

Introdução ao R

Profa. Rosangela Ballini

Bibliografia Básica:

- Venables, W. N., Smith, D. M. and the R Core Team. An Introduction to R: A Programming Environment for Data Analysis and Graphics, Version 3.1.0 (2014-04-10). Disponível em:
<http://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>.
- Wooldridge, J.M. (2016). Introductory Econometrics: a Modern Approach. Stata Press College Station, USA.

Introdução ao R

Objetivo desta aula: introduzir o uso do programa R (R Development Core Team).

- R: proposto por Ross Ihaka e por Robert Gentleman, início dos anos de 1990, na Universidade de Auckland, Nova Zelândia, e vem sendo desenvolvido por um esforço colaborativo de pessoas em vários locais do mundo.
- R: linguagem de programação e ambiente para computação estatística e gráfica.
- Algumas das suas principais características são o seu caráter **gratuito** e a sua disponibilidade para uma gama bastante variada de sistemas operacionais.

Instalando o R

Para baixar o R acesse o site do R www.r-project.org:

- Clique em CRAN (Comprehensive R Archive Network)
- Escolha o espelho de sua preferência (CRAN mirrors). Por exemplo, “Brasil – Universidade Federal do Paraná”
- Clique no sistema operacional desejado (Linux, Mac, Windows)
- Clique em base e salve o arquivo do R. Após é só executar o arquivo.

R é um software de código aberto, de modo que o código-fonte completo está disponível e pode ser inspecionado para entender o que o código faz, aprender com ele e modificá-lo.

Interface do R

- Ao iniciar o R abrirá automaticamente o Console, ou “área de trabalho” ou ainda chamado de *workspace*;
- Internamente ao console, se encontra o *prompt*, indicado pelo sinal **>**;
- Para usar o R é necessário conhecer e digitar comandos;
- Os comandos aparecem em **vermelho** e o *output* do R aparece em **azul**.

Interface do R

- Sinal de $+$ no lugar do *prompt*, indica que ficou faltando algo na linha de comandos anterior (isso acontece quando houve um erro, ou quando a finalização do comando só ocorrerá na próxima linha)
- Símbolo $\#$: usado para comentários, sendo que tudo que é digitado após este símbolo é desconsiderado pelo R.
- Distinção entre maiúsculas e minúsculas. Por exemplo, caracteres “X” e “x” são entendidos como variáveis diferentes
- Interfaces: R-commander ou Tinn-R ou Studio-R
- Studio-R: www.rstudio.com

Menu de ajuda no R para verificar a sintaxe de uma função:

- Pesquisar no R com palavras chave usando a função `help.search()`.

Exemplo: `> help.search("logarithms")` ou
`> ??logarithms`

- Buscar ajuda na internet, no site do R, com a função `RSiteSearch()`:

Exemplo: `> RSiteSearch('logarithms')`

- Arquivos de ajuda do R use o comando `help(nome.da.função)` ou `?nome.da.função`

Exemplo: `> help(log)` ou `> ?log`

O arquivo de `help` do R possui 10 tópicos básicos:

1. **Description** – faz um resumo sobre o uso da função
2. **Usage** – mostra como a função deve ser utilizada e quais argumentos podem ser especificados
3. **Arguments** – explica o que é cada um dos argumentos
4. **Details** – detalhes sobre o uso e aplicação da função
5. **Value** – mostra os resultados
6. **Note** – notas sobre a função
7. **Authors** – lista os autores da função
8. **References** – referências para os métodos usados
9. **See also** – mostra outras funções relacionadas que podem ser consultadas
10. **Examples** – exemplos do uso da função.

Objetos no R

Objetos: caracterizados por nomes contendo, e por atributos que especificam o tipo de dados representados por um objeto.

Todos os objetos têm dois atributos: `mode()` e `length()`:

- `mode()`: tipo base dos elementos do objeto, sendo que há quatro tipos principais: *numeric*, *character*, *complex*, e *logical* (*FALSE* or *TRUE*).
- `length()`: tamanho (ou comprimento) do objeto.

Exemplo:

```
> x = c("Sul", "Sudeste", "Centro - Oeste", "Nordeste", "Norte")
> mode(x)
[1] character
> length(x)
[1] 5
```

Tipos básicos de objetos são:

- **Vetores:** seqüência de valores numéricos, caracteres, complexos, lógicos
- **Matrizes:** coleção de vetores em linhas e colunas, todos os vetores devem ser do mesmo tipo
- **data.frame:** similar a matriz, mas aceita vetores de tipos diferentes
- **Funções:** as funções criadas para fazer diversos cálculos também são objetos do R
- **ts:** numérico, caractere, complexo ou lógico

Comandos Auxiliares

Função	Descrição
ls() ou objects()	lista de variáveis definidas
library()	mostra todos os pacotes instalados
rm(x)	apaga a variável x
rm(x,y)	apaga as variáveis x e y
remove(list=nome)	apaga todos os objetos da lista
class(x)	ver que tipo de objeto é x
q()	sair do R com a opção de salvar a <i>workspace</i> em um arquivo ("Name.RData") e o histórico de comandos em outro arquivo ("Name.RHistory")
ctrl + L	no teclado, pressione "ctrl+L" para limpar a tela da console

Operações e Funções Matemáticas Básicas

Operações	Descrição
+	Adição
-	Subtração
/	Divisão
*	Multiplicação
^ ou **	Potênciação

Exemplos:

$$> 2 + 4$$

$$[1] 6$$

$$> 12/3$$

$$[1] 4$$

$$> 2^3$$

$$[1] 8$$

$$> 3 - 5$$

$$[1] -2$$

$$> 5 * 2$$

$$[1] 10$$

Funções Matemáticas Básicas

Função	Descrição
<code>abs(x)</code>	valor absoluto de x
<code>log(x, b)</code>	logaritmo de x com base b
<code>log(x)</code>	logaritmo natural de x
<code>exp(x)</code>	exponencial elevado a x
<code>sqrt(x)</code>	raiz quadrada de x
<code>round(x, digits = n)</code>	arredonda x com n decimais
<code>length(x)</code>	número de elementos do vetor x
<code>max(x)</code>	seleciona o maior elemento do vetor x
<code>min(x)</code>	seleciona o menor elemento do vetor x
<code>range(x)</code>	retorna o menor e o maior elemento do vetor x
<code>cor(x,y)</code>	correlação entre x e y

Exemplos:

```
> log(100)
```

```
[1] 4.60517
```

```
> exp(1)
```

```
[1] 2.718282
```

```
> sqrt(16)
```

```
[1] 4
```

Vetor: unidade de base no R. Um vetor é gerado utilizando a função `c()`, em que `c` significa “concatenar”. Exemplos:

```
> x <- c(0.8, 2.4, 10, 98.109, 23, 2)
```

gera um objeto `x`, um vetor, contendo as entradas 0.8, 2.4, 10, 98.109, 23, 2.

O tamanho de um vetor é obtido usando `length()`:

```
> length(x)
```

```
[1] 6
```

Vetores no R

A partir do vetor criado podemos realizar cálculos.

Exemplo:

```
> 4 * x + 1
```

```
[1] 4.200 10.600 41.000 393.436 93.000 9.000
```

Funções matemáticas também podem ser aplicadas ao vetor x.

Exemplo:

```
> log(x),
```

o qual retorna um vetor contendo o logaritmo dos valores originais de x:

```
[1] -0.2231436 0.8754687 2.3025851 4.5860791 3.1354942
```

Para arredondar os valores para 3 casas decimais:

```
> round(y, digits = 3)
```

```
[1] -0.223 0.875 2.303 4.586 3.135
```


Construção de Matrizes no R

Uma matriz 2×3 contendo os elementos 1 a 6, nas colunas, é gerada por:

```
> A = matrix(1 : 6, nrow = 2)
```

Alternativamente, `ncol` poderia ser utilizada:

```
> A = matrix(1 : 6, ncol = 3)
```

A dimensão de uma matriz pode ser determinada usando `dim()`, ou `nrow()`, ou `ncol`:

```
> dim(A)
```

```
[1] 2 3
```

```
> nrow(A)
```

```
[1] 2
```

```
> ncol(A)
```

```
[1] 3
```

Construção de Matrizes no R

Suponha uma matriz de ordem 3×3 com os seguintes elementos 1, 3, 2, 2, 8, 9, 3, 2, 7:

```
> A = matrix(c(1, 3, 2, 2, 8, 9, 3, 2, 7), ncol = 3)
```

Note que os números foram “espalhados” pela matriz por coluna, i.e. primeiro foi preenchida a primeira coluna, depois a segunda, etc. Caso não seja isto o que pretendemos, poderemos preencher a matriz linha por linha, usando o comando `byrow`, ou seja,

```
> B = matrix(c(1, 3, 2, 2, 8, 9, 3, 2, 7), ncol=3, byrow=T)
```

em que T representa o comando lógico TRUE.

Operações de Matrizes no R

1. Adição de matrizes: podemos usar o operador +
> `A + B` # A e B matrizes de mesma ordem
2. Subtração de matrizes: operador -
> `A - B` # A e B matrizes de mesma ordem
3. Multiplicação por um escalar: *
> `7*A` # A matriz de ordem $n \times m$
4. Matriz Transposta: `t()`
> `t(A)` # A matriz de ordem $n \times m$
5. Multiplicação de matrizes: `%*%`

Exemplo:

```
> A=matrix(c(1,3,2,2,8,9),ncol=2)
> B=matrix(c(5,8,4,2), ncol=2)
> A %*% B
```

Adicionar Módulos no R

- Para instalar um pacote vá ao site do R (www.r-project.org), clique em CRAN, escolha o espelho e clique em *packages*. Clique no pacote que deseja e depois clique em windows binary e salve o arquivo. Abra o R, clique em *Packages*, depois em *Install Package(s) from local zip files* e selecione o arquivo do pacote que você baixou.
- O R também pode conectar-se diretamente à internet. Desta forma é possível, instalar e atualizar pacotes sem que seja necessário acessar a página do R.

Não basta apenas instalar um pacote. Para usá-lo é necessário “carregar” o pacote sempre que você abrir o R e for usá-lo. Use a função `library()` ou `require()` para rodar um pacote.

Exemplo de Ajuste de Regressão Linear Simples

Considerando o conjunto de dados de Stock & Watson (2007), que contém informações sobre o número de assinaturas de revistas científicas (em inglês, *journals*) de economia nos EUA, feitas por bibliotecas em 2000.

O conjunto de dados, está disponível no pacote **AER** com o nome de Journals.

```
> data('Journals', package = 'AER')  
> list(Journals)  
> dim(Journals)  
> names(Journals)
```

Exemplo de Ajuste de Regressão Linear Simples

O conjunto de dados é composto por 180 observações de 10 variáveis, entre elas, o número de assinaturas para bibliotecas (*subs*), número de citações (*citations*) e preço (*price*).

Interesse: relação entre a demanda por revistas e os respectivos preços. Uma medida adequada do preço dos periódicos científicos é o preço por citação. Um gráfico de dispersão (em inglês, *scatterplot*) em logaritmos desta relação é obtido executando:

```
> plot(log(subs) ~ log(price/citations),  
+data = Journals)
```

A figura ilustra que o número de assinaturas decresce com o preço.

Exemplo de Ajuste de Regressão Linear Simples

Modelo de regressão linear: ajuste por MQO usando a função `lm()` (*linear model*), ou seja,

```
> j.lm = lm(log(subs) ~ log(price/citations),  
+data = Journals)
```

Para adicionar a reta de mínimos quadrados no gráfico de dispersão existente:

```
> abline(j.lm)
```

Para visualizar o sumário detalhado do modelo de regressão linear ajustado, execute:

```
> summary(j.lm)
```

Exemplo de Ajuste de Regressão Linear Múltipla

Estimativa de uma equação de salários na forma semi-logarítmica utilizando os dados de uma amostra aleatória de dados cross-section da *Current Population Survey*, composta por 534 observações por 11 variáveis, disponíveis no pacote **AER**.

Execute:

```
> data('CPS1985', package = 'AER')  
> cps = CPS1985  
> list(cps)  
> dim(cps)  
> names(cps)
```


Exemplo de Ajuste de Regressão Linear Múltipla

A equação de salários é estimada usando o $\log(wage)$ como variável dependente e as variáveis *education* e *experience* como regressores. Para *experience* um termo quadrático também é incluído. A equação de salários é ajustada usando a função `lm()`

```
> cps.lm = lm(log(wage) ~ experience +  
> +I(experience^2) + education, data = cps)  
> summary(cps.lm)
```

Leitura de Dados no R

R usa o diretório de trabalho para ler e/ou escrever em arquivos.

Função `getwd()`: mostra o diretório de trabalho.

Função `setwd()`: altera o diretório de trabalho.

Função `dir()`: lista os arquivos do diretório de trabalho.

Função `read.table()`: leitura de dados armazenados em arquivo texto (ASCII)

R também lê arquivos em outros formatos (Excel, SAS, SPSS,...), e bancos de dados de acesso do tipo SQL, mas as funções necessárias para isso não estão no pacote básico.

As variantes comuns da função `read.table()` são

```
read.csv()
```

```
read.csv2()
```

```
read.delim()
```

```
read.delim2()
```

```
read.xls()
```

Leitura de Dados no R

Exemplo para uso da função `read.table()`:

```
> carregar <- read.table('arquivo.csv',  
+ header = T, sep = ',', dec = '.')
```

em que:

`carregar`: objeto no qual os dados lidos serão reconhecidos pelo R;

`<-`: sinal que atribui os dados lidos ao objeto `carregar`;

`read.table`: função que lê o arquivo do tipo **.csv**;

`header`: parâmetro que permite indicar se o arquivo de dados (`data.frame`) tem ou não o nome nas colunas (título) na primeira linha de dados;

`sep`: parâmetro que permite indicar o tipo de separador dos dados presentes no arquivo;

`dec`: parâmetro que permite indicar o caractere usado como separador de casas decimais dos números reais.

Leitura de Dados no R

Observação: existem outras sintaxes para carregar dados no console do R (verifique isso utilizando o comando `help(read.table)`), porém os argumentos permanecem idênticos aos apresentados acima.

Caso o arquivo tenha título, podemos verificar o nome destes títulos por meio do comando:

```
> names() # no argumento vai sempre o nome do objeto desejado
```

Podemos ver a dimensão do arquivo carregado por meio do seguinte comando:

```
> dim()
```

Escrevendo e Salvando Dados no R

`write.table()`: função para escrever um objeto em um arquivo

?`write.table`: use o help para obter os detalhes desta função