Introdução ao R

O R é uma linguagem para processamento estatístico, inspirada no S, etc. (vide “What is R” no site http://www.r-project.org/)...

O R é Open-Source!
Compilado para Windows, Linux, etc.
Pode ser utilizado de forma interativa (abrindo-se o console do R e executando os comandos) ou via scripts (no Linux, pode-se rodar scripts R diretamente do bash!)

Getting Started

1 - abra seu R
2 - vamos criar uma variável "valores" e adicionar 20 valores aleatórios seguindo uma distribuição normal:
```
> valores = rnorm(20)
```

As atribuições podem ser feitas com "=", como no exemplo acima, ou com o "<-", ex.:
```
> valores <- rnorm(20)
```
Recomenda-se o uso do " <- "

3 - Para ver os valores de uma variável qualquer, digite o nome desta variável. Exemplo:
```
> valores
valores
 [1]  0.30056375 -0.15993135 -0.15387629 -0.10078039  0.18343576  1.11735283
 [7]  1.24036957 -1.54334821  1.25146702 -0.53188146 -0.71065443 -0.66650575
[13] -0.51257763 -0.22184441 -1.09151412 -0.08100808 -0.33180404 -0.38697190
[19]  1.29031965 -0.42876790
```

Neste exemplo, tem-se 6 ítens por linha. Para ajudar, no começo de cada linha, tem o índice ao qual o elemento se refere.

4 - Algumas funções tem saída visual. Para estas, um popup será exibido.
Por exemplo, para exibir um histograma da variável valores:
```
> hist(valores)
```

Em algumas plataformas, e.g., Rstudio, o output será feito na área destinada às saídas gráficas da plataforma, ao invés de uma janela popup.

5 - Para salvar uma figura, é bem simples.

```
> pdf("exemplo1.pdf")
> hist(valores)
> dev.off()
```

Neste caso, tudo que seria exibido num popup, será direcionado à saída indicada (no caso, exemplo1.pdf). Para fechar e salvar o arquivo, é necessário utilizar o dev.off(). Existem outras funções similares o pdf (ex. png).

A sequência mostrada anteriormente poderia ter sido adaptada a um único comando, caso utilizássemos *scripts*. Para tanto, crie um arquivo com um editor de textos de sua preferência:

```
valores <- rnorm(20)
pdf("exemplo1.pdf")
hist(valores)
dev.off()
```

Salve com um nome qualquer cuja extensão seja .R (ex. *exemplo1.R*)

Várias formas de executar o script:
1 - de dentro do R chame: *source("exemplo1.R")*
2 - via o comando Rscript: do sistema operacional, chame: *Rscript exemplo1.R*
3 - pela linha de comando do sistema operacional. No linux, por exemplo, pode-se chamar *R CMD BATCH exemplo1.R*

Teste as três formas!

Buscando ajuda

Caso conheça uma função e queira conhecer mais detalhes, pode-se usar o comando help:

```
> help(hist)
```

Caso nem saiba o nome da função, pergunte ao Google. Geralmente uma busca começada por "R" traz bons resultados, exemplo “R calcular média”!

Bibliotecas/pacotes

Devido à popularidade do R, muitas funções já foram desenvolvidas e estão disponíveis no formato de bibliotecas.

Para carregar uma biblioteca:

> library(fpc)

O R carregará todas as dependências da biblioteca.

Eventualmente, pode ser preciso instalar a biblioteca:
>install.packages(fpc)

Via de regra O R busca e instala todas as dependências.

Nos sites de referência das funções, normalmente a biblioteca é indicada por { }
Ex.: no site http://stat.ethz.ch/R-manual/R-patched/library/base/html/mean.html, a função mean aparece como pertencente à biblioteca {base}. Neste caso, não é preciso carregar a biblioteca.

Atenção! Leia atentamente as explicações! A função pode não ser *exatamente* aquela que você queria....

Exemplos de funções

Atenção, este sumário não substitui uma lida no *help* de cada função!

c - concatena valores em um vetor:
> x <- c(1,2,4)

cria um vetor de três elementos a partir de três vetores de um elemento (o R considera um número como um vetor de um elemento!)

Logo, ao executar
> q <- c(x,x,8)

tem-se que q = :
a. ((1,2,4),(1,2,4),8)
b. (1,2,4,1,2,4,8)
c. (1,2,4,8)
d. Erro! Só se concatenam números!

Teste e confira a resposta.

mean - Calcula a média dos valores em um conjunto
Para calcular a média dos valores de q:
> mean(q)
sd - Calcula o desvio padrão dos valores em um conjunto

> sd (q)

Como saber o valor do 5o elemento de q?
> q[5]

Atenção: o índice do primeiro elemento do vetor é 1!

sum - soma os valores de um vetor.
como calcular a soma dos três primeiros números?
> sum(q[1:3])

leia sobre:
colnames
dim, nrow, ncol

Laços e condicionais em R

Sim, existem while, for, if, else, etc.

Entretanto, programar em R não é a mesma coisa que programar em Java ou C++. Costumeiramente, um mesmo script pode ser executado de forma bem mais simples se você utilizar funções que minimizam o uso de laços e condicionais. O uso destas estruturas, apesar de recomendado, não é obrigatório.

Leia sobre apply, tapply, aggregate, by, etc.

Tipos e estruturas básicas de dados

Existem 3 tipos básicos:
numeric (integer, double)
character
logical

Estruturas básicas:
Vector
Uma dimensão, de um mesmo tipo. Pode ser criado, caso tenham-se os valores, via a função *c*, como foi visto, ou explicitamente:
v = vector(mode = "integer", length = 10)
primeiro elemento: v[1]
último elemento: v[length(v)]

Array
Mais de uma dimensão, mas todos os elementos precisam ter o mesmo tipo.
a = array(1:10, dim = c(5, 2))

a[,1] => Todas as linhas da coluna 1
a[2,] => Todas as colunas da linha 2
a[1,2] => elemento da linha 1, coluna 2

Matrix
Similar ao array, com apenas duas dimensões.
m = matrix(1:10, nrow = 5, ncol = 2)

Veja quais os valores a matriz tem, caso você não passe os valores:
m = matrix(nrow = 5, ncol = 2)

Data frame
Estrutura um pouco mais complexa que permite colunas com tipos diferentes. A quantidade de linhas, contudo, precisa ser a mesma.

df = data.frame(Coluna1 = c("A", "B", "C"),
Coluna2 = c(1, 2, 3),
Coluna3 = c(TRUE, FALSE, TRUE))
df

Para acessar os valores de uma coluna, pode-se utilizar do “$”:
df$Coluna2

List
Similar ao data.frame, mas cada coluna pode ter quantidade de linhas diferente.
Exemplo:
l = list(Atributo1 = c("A", "B", "C"),
Atributo2 = c(1, 2, 3, 4, 5),
Atributo3 = c(TRUE, FALSE))
l

Lendo e escrevendo arquivos

Lendo:
read.table
Tenta abrir os dados de um arquivo, considerando uma distribuição tabular.
Utilização básica: read.table(“arquivo.txt”)

Provavelmente você terá problemas. Leia o help.

Caso, seu arquivo possua um separador, consulte a opção “sep” do read.table. Caso o seu separador seja “,”, ou seja, você tiver um CSV (Comma Separated Values ou valores separados por vírgula), pode usar diretamente o **read.csv**, que é equivalente ao read.table(sep=”,”).

De forma análoga, existem o **write.table** e o **write.csv**.

Filtrando
Uma das tarefas principais. Pode-se usar a função subset (vide help) ou, diretamente pelo índice:

```
x = -5:5
x.positivos = x[x > 0]
```
(à propósito, : denota um intervalo, logo -5:5 equivale a -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5).

Pergunta: x.positivos é um atributo de x ou outra variável??

Código

R é uma linguagem interpretada e tudo pode ser organizado dentro de um mesmo script. Entretanto, essa não é uma boa prática de programação. Logo, é **essencial** que haja um esforço a mais para modularizar seus scripts. Siga no mínimo esses passos:

1. Utilize funções sempre que houver alguma modularidade;
2. Dê nomes significativos a suas variáveis;
3. Use arquivos de cabeçalho. Para isso, basta colocar todas as funções que possuam relacionamentos entre si em um mesmo arquivo .R e carregar esse arquivo no script que for utilizá-los com o *source("seuArquivo.R")*.

Também é uma prática desejável a manutenção de versões. Uma dica é utilizar repositórios, tais como o *github*, para postar os seus códigos.

Warm-up

Tente seguir as tarefas abaixo. A resposta está ao final do documento. Melhor não olhar ainda... Atenção: pode ser necessário a utilização de funções que não estão neste documento!!

1. Carregue o arquivo gastos.csv e exiba um histograma contendo os gastos cujos valores são maiores do que R$ 20.000.

2. Utilizando o mesmo arquivo da questão anterior, salve no arquivo out.txt a descrição dos 5 maiores gastos. P.s.: no arquivo só devem constar a descrição dos gastos, sem aspas, números ou outros artefatos.

Por fim:

Para sair do R :(
q()

Existe a opção de salvar o workspace. Caso faça isso, ao reiniciar o R você pode carregar o workspace salvo com todas as suas variáveis, funções e últimos comandos.

Leia sobre isso! Para quem usa o RStudio é um pouco mais simples.