Breve Introdução à Programação em R

Elsa Moreira

efnm@fct.unl.pt

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

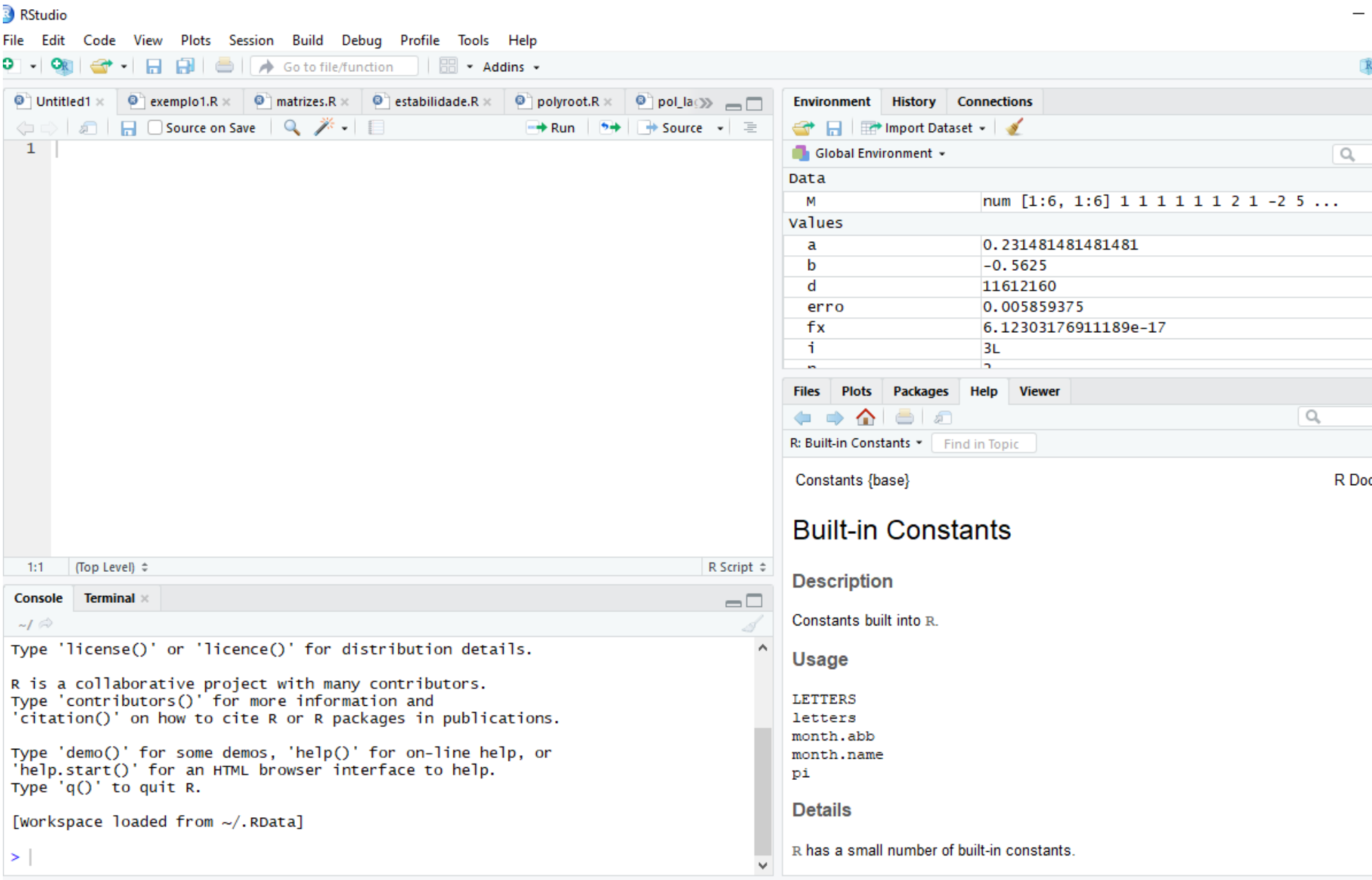
O que é o R e o R studio?

- O software R-project ou, simplesmente, R é um software que embora, mais vocacionado para a análise estatística de dados, é utilizado em todas as sub-áreas da matemática por matemáticos e engenheiros.
- O R é obtido forma gratuita em http://www.r-project.org/.
- Permite efectuar toda uma diversidade de calculos e análises, através da utilização das bibliotecas que possui e de programas feitos pelo utilizador.
- O R Studio é um ambiente de computação/editor do R que facilita a sua utilização;
- O R Studio também é um software gratuito e pode ser descarregado em http://www.rstudio.com/

Instalação

- Descarregar o programa seleccionando um dos CRAN mirror mais próximos, escolhendo a versão associada ao sistema operativo adequado (Windows, e.g.) e seguindo as instruções seguintes:
  - Descarregar o ficheiro executável (e.g.) R-4.3.1-win.exe;
  - Executar esse ficheiro, que permite a instalação do sistema base e dos packages (bibliotecas) recomendados;
- Depois de instalado o programa R pode ser encontrado no Desktop (ambiente de trabalho) pelo menos um atalho para abrir o programa R;
- Para descarregar o R Studio, seguir instruções semelhantes;
- Depois de instalado o R studio, basta duplo clicar no atalho que vai aparecer no ambiente de trabalho.

R Studio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Untitled1
exemplo1.R
matrizes.R
estabilidade.R
polyroot.R
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Console
Terminal
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
>
Environment
History
Connections
Import Dataset
Global Environment
Data
M num [1:6, 1:6] 1 1 1 1 1 1 2 1 -2 5 ...
Values
a 0.231481481481481
b -0.5625
d 11612160
erro 0.005859375
fx 6.12303176911189e-17
i 3L
Files
Plots
Packages
Help
Viewer
R: Built-in Constants
Find in Topic
Constants {base}
R Doc
Built-in Constants
Description
Constants built into R.
Usage
LETTERS
letters
month.abb
month.name
pi
Details
R has a small number of built-in constants.

Início de sessão e comandos

- Abrir um ficheiro "script" na opção "New file" do separador "File";
- Escrever as instruções/comandos que quero executar no R diretamente na janela de trabalho;
- Para executar todas as instruções do ficheiro "script" basta clicar no botão "Source"Para executar apenas a instrução onde está o cursor, clicar no botão "Run"; O resultado aparece na janela da consola;
- Guardar o ficheiro "script" com as instruções dadas de forma a que possa voltar a executá-las novamente noutra altura;
- Para saber em que diretoria estou trabalhar escrever a instrução getwd();
- Para alterar a diretoria de trabalho, escrever a instrução setwd("caminho"), por exemplo: setwd("C:/Users/Elsa Moreira/programas R").

Objectos

- Um objecto pode ser:
  - uma constante;
  - uma variável;
  - um vector;
  - uma matriz;
  - um array;
  - uma função;
  - uma dataframe;
  - list, etc.
- A cada objecto dá-se um nome que pode conter letras minúsculas e/ou maiúsculas, números e o caracter "."
- O R faz a distinção entre maiúsculas e minúsculas. Por exemplo, X e x são objetos diferentes.

Objectos, operadores e expressões

- Ao darmos a instrução a <− 45 criamos um objeto que é uma constante "a" ao qual estamos a atribuir o valor 45.
- Para ver o conteúdo do objecto "a", executo o seu nome.
  Se executar a, vai aparecer na consola:
  [1] 45
- Se executar a expressão b <− 289*3+456/2-11∧2 vai aparecer:
  > b
  [1] 974
- Se executar d <− sqrt(23)/(log(exp(23)+ sin(1/2))) vai aparecer:
  > d
  [1] 0.2085144

Precisão dos valores/algarismos

- O R por defeito usa no cálculo números com uma precisão até 22 digitos, mas imprime apenas 7;
- Para alterar este valor e passar a imprimir com 22 algarismos, utiliza-se o comando
  options(digits=22);
- Quando um número entre -1 e 1 tem casas decimais com 3 ou mais zeros (por ex. 0.0001), por defeito o R imprime o número em notação cientifica, i.e., 1e-04;
- Para desactivar a notação cientifica e aparacerem o número de casas decimais que pretendemos, utiliza-se
  options(digits=17,scipen=999);
- Para arredondar um número "n" às "k" casas decimais, utiliza-se round(n,k).

Tipo de objeto Vector

- Um vetor é uma constante multidimensional que pode ser só de números ou só de caracteres;
- Um vector pode ser criado através da instrução combine, c(·), em que as várias componentes do vetor são separadas por ",";
- Por exemplo a instrução z<− c(21,52,34,41,85,69,73,38,92,12) cria um vector com componentes numéricas;
- Se quizermos saber o nº de componentes de z fazemos lenght(z) e aparece 10;
- Se fizermos z[4] vai aparecer o valor 41 relativo à 4ª componente do vector z;
- A instrução nomes<− c("ana", "paulo", "jorge", "teresa") cria um vector cujas componentes são os nomes "ana", "paulo", "jorge" e "teresa".

Operações com vectores

- Por exemplo, fazendo k<− c(0.12,0.23,0.29) e depois v<− 1000*(k+1), obtemos
  [1] 1120 1230 1290;
- Se fizermos w<− rep(1,3), é criado o vector w com as componentes 1 1 1;
- Então é possível fazer j<− 1000*(k+w) e obtemos o mesmo vector
  [1] 1120 1230 1290;
- Sequências: quando os vectores são uma série de números igualmente espaçados, é possivel criá-los usando o comando
  por exemplo u <− seq(from=1, to=5, by=1) obtendo-se
  [1] 1 2 3 4 5;
- Se fizermos v1 <− exp(u) obtemos
  [1] 2.718282 7.389056 20.085537 54.598150 148.413159.

Tipo de objecto Matrix

- Uma matrix no R pode ser constituida a partir dum vector fazendo por exemplo v<− c(1,2,3,4,5,6,7,8,9) e depois M<− matrix(v, nrow = 3, ncol = 3, byrow = TRUE)) obtendo-se

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

- Alternativamente pode-se usar as intruções

  v1<− c(1,2,3)
  v2<− c(4,5,6)
  v3<− c(7,8,9)
  M<− rbind(v1,v2,v3)

  para obter a mesma matriz, ou a matriz por colunas usando M<− cbind(v1,v2,v3).

Operações com matrizes

- Sendo A uma matriz 3x2 e B uma matriz 2x3, o produto matricial das duas é dado por
  C<− A%*%B;
- A transposta de uma matriz C é dada por
  Ctrans<− t(C);
- A inversa de uma matriz quadrada C é dada por
  Cinv<− solve(C);
- O determinante de uma matriz C é dado por
  Cdet<− det(C);
- Para extrair uma componente da matriz C, por exemplo a da linha 2 coluna 3 basta escrever C[2,3];
- Para extrair uma linha inteira da matriz C, por exemplo a 3ª linha basta escrever C[3,].

Tipo de objeto Data Frame

- Uma data frame é uma base de dados. Pode ser vista como uma matriz, com colunas de modos e atributos eventualmente diferentes. Cada coluna contém a informação sobre uma variável. Pode-se dispor na forma matricial, sendo as suas linhas e colunas acedidas pelas usuais convenções de índices.

- Exemplo - construção de uma data frame:

```
x1 <- 1 : 10
x2 <- 11 : 20
x3 <- letters[1 : 10]
d1 <- data.frame(x1, x2, x3); d1
attributes(d1)
```

Tipo de objeto Data Frame

- As data frames são uma boa forma de introduzir dados que se encontram num ficheiro exterior, por exemplo "Tabeladados", para dentro do R, através da função read.table( ):

  dados <− read.table("Tabeladados", header = TRUE)

- A opção header=TRUE usa-se quando as colunas dos dados no ficheiro têm cabeçalho com os nomes das quantidades que representam.

- O ficheiro "Tabeladados"tem de estar na pasta de trabalho. Se não estiver, há que indicar o caminho da sua localização, e.g. "c:/Dados/Tabeladados"

- Usando o R-studio os dados podem ser importados de forma mais automática usando a opção "Import Dataset".

A instrução if...else...

- if (condição){
  instruções
  }else{
  instruções
  }
  Uma condição é uma expressão lógica que utiliza operadores de comparação ou lógicos, cujo valor pode ser "TRUE"ou "FALSE";
- Exemplo:

```
x<- -1
if (x <=0){
    f<- x^2+3*x+1
}else{
    f<- 2*x^2+5*x
}
f
[1] -1
```

Tipo de objecto Funções

- Uma função tem de ter uma variável ou mais de entrada e um resultado de saída, os quais também podem ser vectores;
- Uma função f é criada fazendo simplesmente:
  f<− function(x) expressão;
- Exemplo para funções mais complexas com vários ramos:

```
classifica<− function(nota){
if (nota <9.5){
    decisão<− "reprovado"
}else{
    decisão<− "aprovado"}
return(decisão)
}
```

- A função classifica pode ser chamada mais tarde, fazendo por exemplo
  classifica(15)

Fazer o gráfico duma função

- Por exemplo, se definirmos a função
  f<− function(x) x∧3+2*x∧2+3*x+1
  podemos fazer o gráfico de f no intervalo, por exemplo [-10, 10], utilizando as instruções:
  x<− seq(from=-10, to=10, by=1/100)
  plot(x, f(x), type="l", col="red")
  O gráfico aparece na janela à direita em baixo.

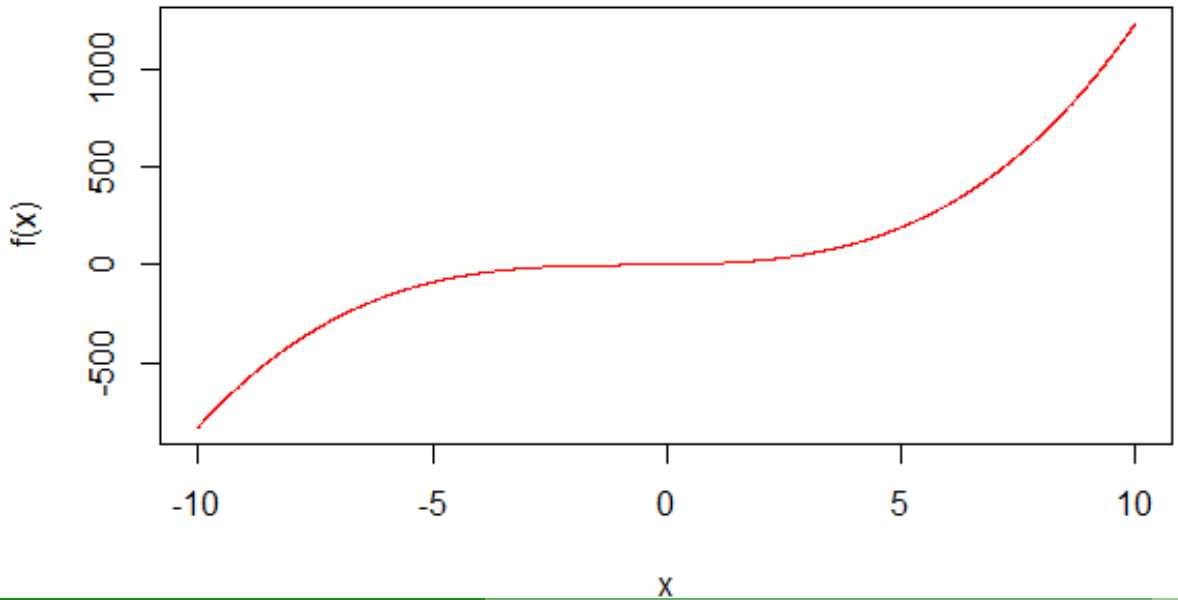


Gráfico de duas ou mais funções

- plot(x, f(x), type="l", col="red")
  lines(x, exp(x),col="green")
  abline(v=0)
  text(6, 120, "exp(x)")
  text(-9, -250, "f(x)")

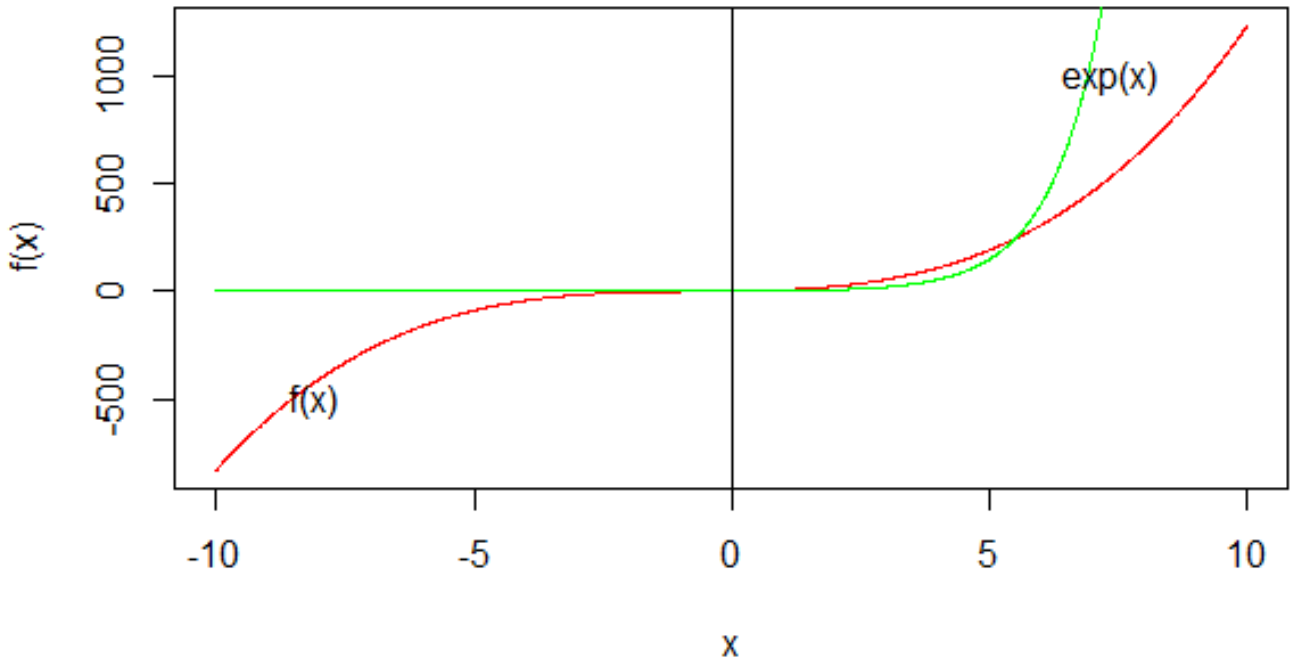
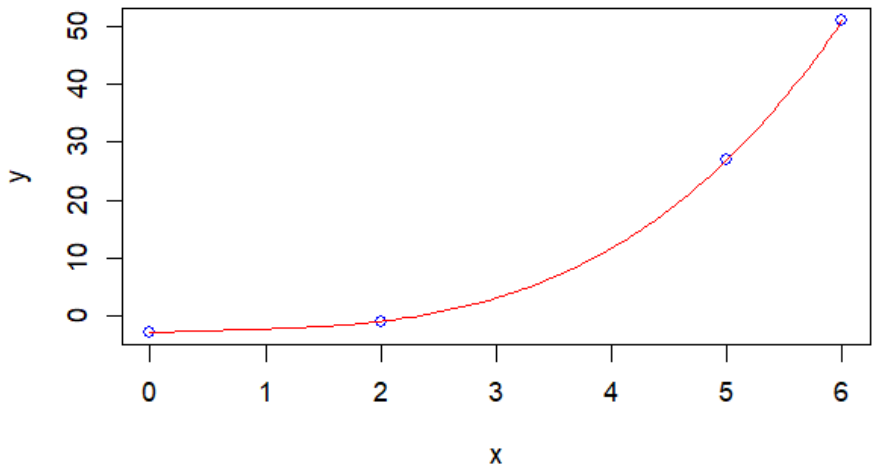


Gráfico de pontos com uma função

- x<− c(0,2,5,6)
  y<− c(-3,-1,27,51)
  plot(x,y,col="blue")
  f<− function(x) 1/3*x∧3-2/3*x∧2+x-3
  curve(f, col="red", add=T)



- Fazer o grafico duma data frame
  tabela=read.table("Tabeladados",header=T);tabela
  plot(tabela)

Instruções repetidas: ciclos for() e while()

- Se quizermos repetir as mesmas instruções i vezes com i=1,2...,n utilizamos:
  n<− 100
  for (i in 1:n){
  instruções
  }
  Em vez de 1:n pode usar-se um vector vec por nós antes definido, mas geralmente usa-se uma sequência de números inteiros consecutivos ou igualmente espaçados, por ex. vec<− seq(1, n, by = 2);
- Se quizermos repetir as mesmas instruções enquanto uma determinada condição se verificar, então utiliza-se:
  while(condição){
  instruções
  }.

Exemplo ciclo for

Calcular e imprimir a soma e o produto do coseno em 5 pontos distintos

```
n<- 5
x <- c(-pi/3, -2, 0 , 3, 2*pi)
soma<- 0
produto<- 1
for (i in 1:n){
soma<- soma + cos(x[i])
produto<- produto*cos(x[i])
}
cos(x)
resultado<- c(soma, produto)
print(resultado)
```

Na consola vai aparecer:

cos(x): [1] 0.5 -0.4161468365 1 -0.9899924966 1

print(resultado) : [1] 1.0938606669 0.2059911228

Instrução solve()

- Na resolução de sistemas de equações lineares utiliza-se a instrução solve(A,b)
  onde A é uma matriz quadrada e b é um vector. Exemplo:

```
v<- c(1,2,0,0,1,-1,3,1,1)
M<- matrix(v, nrow = 3, ncol = 3, byrow = TRUE)
b<- c(1,0,1)
M2 <-  solve(M,b)
M2
```

  resultado: [1] 0.0 0.5 0.5

Instrução polyroot()

- Para obter as raizes dum polinómio utiliza-se a instrução
  polyroot(p)
  onde p<− c(c1,c2,c3,...) é um vector cujas componentes são os coeficientes do termo independente (c1), do x (c2) do x^2 (c3), etc.
  Exemplo:

  pol<− function(x) 3*x∧2-35/2*x + 49/2
  p<− c(49/2,-35/2,3)
  polyroot(p)
  resultado:
  [1] 2.3333333333333339+0i 3.4999999999999996-0i

Bibliografia

- Os manuais sobre o R, incluidos em todas as instalações, são:
  - An introduction to R;
  - Writing R extensions;
  - R data import/export;
  - The R language definition;
  - R installation and administration.
- http://www.r-project.org/: FAC e The R Journal (http://journal.r-project.org/).
- Owen Jones, Robert Maillardet, Andrew Robinson (2014). Introduction to scientific programming and simulation using R, 2nd edition. CRC Press.
- Santos, F. Correia dos; Duarte, Jorge; Lopes, Nuno D., Fundamentos de Análise Numérica (Com Python3 e R), Edições Sílabo, 2019 (2ª edição).