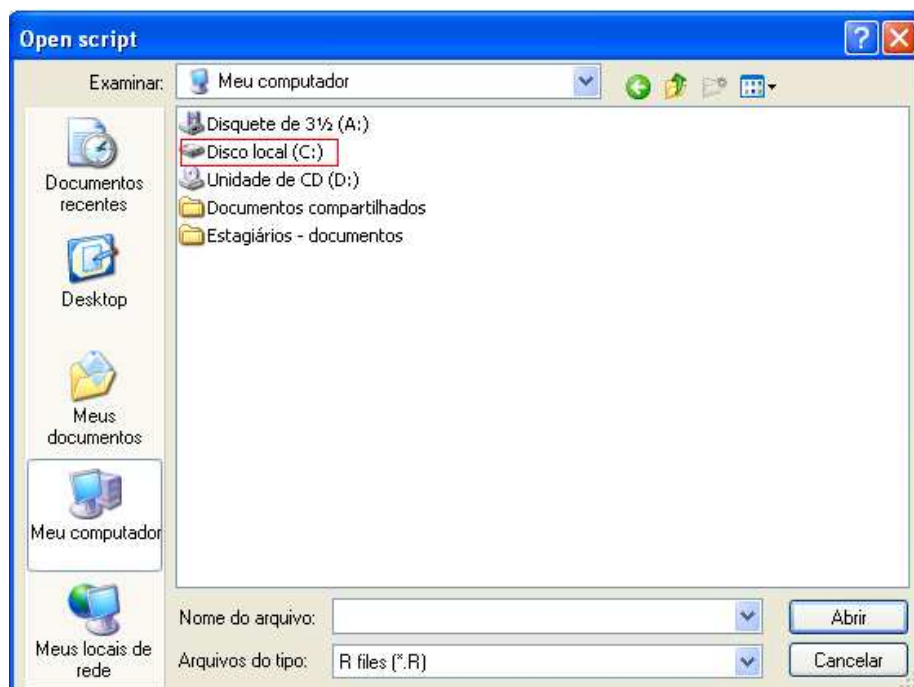
Para abrir a página onde ficam salvos os arquivos, deve-se clicar no símbolo de ‘abrir script’ no menu do R:



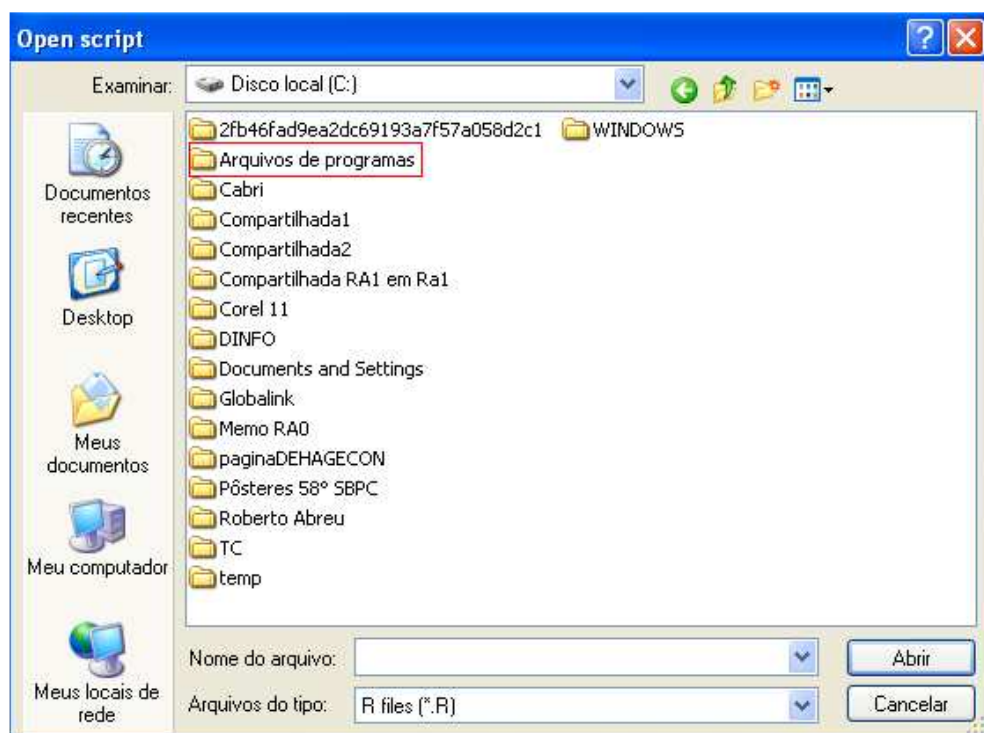
Aguarde abrir uma janela e clique em “Meu computador”:



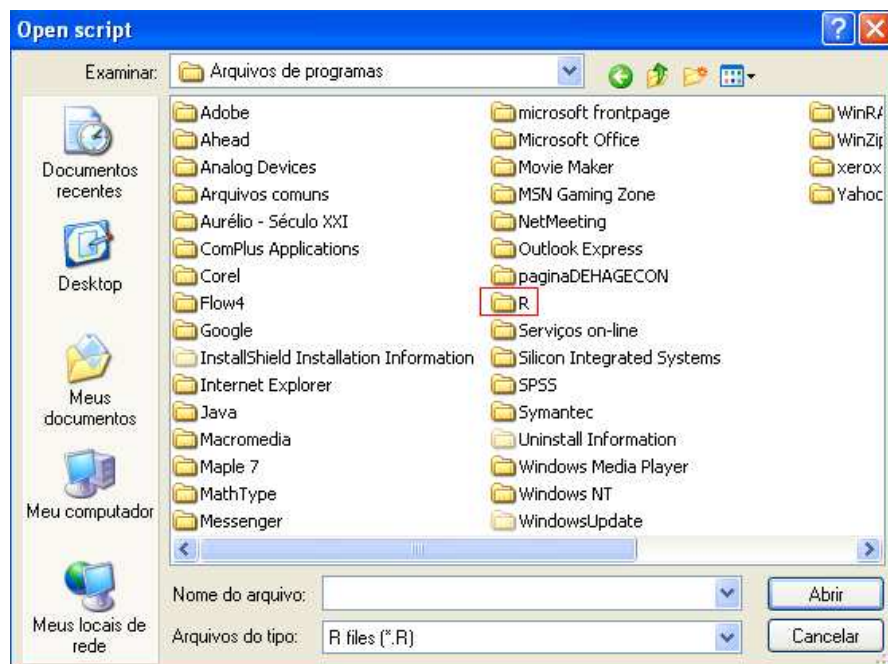
Em seguida, clique em “Disco local (C:)” :



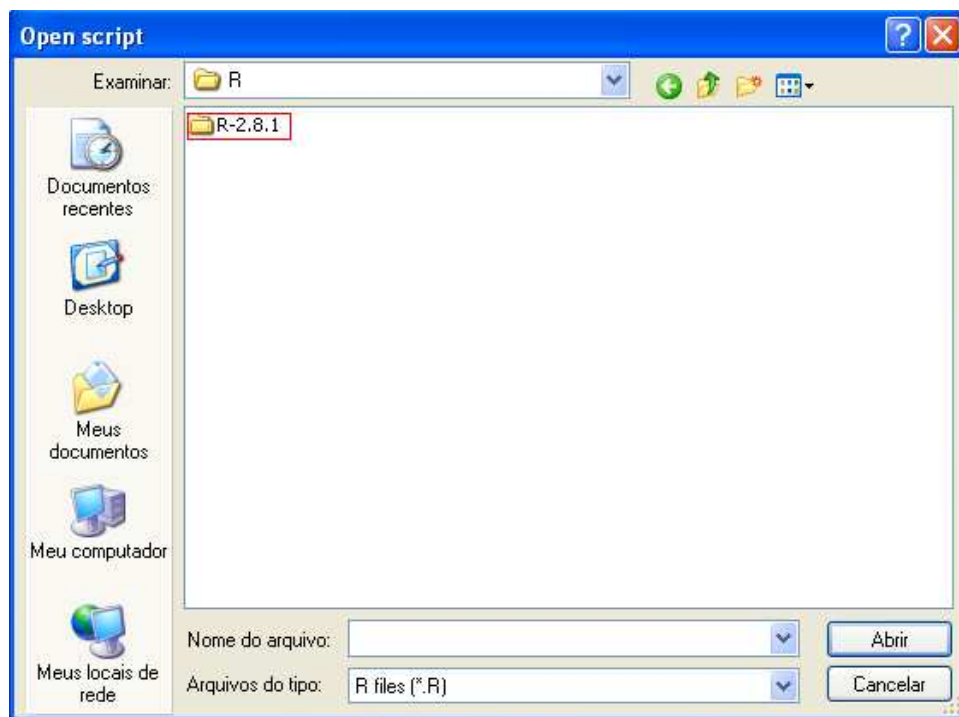
Clique em “Arquivos de programas” :



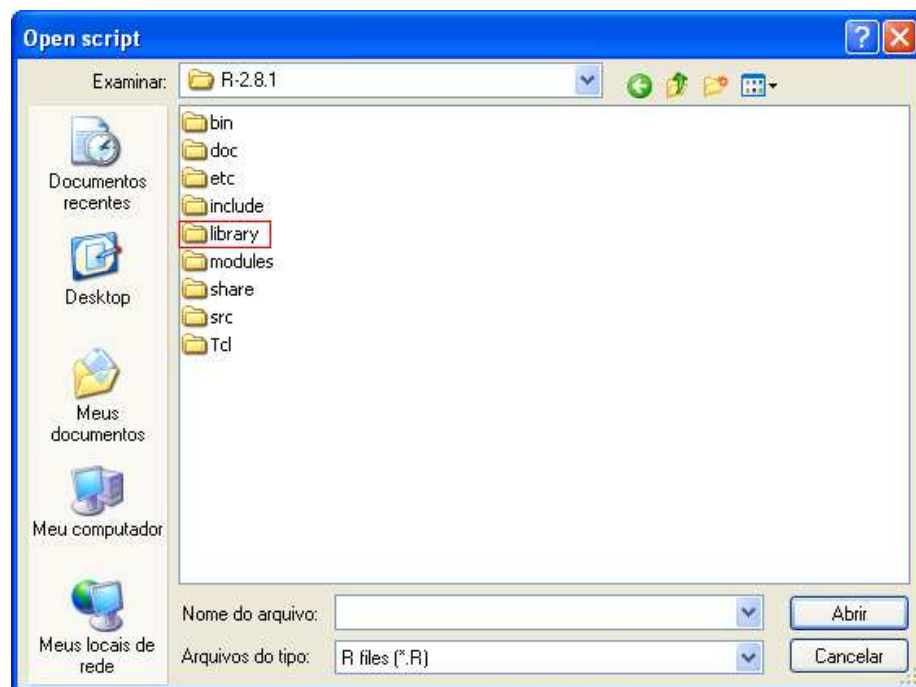
Clique na pasta “R” :



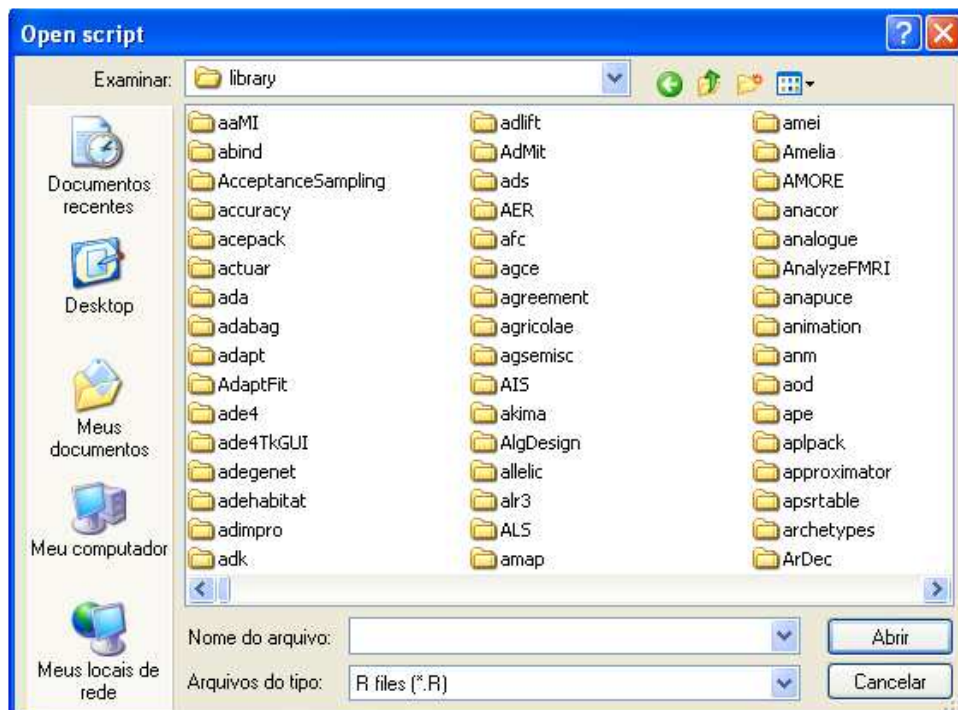
Clique na versão baixada do R:



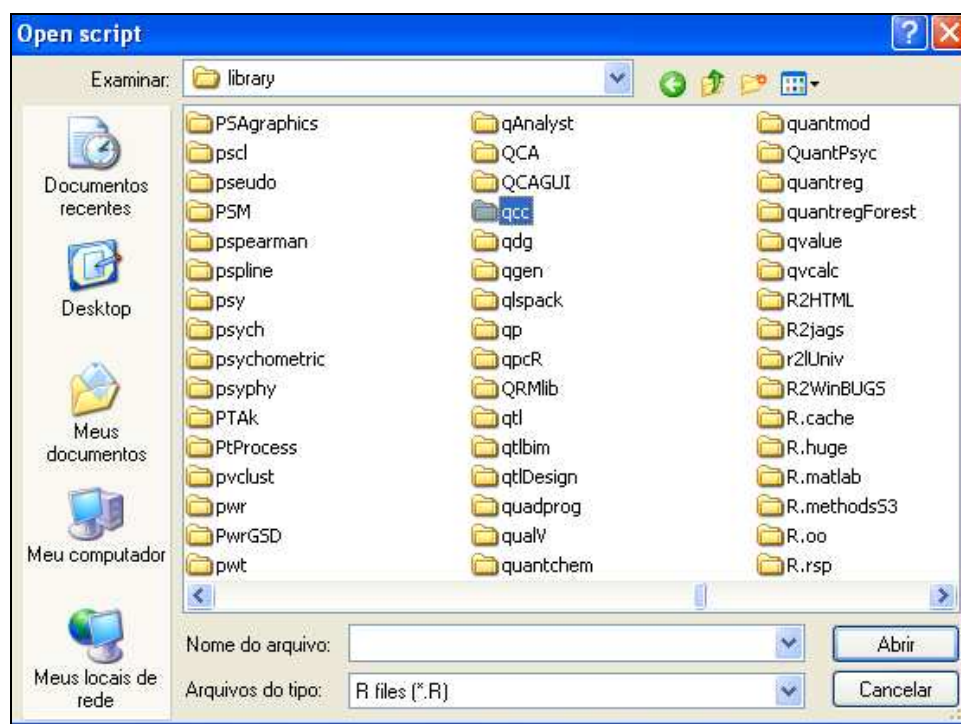
Clique em “library” :



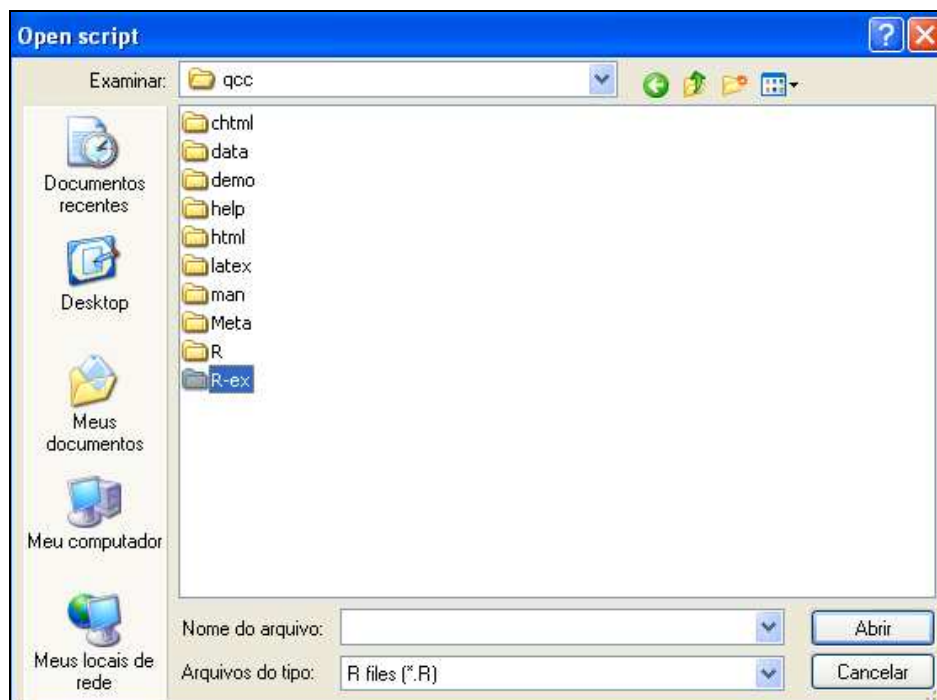
Em seguida, será aberta a pasta “library”, onde estarão armazenados todos os pacotes já instalados. Veja:



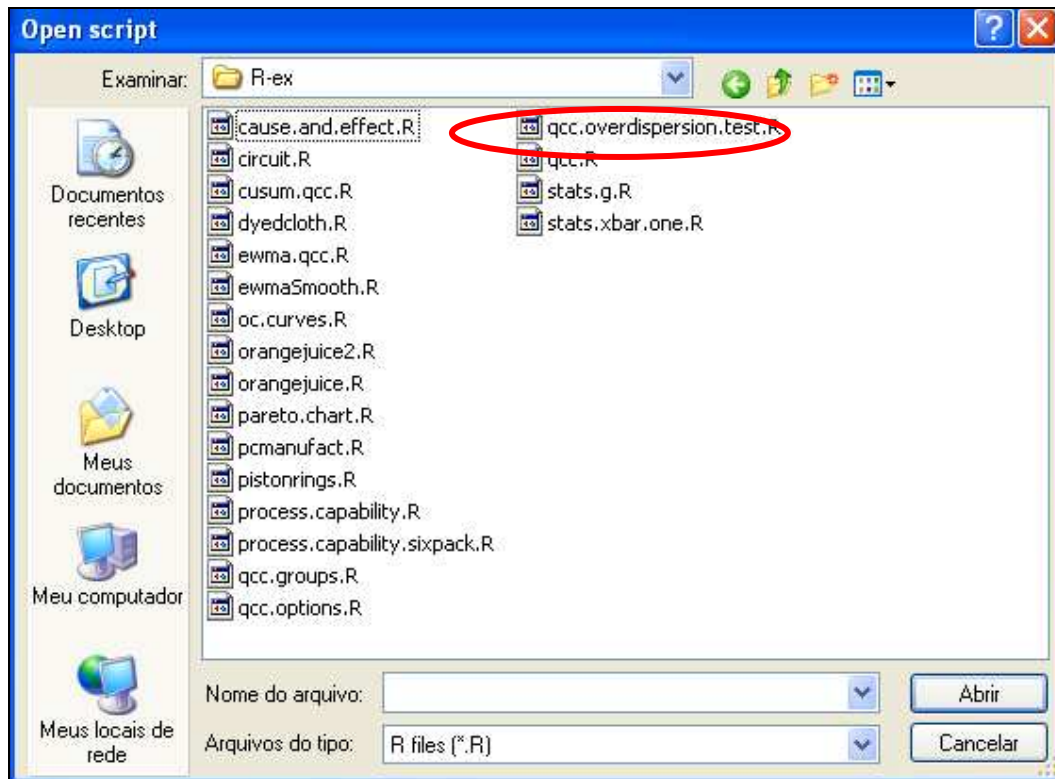
Ao clicar em um dos pacotes, será aberta uma seguinte janela com as opções de recurso de uso do pacote selecionado. Para exemplificar, nesta apostila, iremos selecionar o pacote “qcc”. Observe:



Após clicar no pacote, obtêm-se as opções a seguir:



Destas opções, esta apostila apresentará exemplificação de como utilizar a função “R-ex”, conforme a explicação abaixo:



Após abrir a janela acima, o usuário deve selecionar umas das opções apresentadas clicando duas vezes em cima do nome. Será aberta uma janela onde haverá uma exemplificação da utilização do pacote em pesquisa, com apresentação dos comandos necessários para a execução do pacote.

Observação: Para testar tal exemplo no R, é preciso antes, carregar o pacote, conforme será explicado logo à frente nesta apostila.

Clique duas vezes no nome de uma das opções apresentadas na janela “R-ex”. Como exemplo, nesta apostila, vamos mostrar o processo após escolher o comando “qcc.overdispersion.test.R”, que se trata do comando sobre teste de dispersão. Será aberta a seguinte janela com os comandos das funções do pacote:

```
R C:\Arquivos de programas\R\R-2.8.1\library\qcc\R-ex\qcc.overdispersion.test.R - Edito...
### Name: qcc.overdispersion.test
### Title: Overdispersion test for binomial and poisson data
### Aliases: qcc.overdispersion.test

### ** Examples

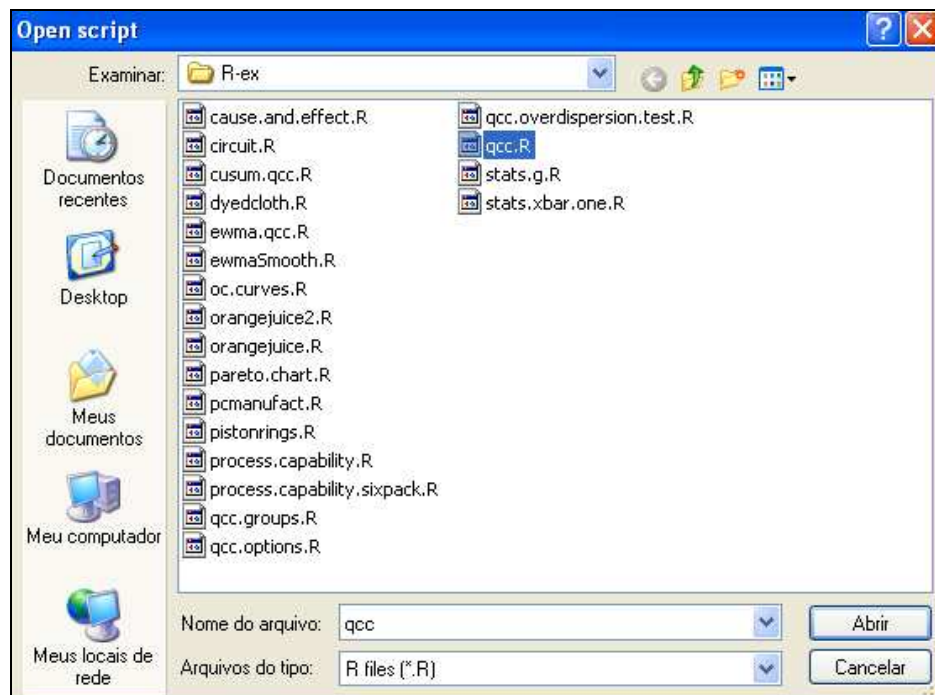
# data from Wetherill and Brown (1991) pp. 212--213, 216--218:
x <- c(12,11,18,11,10,16,9,11,14,15,11,9,10,13,12,
      8,12,13,10,12,13,16,12,18,16,10,16,10,12,14)
size <- rep(50, length(x))
qcc.overdispersion.test(x,size)

x <- c(11,8,13,11,13,17,25,23,11,16,9,15,10,16,12,
      8,9,15,4,12,12,12,15,17,14,17,12,12,7,16)
qcc.overdispersion.test(x)
```

Os comandos apresentados nesta janela, são os comandos que devem ser digitados no console do R para realizar o teste de dispersão com base no banco de dados em pesquisa. Repare que o banco de dados utilizado na janela acima, é um banco de dados aleatório, apenas para exemplificar os comandos. Na execução no R, deve-se alterar apenas o banco de dados.

Para testar a função, basta carregar o pacote no R, selecionar o texto da janela aberta, clicar com o botão direito do mouse e clicar em “Executar linha ou seleção Ctrl+R”, que o programa indicará a solução ou clicar em “copiar” e no console do R em “colar”. Observe outro exemplo com o pacote qcc:

Na janela do “R-ex” clique em “qcc”, como segue:



Será aberta a seguinte janela:

```

R C:\Arquivos de programas\R\R-2.8.1\library\qcc\R-ex\qcc.R - Editor R
### Name: qcc
### Title: Quality Control Charts
### Aliases: qcc print.qcc summary.qcc plot.qcc
### Keywords: htest hplot

### ** Examples

data(pistonrings)
attach(pistonrings)
diameter <- qcc.groups(diameter, sample)

qcc(diameter[1:25,], type="xbar")
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,])
q <- qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], plot=FALSE)
plot(q, chart.all=FALSE)
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], nsigmas=2)
qcc(diameter[1:25,], type="xbar", newdata=diameter[26:40,], confidence.level=0.

qcc(diameter[1:25,], type="R")
qcc(diameter[1:25,], type="R", newdata=diameter[26:40,])

qcc(diameter[1:25,], type="S")
qcc(diameter[1:25,], type="S", newdata=diameter[26:40,])

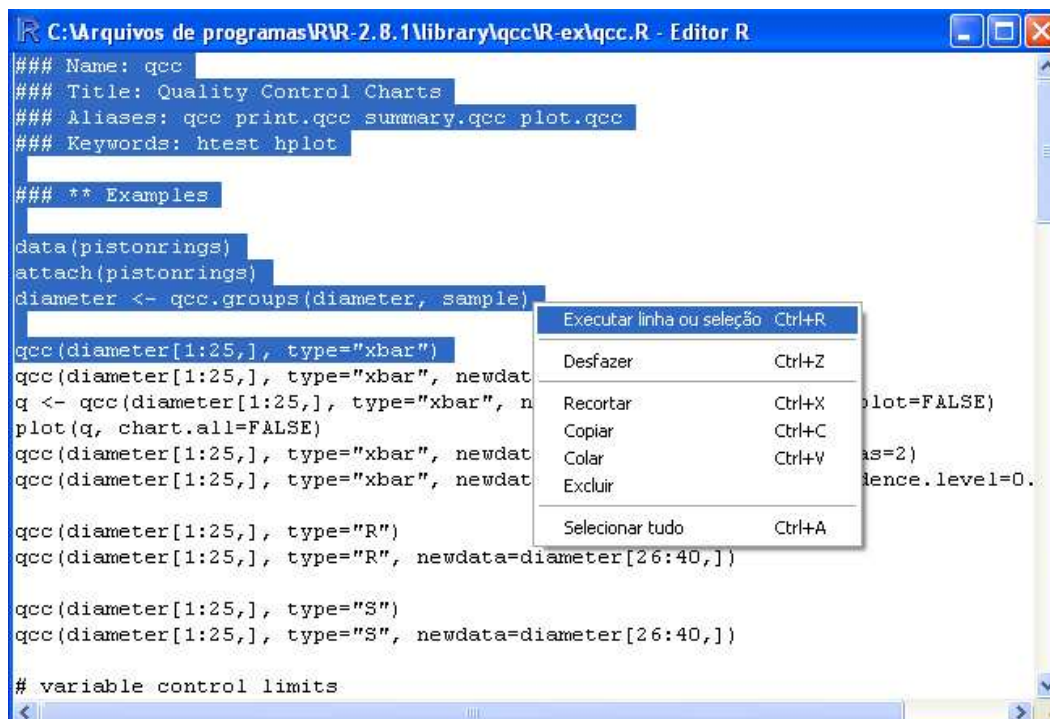
# variable control limits

```

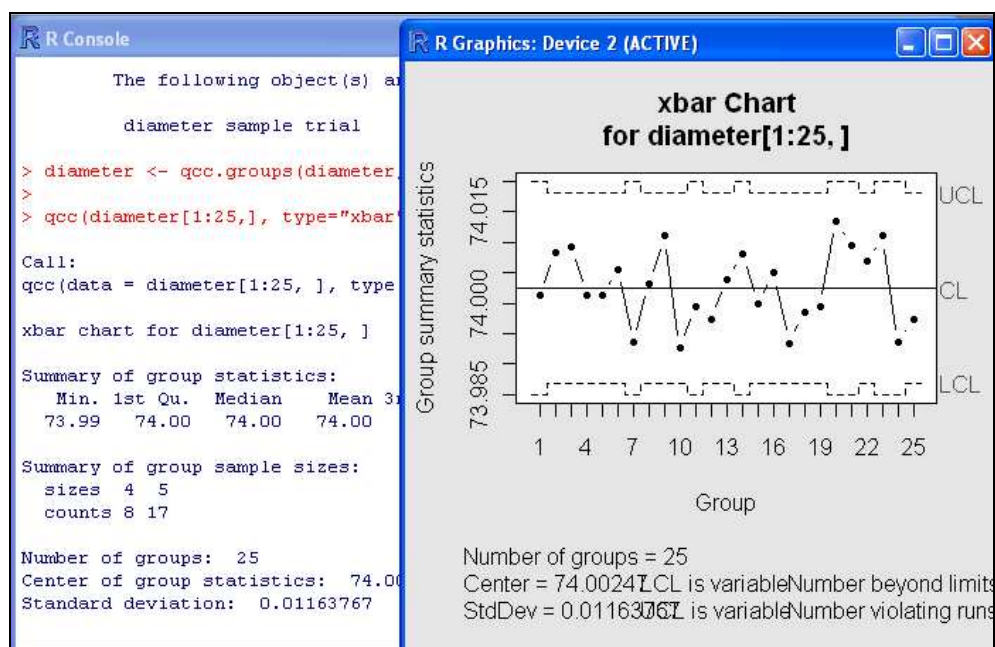
Para exemplificar, digitaremos “library(qcc)” no console do R para ativar o pacote (em caso de dúvida, vide seção seguinte).

Em seguida, selecionaremos parte do texto e copiaremos no R para testar o que executa cada comando. Com a parte do texto selecionada e com o ponteiro

posicionado sobre ele, pressione o botão direito do mouse para abrir um menu exclusivo, conforme ilustrado a seguir:



Após clicar em “Executar linha ou seleção Ctrl+R”, automaticamente aparecerão os comandos no console do R e neste caso, como se trata de uma criação de gráfico, também será aberta uma janela com o gráfico pronto como segue:



Carregando Pacotes:

Após a etapa de instalação (a partir do próprio R ou do site), para usar o pacote, basta carregá-lo da seguinte maneira:

Digite no console do R:

```
>library(nome do pacote) <ENTER>
```

Ou siga as instruções abaixo:

Clique em “pacotes” no menu.

Clique em “carregar pacote(s)”.

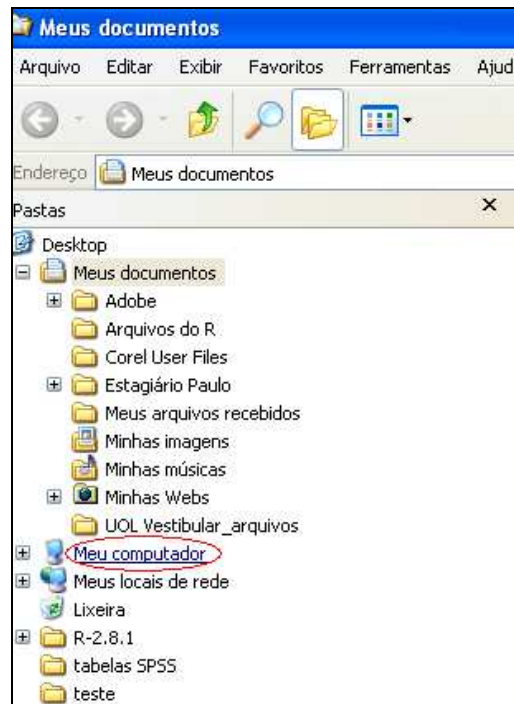
Selecione o(s) pacote(s) desejado(s) e clique em “OK”.

Há pacotes que apenas são ativados no R ,então aparecerá um texto no console indicando que as funções do pacote estão prontas para serem usadas. E há pacotes que ao serem ativados, será aberta uma outra janela onde deverão ser dados os comandos para a realização das funções, como é o caso do pacote Rcmdr (R Commander) que será estudado a seguir:

Pesquisando técnicas ou métodos implementados em pacotes do R

É possível descobrir se existe e qual o nome do pacote que utiliza uma determinada técnica estatística que se queira aplicar. Para isso, devem-se seguir as instruções de busca abaixo:

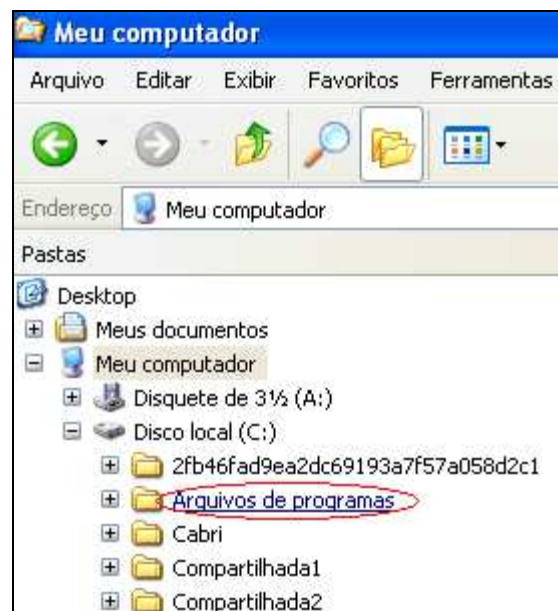
No menu “iniciar” do computador, entre em “Windows Explorer” e clique em “Meu computador”:



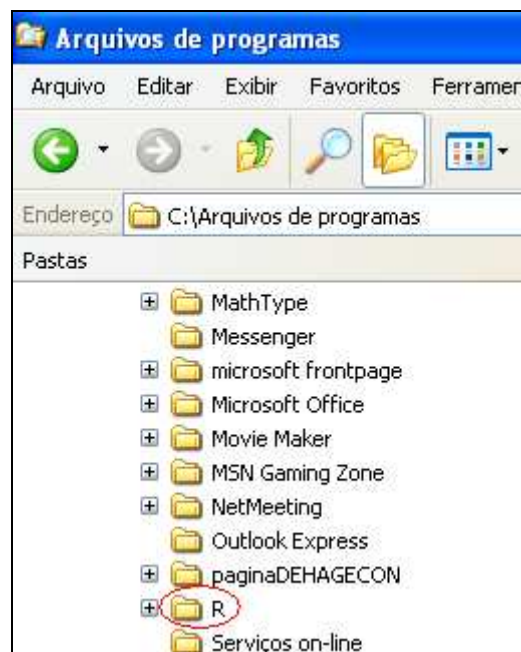
Clique em “Disco local (C:)”:



Clique em “Arquivos de programas”:



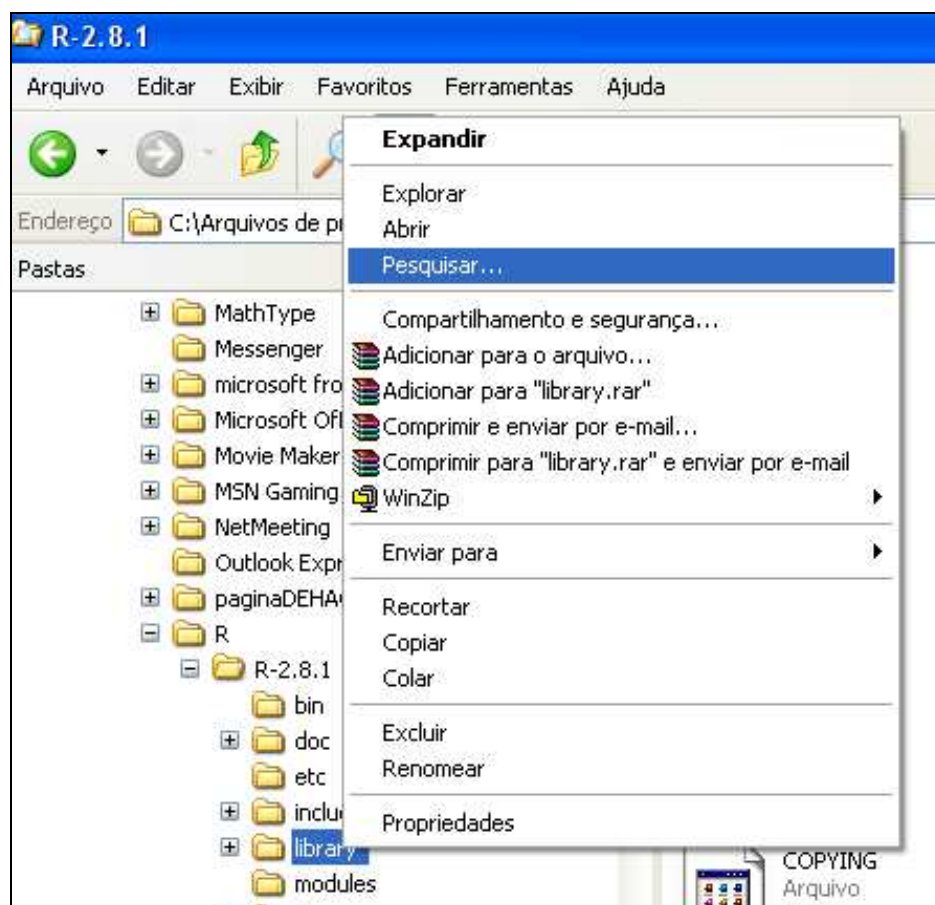
Clique na pasta “R”:



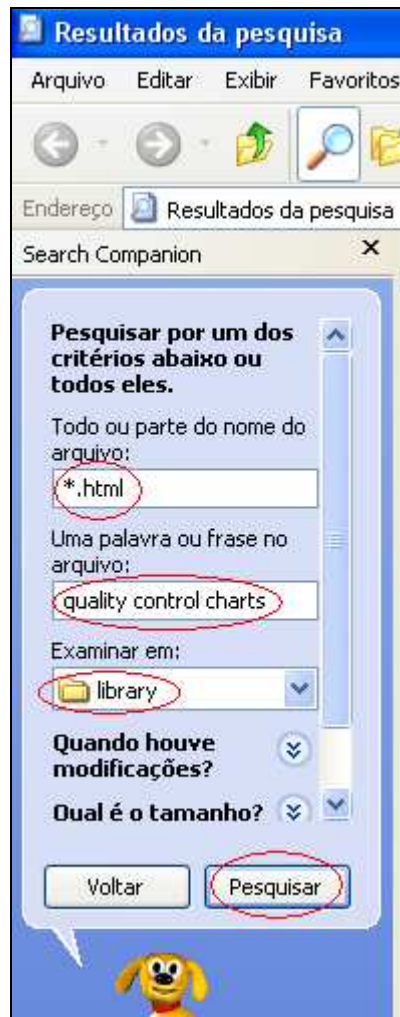
Clique na versão baixada do R:



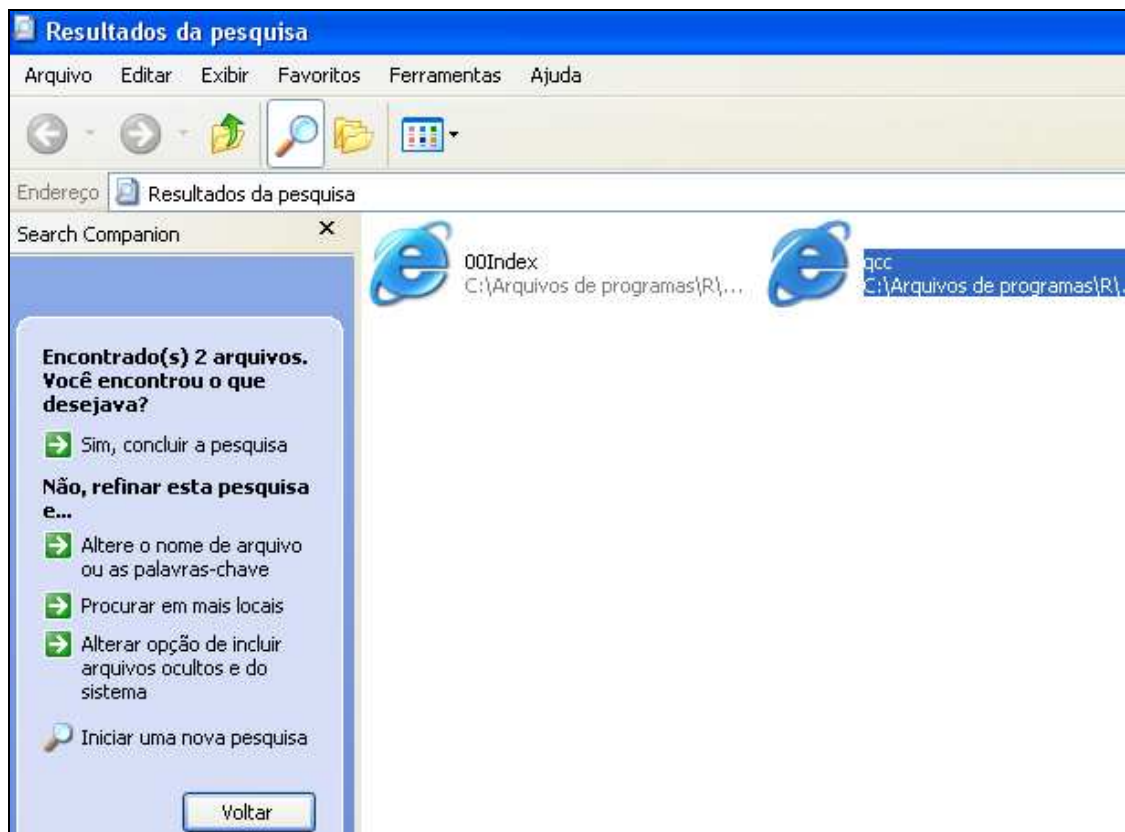
Em seguida, clique com o botão direito do mouse em “library” e em “Pesquisar...”:



Após esta etapa, será aberta uma janela. Maximize a janela e preencha a lacuna e clique em “Pesquisar”, conforme a indicação abaixo. Neste exemplo, utilizaremos como técnica estatística em pesquisa, Controle de Qualidade (quality control charts):



Ao concluir a pesquisa, clique no pacote resultante da pesquisa. Os arquivos que contêm “00” no início do nome, são arquivos HTML usualmente abertos pelo navegador Explorer do Windows, que ao serem abertos, resultam numa página que contém todas as informações sobre as funções do pacote em pesquisa. Neste exemplo, estamos estudando o pacote “qcc”:



Ao clicar duas vezes no nome do pacote, será aberta uma página no navegador padrão da Internet explicando as funções do pacote em pesquisa, com a descrição do pacote, os comandos de uso e com exemplos de aplicação das funções do pacote.

No pacote em estudo, foi aberto o seguinte link:

C:\Arquivos de programas\R\R-2.8.1\library\qcc\html\qcc.html

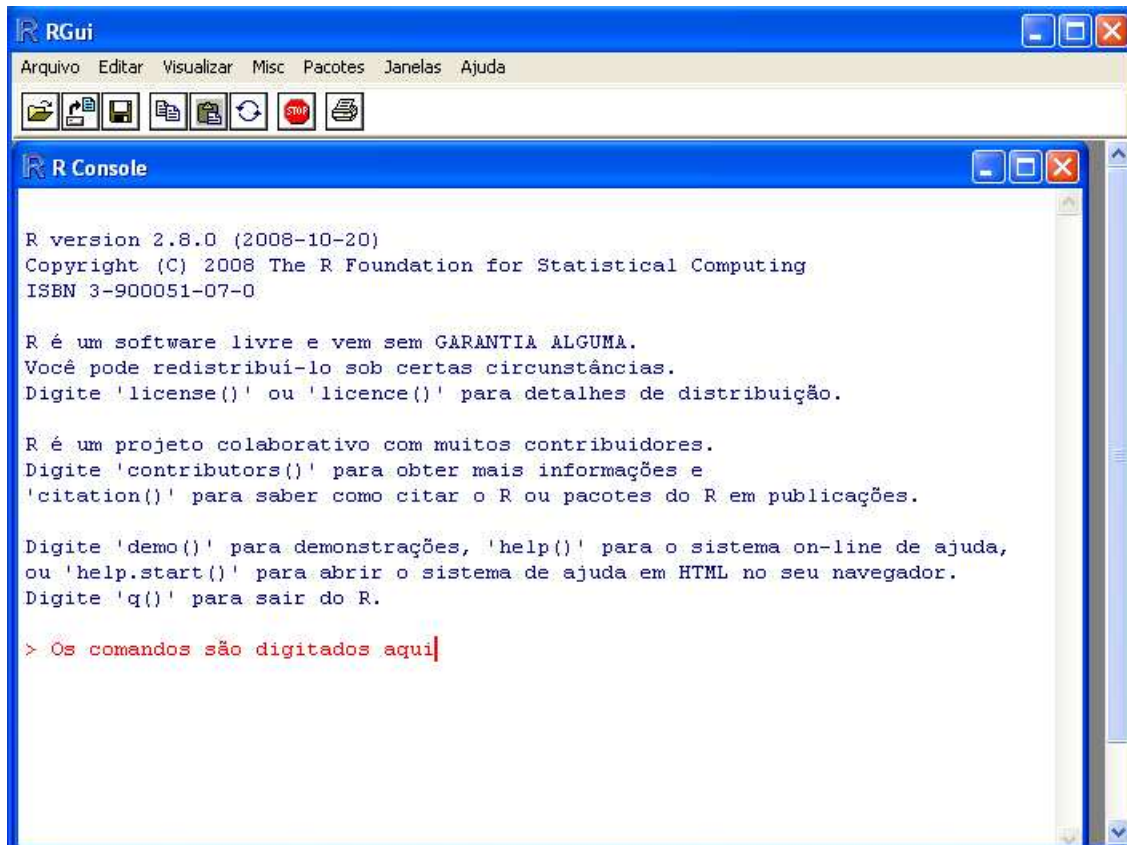
• Introdução ao ambiente de programação

R é uma linguagem e ambiente para computação estatística e para gráficos. Trata-se de um conjunto integrado de facilidades de software para manipulação de dados, cálculo e visualização gráfica. É um projeto disponível como Software Livre sob os termos da Licença Pública Geral GNU da Free Software Foundation na forma de código fonte, ou seja, um projeto com o objetivo de criar um sistema computacional totalmente livre, que qualquer pessoa teria direito de usar, modificar e redistribuir, o programa, seu código fonte, desde que garantindo para todos os mesmos direitos.

Sendo um Software Livre, os códigos fontes do R estão disponíveis e atualmente são gerenciados por um grupo chamado o *Core Development Team* (<http://www.r-project.org/contributors.html>). A vantagem de ter o código aberto é que falhas podem ser detectadas e corrigidas rapidamente e atualizações para Softwares Livres podem ser disponibilizadas em uma questão de dias. Esse sistema de revisão depende pesadamente da

participação dos usuários dispersos, porém conectados à WEB, no mundo inteiro. Em contraste, em vários pacotes comerciais, algumas falhas permanecem sem correção até o próximo lançamento que pode levar muito tempo, com custo adicionais.

R fornece uma interface de entrada por linha de comando (ELC), denominada R Console, ilustrada abaixo. A linguagem da R se torna clara e simples e a flexibilidade da ELC permite que uns poucos comandos simples sejam juntados para criar funções poderosas. Além disso a transparência das funções e a entrada de dados são altamente didáticas. O usuário é sempre consciente do que foi pedido através da ELC.



A plataforma R fornece uma ampla variedade de técnicas estatísticas, como exemplo: modelagem linear e não linear, testes estatísticos clássicos, análise de séries temporais, classificação e agrupamento. Oferece gráficos bem-desenhados com ótima qualidade para publicação, por poderem ser produzidos incluindo símbolos matemáticos e fórmulas quando necessário. Disponibiliza opções para manipulação, armazenamento e ferramentas intermediárias para análise de dados, além de facilidades gráficas de dados visualizadas na tela ou impressas. Possui um conjunto de operadores para cálculos sobre quadros de dados, principalmente matrizes e uma linguagem de programação bem desenvolvida de forma simples e efetiva que inclui condicionais, alças, funções recursivas definidas pelo usuário, e facilidades para entrada e saída. Além de ser disponível para muitas plataformas incluindo Unix, Linux, Macintosh e Windows.

Ao abrir o programa R, será aberto uma janela de comandos chamada “console” com o símbolo “>” que é chamado “prompt” do R, indicando que o programa está pronto para ser

iniciado e receber os comandos sobre objetos criados pelo usuário ao aplicar as análises estatísticas.

Os valores numéricos ou caracteres podem ser dispostos no R em forma de vetores, matrizes ou data frame.

Os argumentos devem ser separados por vírgula e as casas decimais são representadas por ponto. Acentos, espaços e letras maiúsculas têm leituras diferenciadas na linguagem utilizada no R.

O símbolo “#” pode ser inserido no início de uma linha para indicar que o que texto a ser redigido trata-se de um comentário, e portanto, será ignorado pelo R e não ficará salvo na memória como um comando. Sendo este símbolo utilizado mais por uma questão de organização, para por títulos e subtítulos.

É possível ver o histórico de comandos já digitados pressionando a tecla de seta para cima. Pode-se também copiar e colar textos no R. Operações matemáticas simples são feitas de maneira direta, com símbolos de costume usual, podendo o resultado ser indicado diretamente ou atribuído a uma variável. Exemplo:

```
> 2+5 <ENTER>  
[1] 7
```

```
> 3*6 <ENTER>  
[1] 18
```

```
> 8-4 <ENTER>  
[1] 4
```

```
> 10/2 <ENTER>  
[1] 5
```

```
> sin (180) <ENTER>  
[1] 0
```

```
> R<-2+3*4 <ENTER>  
R <ENTER>  
[1] 14
```

- **Principais comandos e funções do módulo ou pacote básico do R:**

Utiliza-se sempre uma variável qualquer seguida do símbolo “<-” para indicar que os dados a serem digitados serão memorizados na variável escolhida e pode-se apresentar os dados entre parênteses:

```
> X<-(83) <ENTER>
> X <ENTER>
[1] 83
```

Ao criar uma seqüência arbitrária, devemos digitar “c” antes dos parênteses para indicar que os dados a serem digitados serão agrupados na forma de vetor:

```
> Y<- c(1,2,3,4,5,9) <ENTER>
> Y <ENTER>
[1] 1 2 3 4 5 9
```

Length => indica o número de termos existentes:

```
> length (Y) <ENTER>
[1] 6
```

NA => indica ausência de dados

NaN => indica que o valor não é um número válido

Factorial(X) => indica o fatorial de X

Seqüência:

```
> X<-1:5 <ENTER>
> X <ENTER>
[1] 1 2 3 4 5
```

OBS: podem-se fazer operações matemáticas simples com seqüências:

```
> X<-X+1 <ENTER>
> X <ENTER>
[1] 2 3 4 5 6
```

Colchetes são usados para extração de elementos especificados:

```
> X [2] <ENTER>
[1] 3
(foi retirado o 2º elemento da seqüência X, o número 3)
```

```
> X [2:4] <ENTER>
[1] 3 4 5
```

```
> X [X<5] <ENTER>
[1] 2 3 4
(elementos de X que são menores que 5)
> X [X<= 5] <ENTER>
```



```
[1] 2 3 4 5
```

(elementos de X que são menores ou iguais a 5)

```
>K<- c(1:5,5) <ENTER>
```

```
>K <ENTER>
```

```
[1] 1 2 3 4 5 5
```

```
>K [K == 5] <ENTER>
```

```
[1] 5 5
```

(valores de K que são iguais a 5)

$Y \sim X \Rightarrow$ indica Y em função de X.

“ ” \Rightarrow delimitam um caractere.

‘Inf’ ou ‘-inf’ \Rightarrow indica que o valor tende a mais ou menos infinito.

$\text{mean}(X) \Rightarrow$ indica a média de X.

$\text{var}(X) \Rightarrow$ indica a variância de X.

$\text{sd}(X) \Rightarrow$ indica o desvio padrão.

$\text{max}(X) - \text{min}(X) \Rightarrow$ indica a amplitude total.

$\text{Range}(X) \Rightarrow$ indica o menor e o maior valor de X, respectivamente.

$\text{tapply}(X,Y,\text{mean}) \Rightarrow$ indica a média de X para cada nível de Y.

OBS: neste comando, pode-se usar qualquer medida, usou-se ‘mean’ apenas como exemplo.

$\text{median}(X) \Rightarrow$ indica a mediana de X.

$\text{quantile}(X) \Rightarrow$ indica ao quartis de X.

$\text{table}(X) \Rightarrow$ fornece as frequências dos valores de X.

$\text{subset}(\text{table}(X), \text{table}(X) == \text{max}(\text{table}(X))) \Rightarrow$ indica a moda de X.

$\text{summary} \Rightarrow$ obtêm-se resumo de dados estatísticos.

?nome da função \Rightarrow para obter informações de ajuda sobre a função desejada.

$\text{help}(\text{nome do pacote}) \Rightarrow$ para obter informações de ajuda sobre o pacote desejado.

`library(pacote)` => carrega ou ativa o pacote desejado (o uso de pacotes será explicado mais à frente).

`rm(objeto)` => remove o objeto desejado da memória do R.

`factor(objeto)` => transforma o objeto escolhido em fator, isto é, o divide em níveis.

`attach(variável)` => reconhece como objeto isolado cada coluna de um Data Frame.

`detach(variável)` => desfaz a função anterior, é aconselhável para evitar conflitos de memória no R.

`fix(K)` => edita K, podendo este ser vetor, matriz ou data frame.

`example(nome da função)` => exibe exemplos da função escolhida.

`demo(package="nome do pacote")` => informa os nomes dos demos (*scripts* que contêm demonstrações, demos, de funcionalidades do pacote), caso existam para o pacote especificado, o qual deve estar ativo e possuir demonstrações durante este comando.

`demo(nome do demo,package="nome do pacote")` => executa o demo especificado.

`choose(n,k)` => indica combinação de n tomado de k a k.

`sqrt(X)` => indica raiz quadrada de X.

1. Comandos usados para criar vetores:

`rep(5,n)` => vetor que contém n vezes o número 5. Exemplo:

```
> rep(7,3) <ENTER>
```

```
[1] 7 7 7
```

`seq(a,b,c)` => cria o intervalo [a,b] separado em c unidades. Exemplo:

```
> seq(1,10,2)
```

```
[1] 1 3 5 7 9
```

`gl(i,r)` => cria um vetor representado pela seqüência de 1 até i com r repetições em cada nível. Exemplo:

```
> gl(4,3)
```

```
[1] 1 1 1 2 2 2 3 3 3 4 4 4
```

```
Levels: 1 2 3 4
```

rank(V) => exibe a posição de cada valor do vetor V ordenado.

sort(V) => exibe os valores do vetor V em ordem crescente.

2. Comandos usados para criar matrizes:

Primeiramente, deve-se criar o código o qual iremos atribuir à matriz.

Em seguida, deve-se criar o vetor que indicará cada item das linhas e colunas da matriz.

Por último, indicar o número de linhas e o número de colunas, respectivamente, que a matriz terá.

OBS: o número de linhas e de colunas, deve ser múltiplo ou sub-múltiplo do número de elementos do vetor.

Exemplo:

```
>M1<-matrix(c(1,3,5,7,9,11),nrow=2,ncol=3) <ENTER>
```

```
>M1 <ENTER>
```

```
      [,1] [,2] [,3]  
[1,]  1   5   9  
[2,]  3   7  11
```

Pode-se também criar o vetor isolado da matriz. Veja:

```
>V<-1:6 <ENTER>
```

```
>V <ENTER>
```

```
[1] 1 2 3 4 5 6
```

```
>M2<-matrix(data=V,nrow=3,ncol=2) <ENTER>
```

```
>M2 <ENTER>
```

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

Pode-se também escrever de forma direta a matriz, ocultando “nrow” e “ncol”. Considerando o primeiro valor como o indicador do número de linhas e o segundo como indicador do número de colunas. Observe:

```
>M3<-matrix(c(7,20,9,212,1,2,3,4,15,23,65,77),4,3) <ENTER>
```

```
>M3 <ENTER>
```

```
      [,1] [,2] [,3]  
[1,]  7   1  15
```

```
[2,] 20  2  23
[3,]  9  3  65
[4,] 212  4  77
```

```
>M4<-matrix(1:6,6,2) <ENTER>
```

```
>M4 <ENTER>
```

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

O R tem como padrão seguir a ordem dos dados por colunas, mas pode-se mudá-la para linhas, veja o exemplo utilizando ainda M4:

```
>M4<-matrix(1:6,6,2,byrow=T) <ENTER>
```

```
M4 <ENTER>
```

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

Pode-se retirar apenas parte da matriz. Exemplo:

```
>M1[1,2] <ENTER>
```

```
[1] 5
```

```
>M4[,2] <ENTER>
```

```
[1] 2 4 6 2 4 6
```

```
>M2[1,] <ENTER>
```

```
[1] 1 4
```

As funções “cbind” e “rbind” adicionam coluna e linha, respectivamente. Podendo estas serem adicionadas no final ou no início da matriz. Veja:

```
>m1<-cbind(M1,c(7,8)) <ENTER>
```

```
>m1 <ENTER>
```

```
      [,1] [,2] [,3] [,4]
[1,]    1    5    9    7
```

```
[2,] 3 7 11 8
```

```
>m1<-rbind(c(7,8,9,10),m1) <ENTER>
```

```
>m1 <ENTER>
```

```
      [,1] [,2] [,3] [,4]  
[1,]  7   8   9  10  
[2,]  1   5   9   7  
[3,]  3   7  11   8
```

Multiplicação de matrizes:

```
>M1%*%M2 <ENTER>
```

```
      [,1] [,2]  
[1,] 38  83  
[2,] 50 113
```

Outros comandos:

Considerando uma matriz M qualquer:

fix(M) => abre uma janela da matriz M editada, podendo fazer alterações na matriz.

solve(M) => calcula a inversa da matriz M.

t(M) => calcula a transposta da matriz M.

det(M) => calcula o determinante da matriz M.

3. Comandos usados para criar Data Frame:

Um Data frame pode ser criado através da entrada de dados externos ou pode ser criado no próprio R, adotando cada coluna do Data Frame como um vetor.

Ao criar os vetores, pode-se usar a função `c()`, já conhecida, ou a função `scan()` da seguinte maneira:

```
> X<-scan()<ENTER>
```

```
1: 2 <ENTER>
```

```
2: 3 <ENTER>
```

```
3: 4 <ENTER>
```

```
4: 5 <ENTER>
```

```
5: <ENTER>
```

```
Read 4 items
```

```
>X <ENTER>
```

```
[1] 2 3 4 5
```

```
> Y<-c("A","B","C","D") <ENTER>
```

```
> Y <ENTER>
```

```
[1] "A" "B" "C" "D"
```

(foram criados os vetores X e Y utilizando a função scan())

Após criar os vetores desejados, basta juntá-los em um Data Frame e apagar os vetores criados da memória, pois eles já estarão armazenados no Data Frame.

```
>D1<-data.frame(X,Y) <ENTER>
```

```
>rm(X,Y) (para remover os vetores X e Y) <ENTER>
```

```
>D1 <ENTER>
```

```
  X Y
```

```
1 2 A
```

```
2 3 B
```

```
3 4 C
```

```
4 5 D
```

É possível alterar o nome das colunas criadas no Data Frame através do seguinte comando:

```
>names(D1)<-c("coluna1","coluna2") <ENTER>
```

```
>attach(D1) <ENTER>
```

(para poder usar as colunas criadas apenas com o nome delas, veja:)

```
>coluna1 <ENTER>
```

```
[1] 2 3 4 5
```

É possível isolar colunas também com o seguinte comando:

```
D1$(nome da coluna)
```

No caso, D1\$coluna1

OBS: é importante ressaltar a diferença entre Data Frame e matriz. No Data Frame é possível trabalhar com valores numéricos e caracteres ao mesmo tempo e também pode-

se alterar o nome de suas colunas, podendo ter acesso a cada uma delas separadamente pela especificação deste nome.

Entrada de dados externos, exemplo:

Considerando “Tabela1” o nome de uma tabela qualquer criada no Excel, salva em CSV. Deve-se fazer o seguinte processo:

- 1) Ir ao menu “Arquivo”, “mudar diretório” e selecionar a pasta na qual o arquivo foi salvo.
- 2) Digitar: `Y<-read.table(“Tabela1.CSV”,header=T,dec=”,”,sep=”,;”)`

Digitando “Y” no console, será aberta a Tabela1.

- **Gráficos:**

Cada tipo de gráfico possui um comando específico, porém quase todos terão suas configurações controladas pelo comando “par” em função de argumentos que são utilizados para a criação e manipulação de gráficos. São eles:

mfrow	divide a janela onde os gráficos serão construídos, cujo valor é do tipo c(nl, nc), em que nl é o número de linhas e nc o número de colunas em que a janela será dividida
mfcoll	idêntica à mfrow, porém seu valor será do tipo c(nc, nl)
ps	controla o tamanho de todos os textos nos gráficos, cujo valor deve ser um número inteiro
pty	indica a área em que o gráfico será construído, seus valores são: "m" (área máxima) ou "s" (área quadrada)
bg	controla a cor de fundo da janela dos gráficos, sendo que as 657 cores disponíveis são visualizadas por meio do comando colors()
fg	controla a cor dos eixos e das bordas dos símbolos dos gráficos
col.main, col.lab, col.sub, col.axis	controla as cores do título, dos nomes dos eixos, do rodapé e dos valores dos eixos, respectivamente
cex.main, cex.lab, cex.sub, cex.axis	controla o tamanho da fonte, do título, dos nomes dos eixos, do rodapé e dos valores dos eixos, respectivamente, sendo que os valores positivos menores ou maiores do que 1, diminuem ou aumentam o tamanho, respectivamente
font.main, font.lab, font.sub, font.axis	controla a fonte a ser usada, com base em números inteiros de 1 a 20, sendo que o número 1 indica texto normal, o 2 negrito, o 3 itálico e o 4 negrito + itálico
adj	indica a posição dos textos através de valores de 0 a 1, sendo que 0 indica texto à esquerda, 0.5 centralizado e 1 à direita
las	indica a posição dos valores nos eixos x e y dos gráficos, sendo o padrão neste material de las=1, que indica que os valores dos eixos x e y devem ser postos na horizontal

É possível copiar os gráficos criados no R para editores de textos como Word. Para isso basta selecionar o gráfico, pressionar 'Ctrl + c' para copiar e já na página do editor pressionar 'Ctrl + v' para colar. Ou clicar com o botão direito sobre o gráfico e depois em 'Copy as Metafile' com o botão esquerdo para copiar e na página do editor, pressionar 'Ctrl + v' para colar o gráfico.

1. Histograma

Serão criadas 2 variáveis:

```
X1<-c(gl(3,4),5,6,7,7) <ENTER>
X2<-c(1,2,3,4,5,6,7,8,9,10,10,11,12,13,14,15) <ENTER>
X<-c(X1,X2) <ENTER>
rm(X1,X2) <ENTER>
hist(X) => criar o gráfico.
```

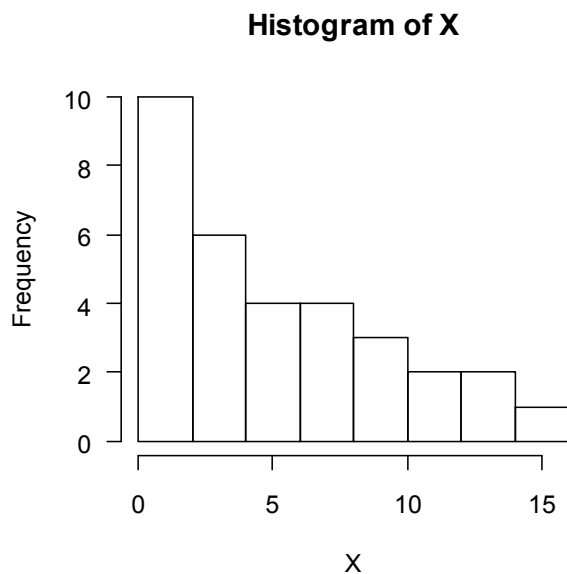
Exemplos de como fazer alterações no gráfico (função 'par'):

```
par(las=1) => para exibir os valores dos eixos x e y.
par(bg='nome da cor') => para escolher a cor de fundo.
par(mfrow=c(1,1)) => para dividir a janela dos gráficos em linhas e colunas, no
caso, em 1 linha e 1 coluna.
hist(X)
```

(Após digitar os comandos da função 'par' deve-se digitar 'hist(X)' de novo para atualizar as alterações)

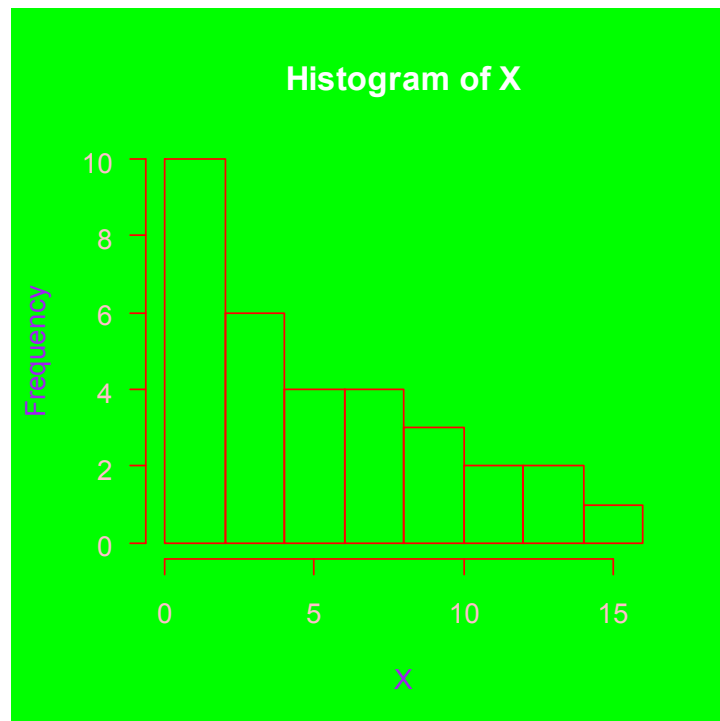
Demonstração:

```
> hist(X) <ENTER>
```



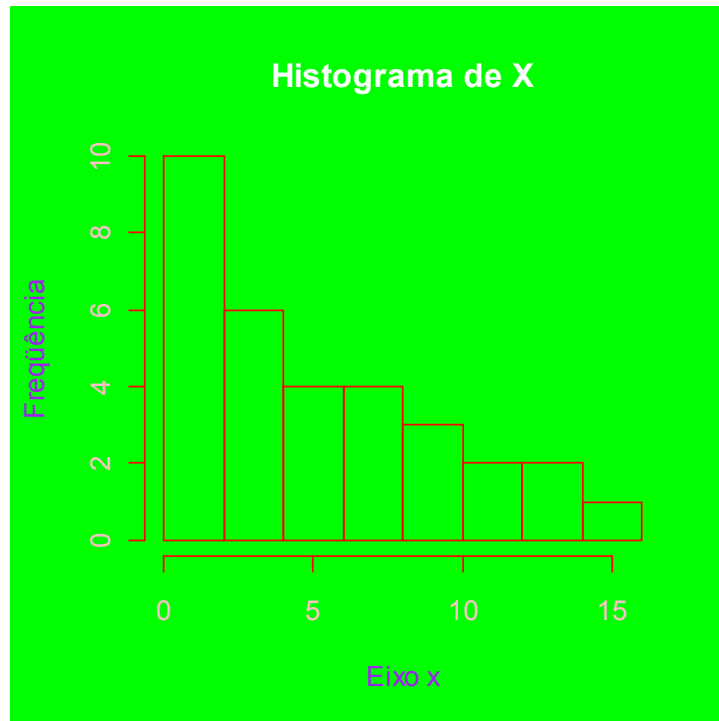
```
> par(bg="green") <ENTER>
> par(fg="red") <ENTER>
```

```
> par(col.main="white") <ENTER>  
> par(col.lab="purple") <ENTER>  
> par(col.axis="pink") <ENTER>  
> hist(X) <ENTER>
```



Para alterar o título do gráfico e o nome dos eixos deve-se seguir o comando abaixo:

```
> hist(X,main="Histograma de X",xlab="Eixo x",ylab="Frequência") <ENTER>
```



2. Gráfico de Ramo e Folhas:

Serão criadas 2 variáveis:

```
> Y1<-rnorm(50,150,20) <ENTER>
> Y2<-rnorm(50,100,20) <ENTER>
> Y<-c(Y1,Y2) <ENTER>
> rm(Y1,Y2) <ENTER>
> stem(Y) => comando para a construção do gráfico
```

The decimal point is 1 digit(s) to the right of the |

```
4 | 59
6 | 335137
8 | 112233356902799
10 | 011233344678993556789
12 | 01134567999022223479
14 | 23001122556667799
16 | 01222335589133469
18 | 0
20 | 4
```

Caso seja necessário estratificar o gráfico, deve-se digitar os comandos abaixo:

```
> stem(Y[X==1]) <ENTER>
```

The decimal point is 2 digit(s) to the right of the |

```
0 | 888
1 | 001112233
1 | 6667789
```

```
> stem(Y[X==2]) <ENTER>
```

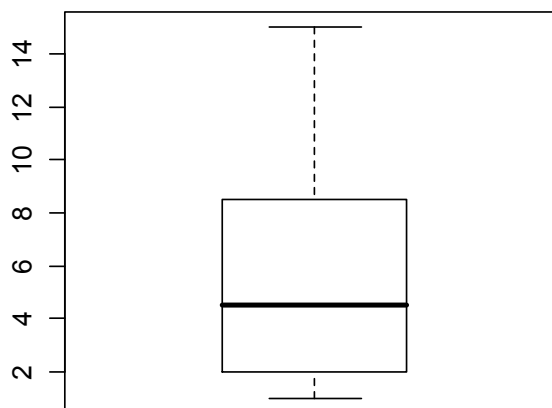
The decimal point is 2 digit(s) to the right of the |

```
0 | 8
1 | 012344
1 | 5556667
2 | 1
```

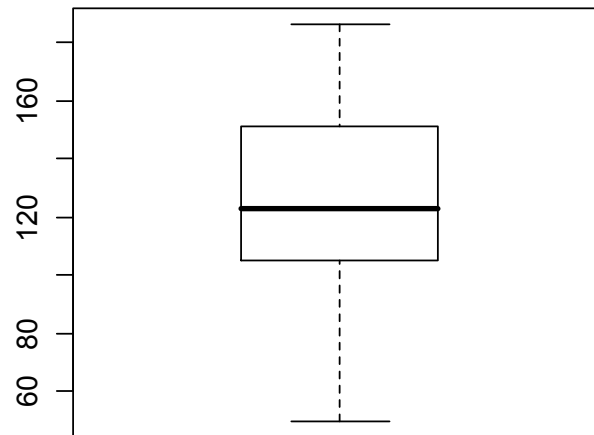
3. Box – plot:

Com base nas variáveis X e Y usadas nos exemplos acima, construiremos o gráfico Box-plot a seguir:

```
> boxplot(X) <ENTER>
```



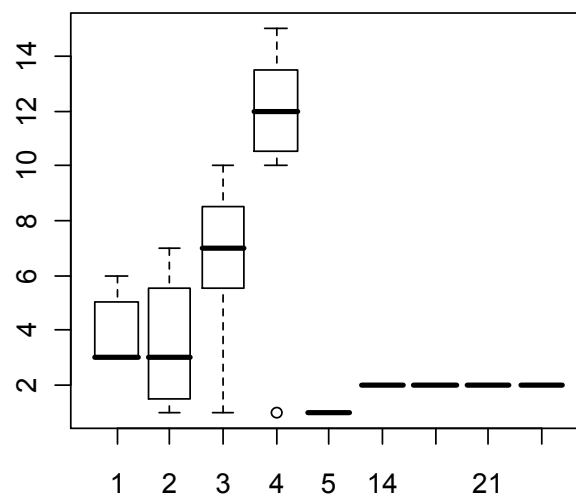
```
> boxplot(Y) <ENTER>
```



Criaremos agora um vetor W com o mesmo tamanho de X:

```
> W<-c(2,3,4,5,21,32,14,15,gl(4,6)) <ENTER>
```

boxplot(X~W) => Box-plot estratificado por W



O pacote R Commander:

Este pacote é um dos mais promissores para a disseminação do uso do ambiente R, por ser uma interface gráfica amigável que possui uma excelente eficácia na importação de dados e na construção de gráficos, além de ter uma execução bem simples. Observe:

Para utilizar as funções do pacote, deve-se carregá-lo no menu do programa, aguardar a janela com os nomes dos pacotes abrir, selecionar “Rcmdr” (R Commander) e clicar em “OK”. Será aberta a janela deste pacote onde deverão ser digitados os comandos para a realização das funções desejadas.

Entrada de dados externos

Para a importação de dados devem ser seguidas as instruções abaixo:

Ir ao menu “Data”.

Clicar em “Import data” .

Clicar na última opção: “from Excel, Access or dBase data set...”.

Digite o código que você deseja utilizar para tal banco de dados no R e dê “ok”.

Abrirá uma janela, na qual deve-se procurar o arquivo que se quer importar e abri-lo.

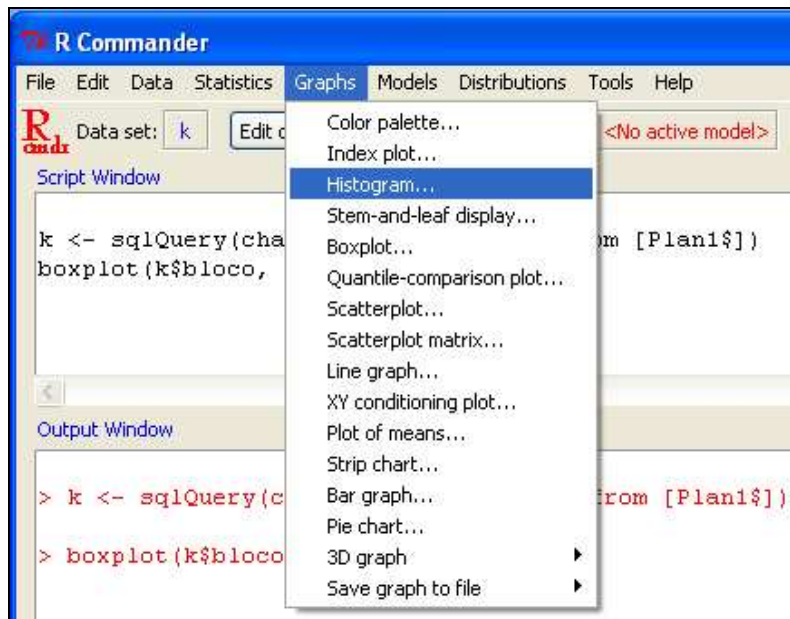
Em seguida, uma outra pequena janela será aberta, clique em “Plan1” e dê “ok”.

Vá à janela de Console do R e digite o código que você designou para a tabela importada.

Será aberto o banco de dados e será permitida a manipulação deste.

É possível criar gráfico a partir deste pacote. Para tal função, após clicar em “Plan1” e dar “OK”, o usuário deve retornar ao menu da janela do R Commander, clicar em “Graphs” e escolher o tipo de gráfico desejado.

Seguindo estas instruções, o gráfico aparecerá em uma nova janela à frente do console do R. Veja:



Após esta etapa basta ir à janela de console do R que o gráfico estará pronto.

Exportação de dados

Após criar um vetor, matriz, ou qualquer objeto do gênero, é possível salvá-lo em um arquivo de texto ou planilha. Para isso basta executar o comando **write.table** como segue abaixo:

```
> write.table(VETOR,file="C:\\pasta_desejada\\nomedoarquivo.txt",sep=" ")
```

Exemplo:

```
> # Calcular a média de um vetor x qualquer com uma poda de 20%
> <ENTER>
> mean(x,trim=0.2)
```

Agora vamos aprender como definir uma função, ou seja, como pré-definir um objeto no R:

Imagine um função y tal que $y = (x+1)^2 / 2x$, vamos pré-definir essa função no R:

```
> Y <- function(x){
+ return( ( ( x+1 )^2 ) / ( 2*x ) )
+ }
```



```
<ENTER>  
> Y(2)  
[1] 2.25
```

Gráfico 3D

Dentro do pacote R Commander, é possível criar gráficos 3D. Para tal, iremos exemplificar através de uma função, conforme segue abaixo:

Considere $Z = X^2 + 2XY - Y$

Para os valores de X variando de 1 até 10 e de Y variando de acordo com a seqüência arbitrária 0, 2, 4, 6, 1, 3, 5, 7, 9 e -4.

Obteremos os seguintes valores de Z : 1, 10, 29, 58, 34, 69, 114, 169, 234 e 24.

No R, após carregar o pacote “Rcmdr” (RCommander) e esperar abrir sua página de comandos, deve-se clicar em “Data” e em seguida em “New Data Set”.

Na janela que abrirá, o usuário deve digitar o nome pelo qual deseja gravar os dados a serem salvos e clicar em “OK”. Neste exemplo, salvaremos como “dados”. Observe:



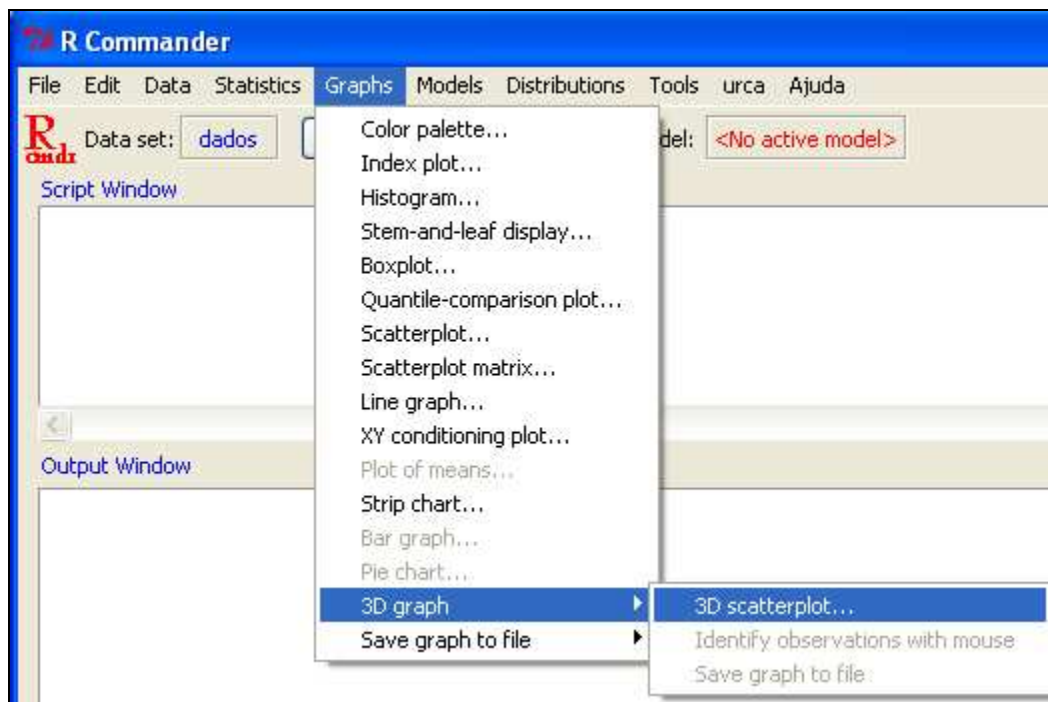
Após dar “OK”, por cima do Console do R, abrirá uma janela com uma tabela, onde o usuário deverá digitar os valores dos pontos que deseja que apareçam no gráfico. Nela, é possível mudar o nome das variáveis, para isso, deve-se clicar em “var1”, “var2”, “var3” e assim sucessivamente conforme for o número de variáveis, esperar abrir uma pequena janela e nela digitar o símbolo ao qual deseja associar cada variável e dar “Enter”. Veja:



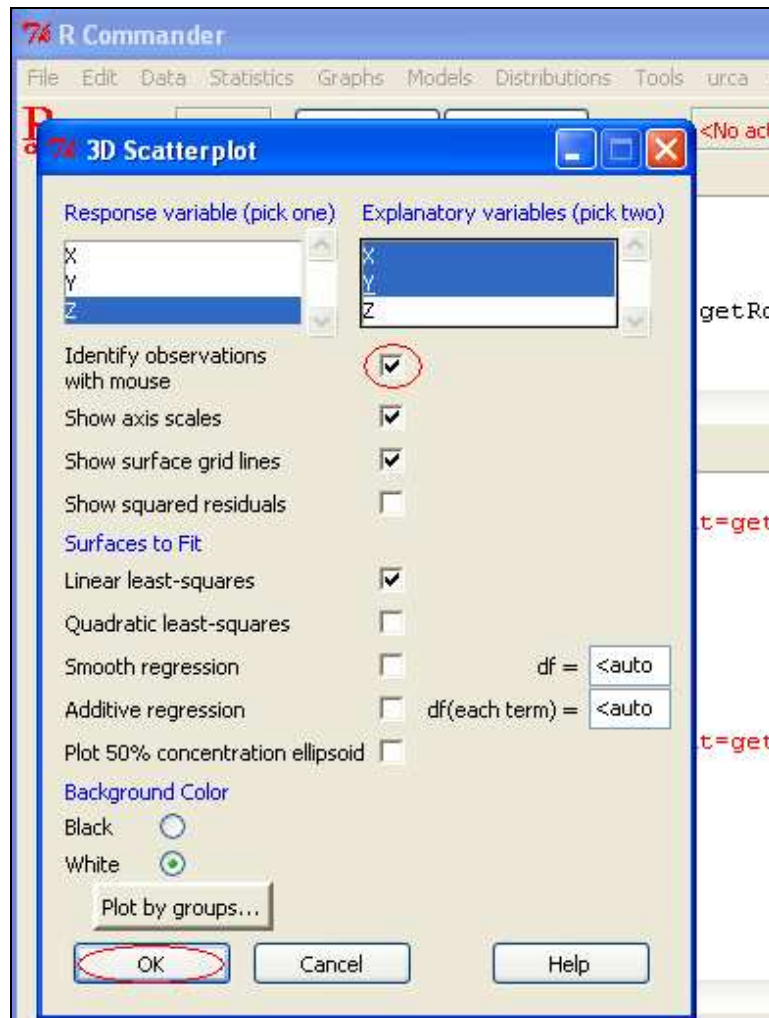
Depois disso, deve-se colocar os valores na tabela. Observe:

	X	Y	Z	var4
1	1	0	1	
2	2	2	10	
3	3	4	29	
4	4	6	58	
5	5	1	34	
6	6	3	69	
7	7	5	114	
8	8	7	169	
9	9	9	234	
10	10	-4	24	
11				

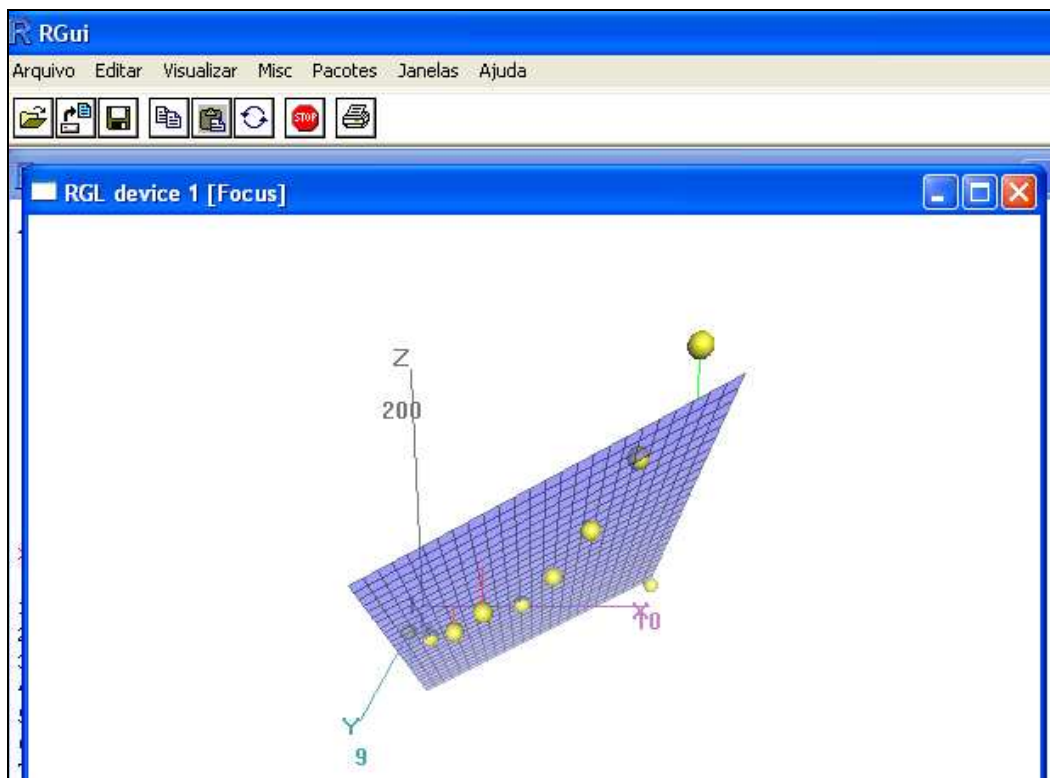
Após montar a tabela com os dados do gráfico, deve-se retornar à janela o R Commander, clicar no menu “Graphs”, “3D Graph” e em “3D scatterplot...”.



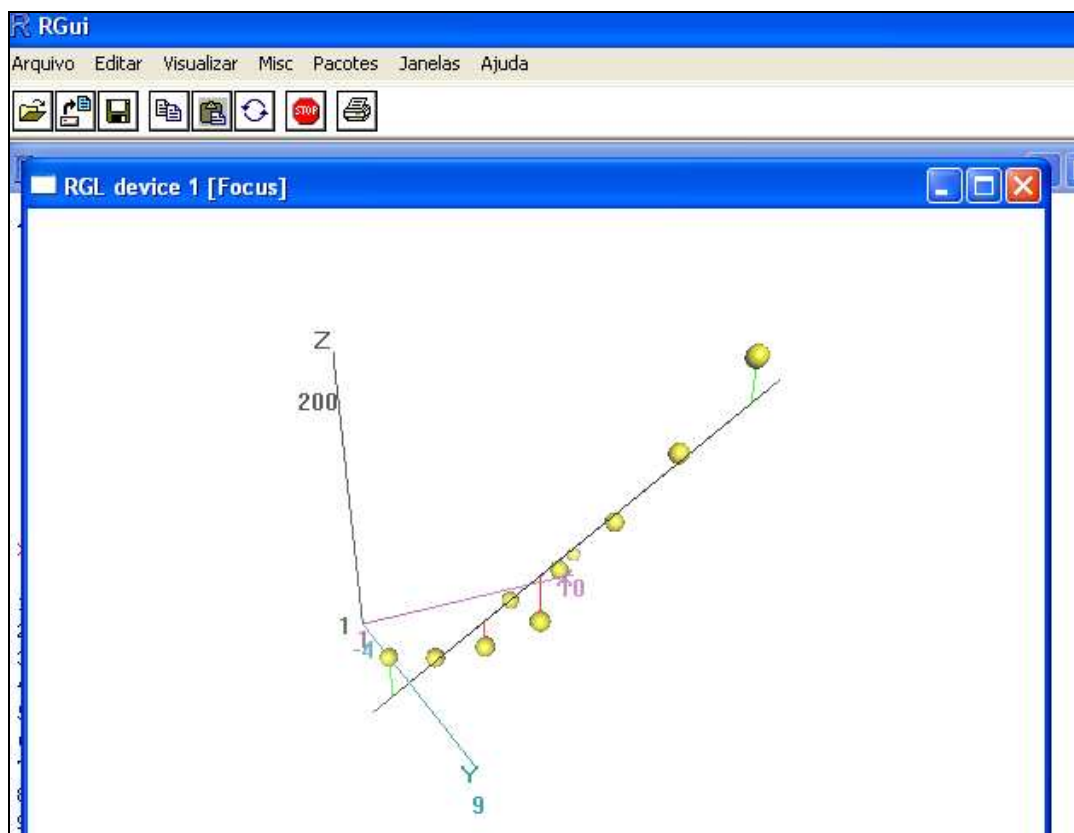
Depois de clicar em “3D scatterplot”, será aberta uma janela, na qual o usuário deve selecionar as variáveis que deseja. Neste exemplo, iremos colocar Z em função de X e Y e selecionaremos também a opção “Identify observations with mouse”, conforme segue abaixo:



Após dar “OK”, uma janela com o gráfico aparecerá por cima do Console do R, como segue:



Com o mouse é possível mexer o gráfico, mudando seu ângulo de observação. Veja:



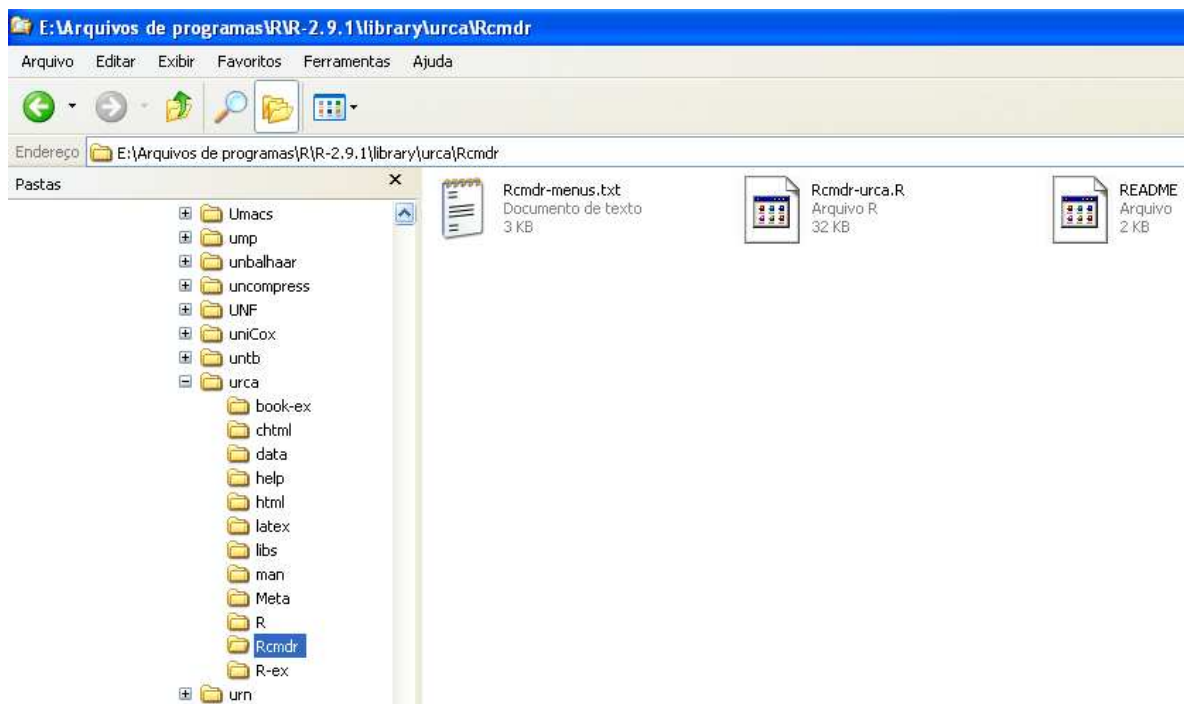
Criando um novo item de menu dentro do R Commander

Nesta seção vamos ilustrar a criação de um item de menu personalizado. Vamos aproveitar para isso de um conjunto de instruções que já vem programado e explicado junto com o pacote URCA. Este pacote serve como apoio ao livro “Analysis of Integrated and Cointegrated Time Series with R (Use R)” de Bernhard Pfaff comercializado pela editora Springer:



Tal como este livro, há uma quantidade grande de livros que estão sendo publicados, com ou sem pacotes auxiliares, versando sobre metodologias, estatísticas ou não, que usam o R (USE R!). Na data da elaboração desta apostila, haviam listados oitenta e cinco livros no link: <http://www.r-project.org/doc/bib/R-books.html> .

Vamos usar o pacote URCA, porque ele contém um subdiretório especial, denominado Rcmdr, conforme podemos visualizar na imagem seguinte:



Quando um pacote contém um subdiretório Rcmdr, isto é indício de que ele pode trabalhar em sintonia com o pacote R Commander. Ao abrir o arquivo README, encontramos instruções sobre como instalar um add-in (suplemento) pré-programado para incluir um item de menu que implementa funções do pacote URCA amigavelmente dentro do R Commander:

```

README - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda

#####
# R E A D M E #
#####

The two files in the '/Rcmdr' directory of urca, namely 'Rcmdr-urca.R'
and 'Rcmdr-menus.txt', can be utilised as an add-in to 'Rcmdr' of
John Fox. For more information on 'Rcmdr' please visit the URL:

    http://socserv.socsci.mcmaster.ca/jfox/Misc/Rcmdr/

as well as the man pages of the package itself. Incidentally, a
tutorial 'Getting-started-with-the-Rcmdr.pdf' is shipped in the
'/doc' subdirectory of 'Rcmdr'.

In order to install the 'urca' add-in to 'Rcmdr' two steps are
necessary:

1. step:
*****
Copy the 'Rcmdr-urca.R' into the '/etc' sub-directory of the package
'Rcmdr'; that is: 'Rcmdr/etc' should now contain this file.

2. step:
*****
Copy the contents of the file 'Rcmdr-menus.txt' in the 'urca/inst'
directory into the file 'Rcmdr/etc/Rcmdr-menus.txt' at a suitable
place. For example, you can place the contents right before the
definition of the 'Help-menu' (look for a line, similar to:

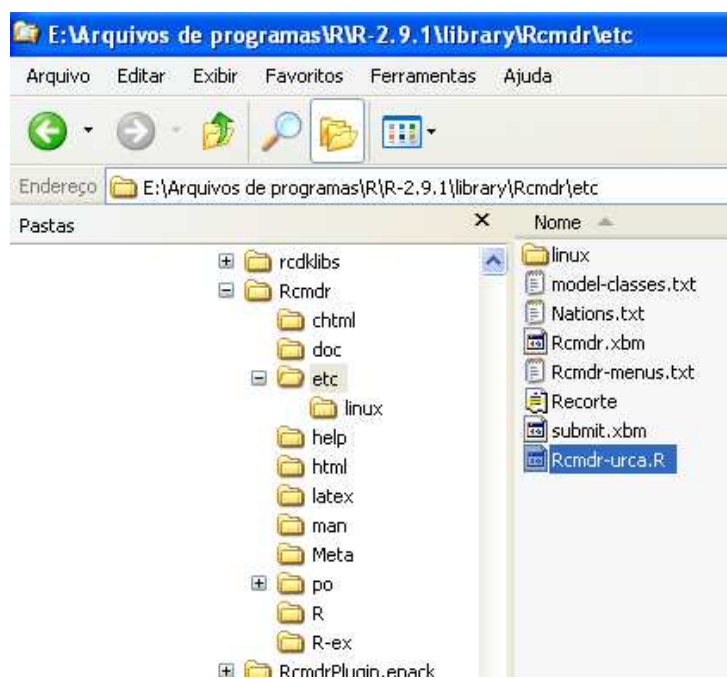
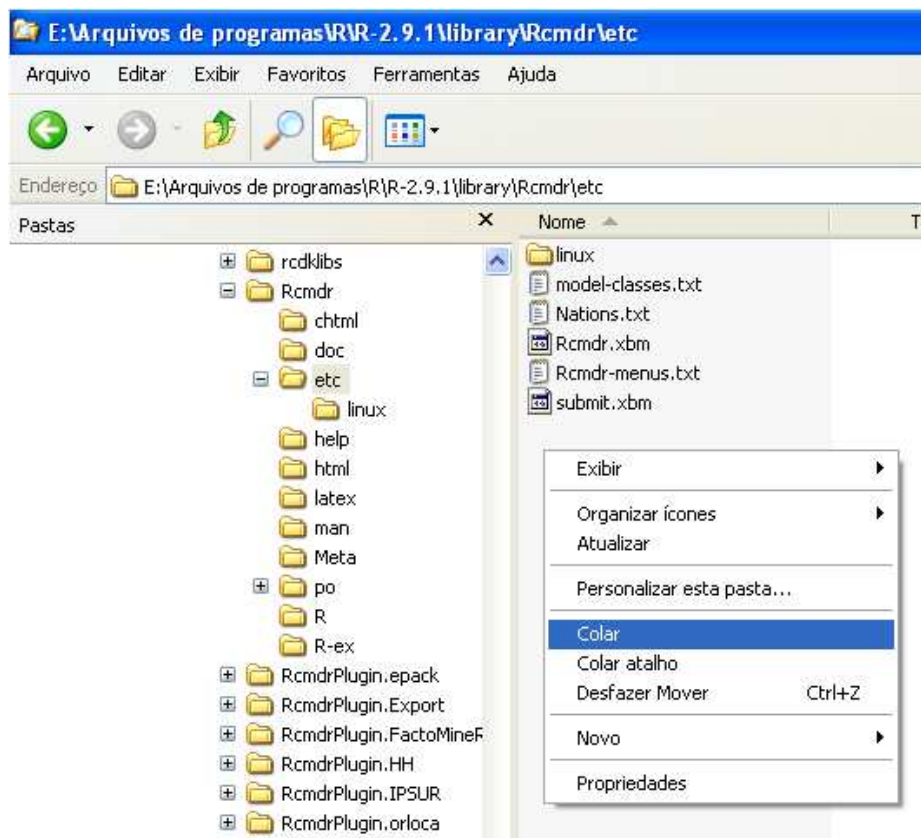
    menu helpMenu topMenu "" "" ""

Then, the newly created 'urca-menu' will be placed left to the
'Help-menu' of 'Rcmdr' once you executed: library(Rcmdr) from a
R Console.

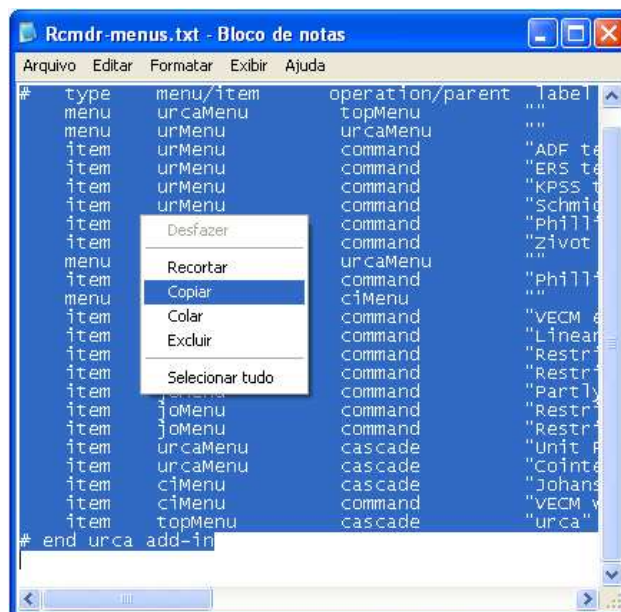
```

Seguindo as instruções contidas neste README, devemos:

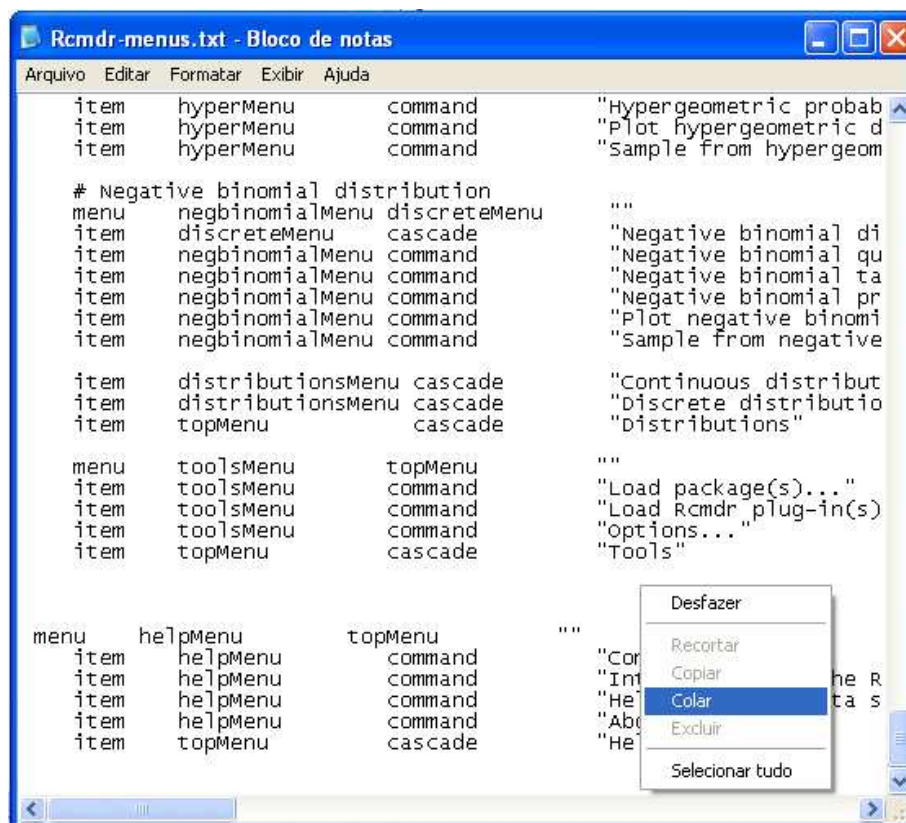
1º passo – copiar o arquivo 'Rcmdr-urca.R' para o subdiretório '/etc' do pacote R Commander, conforme exemplo ilustrado a seguir:



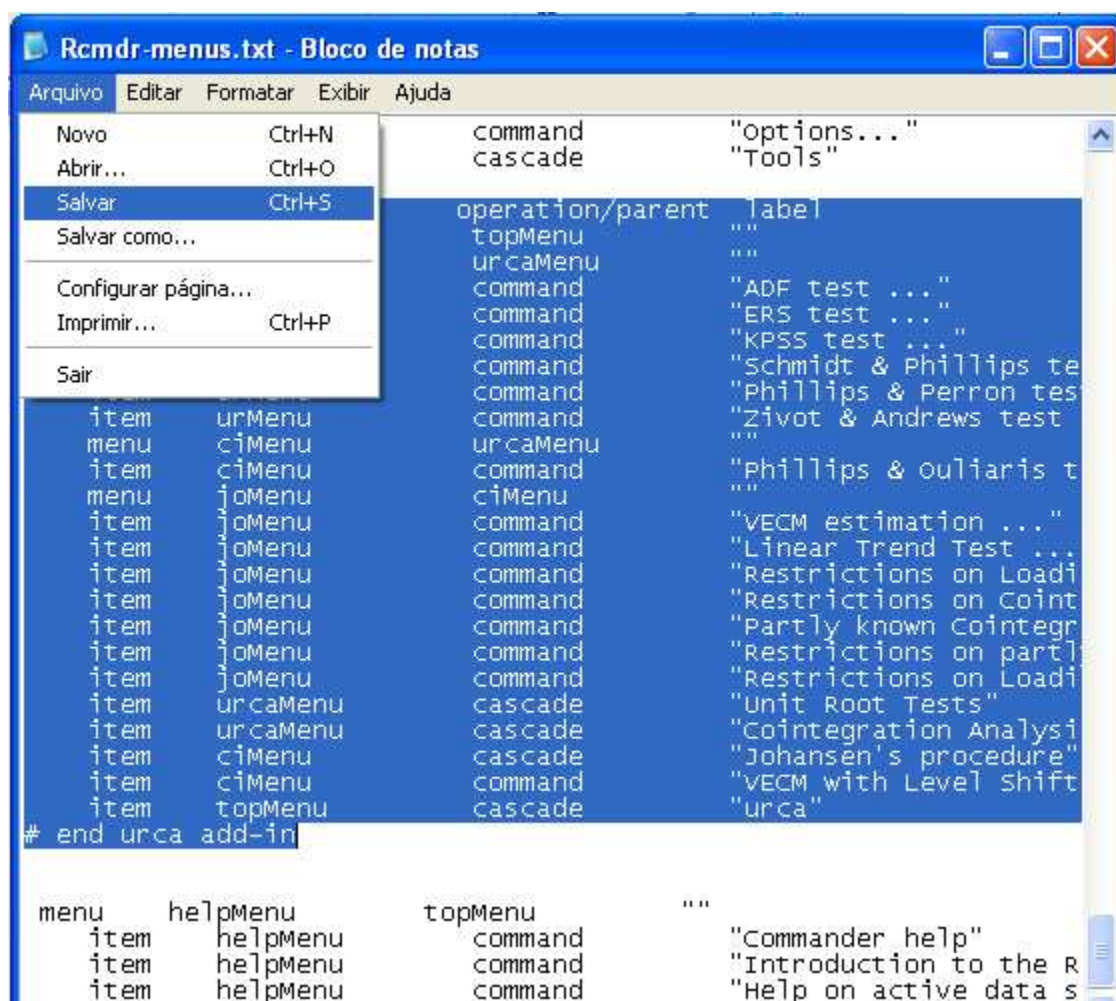
2º passo – copiar os conteúdos do arquivo ‘Rcmdr-menus.txt’:



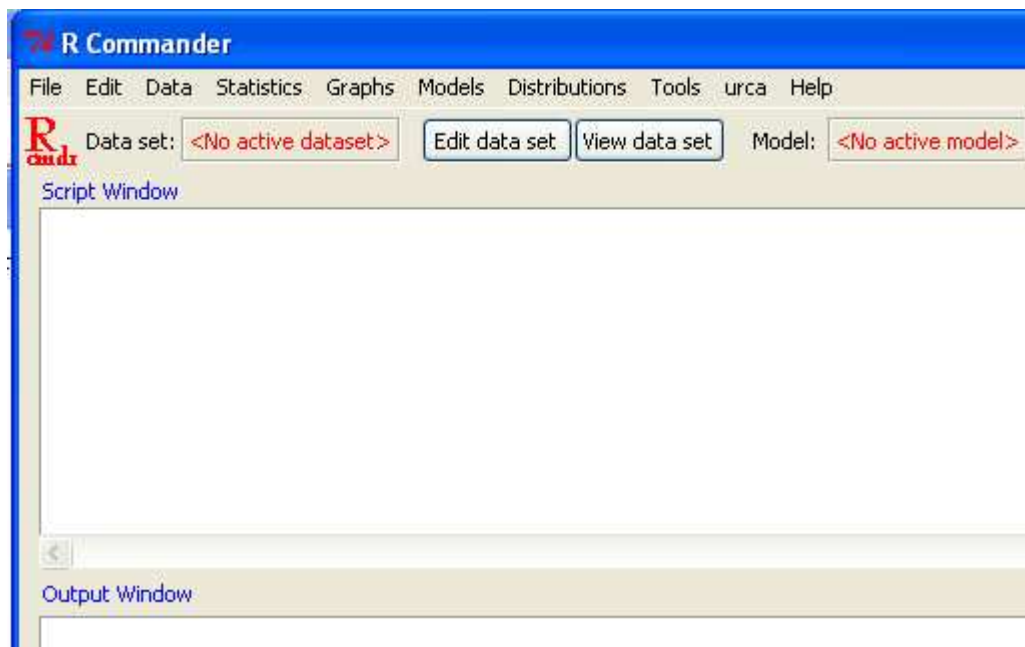
E colar para dentro do arquivo de mesmo nome que situa-se no subdiretório ‘/etc’ do pacote R Commander, acima da linha ‘ menu helpMenu topMenu "" "" "" ‘:



Salvando-o em seguida:



Repare que ao fazer essa sequência de instruções e em seguida abrindo-se o R Commander, o menu denominado 'urca' aparecerá à esquerda do menu 'Help'.



A partir deste exemplo, analisando-se atentamente a estrutura de comandos que foi copiada para dentro do arquivo 'Rcmdr-menus.txt' e observando-se como esse novo item de menu adicionado funciona dentro do R Commander, é possível descobrir como se programa qualquer nova estrutura de menus dentro do R Commander, customizada para qualquer outra necessidade específica. Todas as funções que são chamadas neste exemplo de programação de menu customizado estão devidamente programadas no arquivo 'Rcmdr-urca.R', o qual foi anteriormente copiado para o subdiretório '/etc' do pacote R Commander.

Estruturas de repetição

Funções de Loop

- **FOR**

- Exemplo: Criar um vetor resultante da seguinte operação:
Elevar cada elemento de um dado vetor ao elemento de mesma posição referente a um outro vetor distinto.

Criando os vetores:

```
> vetor1<-c(1:50)
> vetor2<-c(rep(1,10),rep(2,20),rep(3,10),rep(4,10),rep(5,10))
> vetor3<-NULL
```

Usando a estrutura FOR:

```
> for(i in 1:50){  
+ vetor3[i]<-vetor1[i]^vetor2[i]  
+ }
```

Vetor resultante:

```
> vetor3  
[1] 1 2 3 4 5 6 7 8 9  
[10] 10 121 144 169 196 225 256 289 324  
[19] 361 400 441 484 529 576 625 676 729  
[28] 784 841 900 29791 32768 35937 39304 42875 46656  
[37] 50653 54872 59319 64000 2825761 3111696 3418801 3748096 4100625  
[46] 4477456 4879681 5308416 5764801 6250000
```

- **IF**

- Exemplo: Criar uma matriz e retornar, caso ela não seja inversível, uma mensagem indicando que ela não é inversível, caso contrário, retornar a sua matriz inversa:

Ex. 1)

```
> M1<-matrix(c(1,0,0,0),2,2)  
> if (det(M1)==0){print(c("Não é inversível"))}else {solve(M1)}  
[1] "Não é inversível"
```

Ex. 2)

```
> M2<-matrix(c(2,-1,3,1),2,2)  
> if (det(M2)==0){print(c("Não é inversível"))}else {solve(M2)}  
[1,] [2]  
[1,] 0.2 -0.6  
[2,] 0.2 0.4
```

Funções no R

Pode-se definir o conceito de função no R, como sendo um objeto pré-definido que depende de uma ou mais variáveis que serão dadas pelo usuário, a fim de achar um valor específico. Como por exemplo a média, que é uma função do R, ou seja, um objeto que, para ser calculado, depende de duas variáveis: o vetor no qual será calculado a média, e a porcentagem de poda da média.

Exemplos:

Calcular a média do vetor x com uma poda de 20%.

```
> x <- seq (1:10) <ENTER>
> x <ENTER>
[1] 1 2 3 4 5 6 7 8 9 10
> mean(x,trim=0.2) <ENTER>
[1] 5.5
```

Criar uma função que calcule o valor de y tal que $y = (x+1)^2 / 2x$.

```
> Y <- function(x){
+ return( ( ( x+1 )^2 ) / ( 2*x ) )
+ } <ENTER>
> Y(2) <ENTER>
[1] 2.25
```

Criação de funções no R

- **Taxas Equivalentes**

Duas taxas são equivalentes se aplicadas, através de diferentes formas de capitalização, ao mesmo capital, pelo mesmo período de tempo, resultam no mesmo montante.

Exemplos:

Considere i_m = taxa mensal e i_d = taxa diária.

$$1 + i_m = (1 + i_d)^{30} \quad (\text{pois 1 mês} = 30 \text{ dias})$$

Aplicação no R

Veja os exemplos abaixo de funções que calculam taxas equivalentes:

Ex. 1)

```
> # Dia para mês  
>  
> DpM <- function(t){  
+ return(((1+t)^30)-1)  
+ }  
> DpM(0.02)  
[1] 0.8113616
```

Ex. 2)

```
> # Mês para ano  
>  
> MpA <- function(t){  
+ return(((1+t)^12)-1)  
+ }  
> MpA(0.02)  
[1] 0.2682418
```

Ex. 3)

```
> # Ano para mês  
>  
> ApM <- function(t){  
+ return(((1+t)^(1/12))-1)  
+ }  
> ApM(0.1)  
[1] 0.00797414
```

Ex. 4)

```
> # Mês para dia  
>  
> MpD <- function(t){  
+ return(((1+t)^(1/30))-1)  
+ }  
> MpD(0.1)  
[1] 0.003182058  
>  
> MpD(ApM(0.5))  
[1] 0.001126926
```

Ex. 5)

```

> # Semestre para ano
>
> SpA <- function(t){
+ return(((1+t)^2)-1)
+ }
> SpA(0.2)
[1] 0.44

```

- **Juros Simples**

É o sistema no qual os juros incidem apenas sobre capital inicial, ou seja, quando os rendimentos são devidos única e exclusivamente sobre o principal ao longo dos períodos de tempo a que se referir uma determinada taxa de juros.

Aplicação no R

Lembrando que:

Juros = Capital x Taxa x Período

Montante = Capital + Juros

Chamaremos de JS o juros simples e MS o montante referente aos juros simples.

Considere uma aplicação para um Capital de 100 reais, a uma taxa de 10% ao período, por 8 períodos.

No R, pode-se criar a função através dos seguintes comandos:

c = 100, t = 0.1 e n = 8

```

> JS <- function(c,t,n){
+ return(c*t*n)
+ }
> JS(100,0.1,8)
[1] 80

```

Para calcular o montante, deve-se fazer:

```

> MS <- function(c,t,n){
+ return(c*(1+t*n))

```

```
> }
> MS(100,0.1,8)
[1] 180
```

- **Juros Compostos**

É o sistema no qual os rendimentos incorporados ao principal em cada período de tempo a que se referir uma dada taxa de juros, passam, também, a render juros no período seguinte, dessa maneira, conclui-se que o capital investido ou emprestado será acrescido do rendimento de juros, compondo um novo principal, o qual no período seguinte será acrescido de rendimento de juros e assim sucessivamente, ou seja, o rendimento é calculado com juros em cima de juros.

Aplicação no R

Lembrando que:

Juros = Capital $\times ((1 + \text{taxa})^{\text{período}} - 1)$
 Montante = Capital + Juros

Chamaremos de JC o Juros Compostos e de MC o Montante referente aos Juros Compostos.

Considere uma aplicação para um Capital de 100 reais, a uma taxa de 10% ao período, por 8 períodos.

No R, pode-se criar a função através dos seguintes comandos:

```
> JC <- function(c,t,n){
+ return(c*((1+t)^n)-1))
+ }
> JC(100,0.1,8)
[1] 114.3589
```

```
> MS <- function(c,t,n){
+ return(c*(1+t*n))
+ }
> MS(100,0.1,8)
[1] 180
```


Anuidades

O estudo das anuidades nos fornece o instrumento necessário para estabelecer planos de poupança, de financiamento, de recomposição de dívidas e avaliação de alternativas de investimento.

Define-se anuidade, renda certa ou série, a uma sucessão de pagamentos ou recebimentos, exigíveis em épocas pré-determinadas, destinada a extinguir uma dívida ou constituir um capital.

Cada um dos pagamentos que compõem uma série denomina-se termo de uma renda e conforme sejam iguais ou não, a anuidade se denominará, respectivamente, constante ou variável.

Se os pagamentos forem exigidos em épocas cujos intervalos de tempo são iguais, a anuidade se denominará periódica; caso contrário, a série se denominará não-periódica.

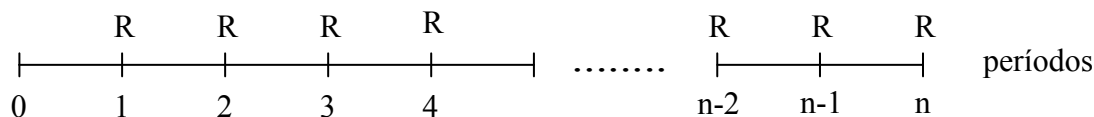
Se o primeiro pagamento for exigido no primeiro intervalo de tempo a que se referir uma determinada taxa de juros, teremos uma anuidade imediata; caso contrário, ou seja, se o primeiro pagamento não for exigido no primeiro intervalo de tempo, ela será diferida. Caso não sejam contados juros durante o período de diferimento, este se denominará período de carência.

Tem-se uma anuidade temporária ou uma perpetuidade conforme seja, respectivamente, finito ou infinito o número de seus termos.

Classificação das anuidades periódicas

- Postecipadas

Quando os pagamentos ou recebimentos forem efetuados no fim de cada intervalo de tempo a que se referir à taxa de juros considerada.



Fórmulas para anuidades postecipadas:

$$C = R \frac{(1+i)^n - 1}{i(1+i)^n}$$

$$S = R \frac{(1+i)^n - 1}{i}$$

Onde:

C = valor atual, principal ou valor financiado

S = valor futuro, montante ou valor de resgate

R = pagamentos ou recebimentos

i = taxa de juros

n = número de períodos

Aplicação no R

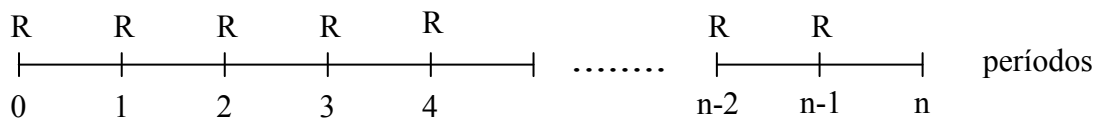
Criaremos funções para calcular C e S:

```
> # Para calcular o valor presente (atual)
> ENTER
> CaP <- function(R,i,n){
+ return(R*(((1+i)^n)-1)/(i*((1+i)^n)))
+ }
> # Exemplo:
> CaP(10000,0.08,12)
[1] 75360.78
```

```
> # Para calcular o valor futuro (final)
> ENTER
> SaP <- function(R,i,n){
+ return(R*(((1+i)^n)-1)/i)
+ }
> # Exemplo:
> SaP(10000,0.8,12)
[1] 14447892
```

- **Antecipadas**

Quando os pagamentos ou recebimentos forem efetuados no início de cada intervalo de tempo a que se referir a taxa de juros considerada.



Fórmulas para anuidades antecipadas:

$$C = R \frac{(1+i)^n - 1}{i(1+i)^{n-1}} \quad S = R \frac{((1+i)^n - 1)}{i} (1+i)$$

Aplicação no R

Criaremos funções para calcular C e S:

```
> # Para calcular o valor presente (atual)
> ENTER
> CaA <- function(R,i,n){
+ return(R*( ( (1+i)^n) - 1 ) / (i*( (1+i)^(n-1) ) ) )
+ }
> # Exemplo:
> CaA(10000,0.8,12)
[1] 22480.55
```

```
> # Para calcular o valor futuro
> ENTER
> S <- function(R,i,n){
+ return( ( R*( ( (1+i)^n) - 1)/i )*(1+i) )
+ }
> # Exemplo:
> S(10000,0.8,12)
[1] 26006206
```

Perpetuidades:

No caso dos pagamentos serem infinitos, teremos o que se denomina de perpetuidade. Como o número de pagamentos é infinito, não tem sentido o cálculo do montante, mas podemos calcular o valor atual.

- Postecipada

$$C = \frac{R}{i}$$

Aplicação no R

```
> CpP <- function(R,i){
+ return(R/i)
+ }
> # Exemplo:
> CpP(10000,0.8)
[1] 12500
```

- Antecipada

$$C = \frac{R}{i} \times (1+i)$$

Aplicação no R

```
> CpA <- function(R,i){
+ return( (R/i) * (1+i) )
+ }
> # Exemplo:
> CpA(10000,0.8)
[1] 22500
```

Métodos de Tarifação

A análise sistemática das contingências de vida humana (sobrevivência e morte) constitui os fundamentos do trabalho de um atuário. Na solução dos problemas envolvendo essas contingências, é necessário proceder alguns tipos de medição dos seus efeitos através dos métodos de tarifação.

Os métodos de tarificação são formas estudadas para indicar o valor que deverá ser cobrado por um serviço a fim de que este valor cubra o total de gastos relativos ao investimento e a operação de serviço. Há diversos métodos, como prêmio puro, sinistralidade, julgamento ou subjetivo e o método das tábuas biométricas, que será o enfoque deste capítulo.

As Tábuas Biométricas, também conhecidas como tábuas de mortalidade, são um método de tarificação utilizado como parâmetro para tarifar os planos de previdência aberta complementar. Portanto, é utilizado no seguro de Pessoas e de anuidades, trata-se do instrumento que mede a duração da vida humana, ou seja, mede a probabilidade de sobrevivência e de morte ou a probabilidade de entrada em invalidez. Esta, pode ser de dois tipos: tábua de entrada em invalidez ou tábua de mortalidade de inválidos.

As tábuas de mortalidade são um método determinístico por aplicarem fórmulas determinísticas e probabilidades de morte constituídas a partir de estudos estatísticos prévios realizados pelos atuários. Elas são construídas baseadas em informações brutas de mortalidade, passando inicialmente por um processo estatístico de regularização, em seguida por um processo de ajustamento analítico e por fim aplica-se um carregamento de segurança positivo, se a tábua for para coberturas de riscos ou negativo, se a tábua for para cálculo de anuidades.

Chama-se de raiz da tábua o número inicial de pessoas da amostra a ser observada. E agrupam-se 5 colunas para a formação da tábua, de maneira que:

X = idade

l_x = número de pessoas

d_x = número de pessoas mortas com idade x , ou seja, pessoas que atingiram a idade x , mas não chegaram à idade $x + 1$

Q_x = probabilidade de uma pessoa com idade x morrer, obrigatoriamente, antes de atingir a idade $x + 1$

p_x = probabilidade de uma pessoa com idade x sobreviver à idade $x + 1$, ou seja, é a probabilidade de uma pessoa com idade x , sobreviver, pelo menos, mais um ano (chegar à idade $x + 1$)

Aplicação no R

Para exemplo, criaremos agora uma função no R que monte uma tábua de mortalidade conforme os comandos abaixo:

Usaremos como parâmetro as idades de 0 até 100 anos.

```
idade<-seq(0,100,5)
>
> qx<-0.0001+0.0002*((idade^2)/10)
>
> px<-1-qx
>
> lx<-NULL
> lx[1]<-10^6
> for(i in 2:length(idade)){
+ lx[i] <- lx[i-1]-lx[i-1]*qx[i-1]}
>
> dx <- NULL
> for(i in 1:length(idade)){
+ dx[i] <- qx[i]*lx[i]}
>
> tabuabiometrica<-data.frame(idade,qx,px,lx,dx)
> tabuabiometrica
```

	idade	qx	px	lx	dx
1	0	0.0001	0.9999	1000000.0	100.000
2	5	0.0006	0.9994	999900.0	599.940
3	10	0.0021	0.9979	999300.1	2098.530
4	15	0.0046	0.9954	997201.5	4587.127
5	20	0.0081	0.9919	992614.4	8040.177
6	25	0.0126	0.9874	984574.2	12405.635
7	30	0.0181	0.9819	972168.6	17596.251
8	35	0.0246	0.9754	954572.3	23482.480
9	40	0.0321	0.9679	931089.9	29887.985
10	45	0.0406	0.9594	901201.9	36588.796
11	50	0.0501	0.9499	864613.1	43317.115
12	55	0.0606	0.9394	821296.0	49770.535
13	60	0.0721	0.9279	771525.4	55626.983
14	65	0.0846	0.9154	715898.4	60565.008
15	70	0.0981	0.9019	655333.4	64288.210
16	75	0.1126	0.8874	591045.2	66551.693
17	80	0.1281	0.8719	524493.5	67187.622

18	85	0.1446	0.8554	457305.9	66126.435
19	90	0.1621	0.8379	391179.5	63410.193
20	95	0.1806	0.8194	327769.3	59195.133
21	100	0.2001	0.7999	268574.2	53741.688