Aula 1.1 - objetos e funções

Elder Sodré

Monday, February 23, 2015

### Objetos e funções

O R é orientado por objetos. Para um objeto existir, ele precisa ser criado com a seta para a direita ( <- )

5+7

## [1] 12

x <- 7  
5+x

## [1] 12

y<-2\*(x-3)

Quando um objeto é criado, o R não retorna o seu valor. Para ver um objeto que foi criado, basta digitar:

print(x)

## [1] 7

print(y)

## [1] 8

ou, simplesmente:

x

## [1] 7

y

## [1] 8

Quando um novo objeto é criado com o mesmo nome de um objeto preexistente, o objeto antigo é automaticamente apagado.

x <- -10  
y <- y^4

Um tipo especial de objeto no R são as funções. A estrutura básica de uma função é: funcao(argumento1,argumento2,argumento3)

Após o nome da função, sempre usa-se parênteses! Argumentos são os parâmetros de uma função. Eles sempre vem entre parênteses e são sempre separados por vírgulas.

Vamos dar uma olhada na função rnorm(), que cria um conjunto de números com distribuição normal. A função rnorm possui três argumentos:

* n: a quantidade de números desejada
* mean: a média dos números
* sd: o desvio padrão

Agora vamos testar a função:

rnorm(n=15,mean=-1,sd=12)

## [1] -24.2738 3.6702 -0.8439 -9.2734 -19.5579 -7.6045 -9.5413  
## [8] -19.6283 -3.9736 21.7542 2.6147 -8.9577 -3.5111 9.4852  
## [15] 21.5465

rnorm(n=2,mean=300,sd=3)

## [1] 298.8 297.2

rnorm(500,mean=0,sd=10)

## [1] 6.46780 1.17031 11.62642 -5.48510 5.10368 -1.08225 -9.68379  
## [8] 20.86617 -14.67320 2.14929 -2.24793 0.73969 -0.33717 12.68948  
## [15] 10.89639 -1.76322 8.37495 2.59483 14.16638 14.04239 3.55125  
## [22] 12.05787 8.23977 7.51653 -3.41949 -15.72685 -2.52785 11.19908  
## [29] -8.31778 -20.28640 24.26865 -3.44879 -11.33899 2.07523 0.57637  
## [36] 5.31554 -9.96305 -4.36328 3.99559 -7.03720 0.60623 -10.11041  
## [43] 1.53655 2.63232 -6.12915 3.01558 -9.66425 -2.72081 8.81053  
## [50] -13.62757 14.80110 8.79328 10.38639 3.32253 -1.24931 9.00900  
## [57] 2.90970 -8.37569 -16.26322 -6.73985 -2.10481 -1.05339 4.08408  
## [64] 8.74176 -17.93349 -0.56707 -7.70570 8.09621 -11.66863 0.42319  
## [71] -6.88689 -8.14798 -9.80646 -9.10843 13.36553 -6.96267 -6.70959  
## [78] 3.19723 24.61769 6.29986 9.42057 -5.93575 8.77581 7.56955  
## [85] -5.83730 12.44661 4.59428 6.58123 -18.54729 -1.96166 6.37695  
## [92] 9.70787 -12.74956 16.18909 -5.90683 11.57689 6.75813 -10.96518  
## [99] 0.82157 -13.78429 -4.21047 -9.44962 -6.59071 -5.19241 4.13488  
## [106] -16.85195 2.68962 -21.33193 0.21837 -11.05748 -12.92544 5.13597  
## [113] 23.39745 10.29641 4.77342 4.52890 19.32288 -12.00450 20.33380  
## [120] -6.85253 -13.97999 8.49462 3.44014 12.68932 4.60102 -9.41604  
## [127] 8.00887 0.76179 14.28896 -15.50517 -8.01761 14.94135 3.55527  
## [134] 7.15698 -22.33541 2.54569 4.08876 10.13142 -9.72438 1.70589  
## [141] -11.85735 -3.92419 -13.51196 1.52468 5.06688 15.33465 3.37757  
## [148] -9.41098 -7.67277 10.60418 3.85211 0.02849 8.51632 -4.65418  
## [155] 9.27005 -8.11506 0.18634 -4.19362 0.71652 0.80320 8.40432  
## [162] -17.41332 -6.63400 -9.88511 4.12613 10.04583 12.52666 1.51491  
## [169] -24.29599 -7.57880 -13.13189 4.92955 -1.58443 -5.66528 -19.08981  
## [176] 17.08210 -7.50890 16.04794 -16.39093 -1.65938 -8.57921 5.93588  
## [183] -16.21778 -18.74729 5.96668 0.51760 -1.00685 3.17564 -3.18865  
## [190] 9.62241 0.64379 3.34669 1.40289 6.70321 -6.01005 1.13146  
## [197] 5.45727 -17.64836 17.46113 6.65866 -3.79896 7.13035 1.14518  
## [204] 12.89264 -23.10169 -6.07554 6.67931 -9.21085 -9.91499 2.37118  
## [211] 1.25675 15.73349 18.74416 -7.41136 9.96553 5.86010 3.10086  
## [218] -20.05438 17.71301 3.55431 -1.28143 -9.80635 -1.01020 -16.26019  
## [225] -17.04535 4.30078 11.12874 -12.84472 6.64645 -5.99898 4.46812  
## [232] -10.06558 -13.58866 -2.38714 8.14815 -3.39350 15.55420 -11.85132  
## [239] -2.84902 4.34943 -2.06950 2.15186 7.71945 1.10417 12.00235  
## [246] -8.36585 -2.44548 8.00295 18.82662 2.76598 -14.40276 -1.38639  
## [253] -4.73602 1.44520 0.61791 3.88194 -17.57648 -16.38883 4.94101  
## [260] -13.35668 16.05212 -17.20994 -8.30075 -7.76420 -1.57398 18.36033  
## [267] -15.72471 8.48896 5.66028 8.99629 -2.87430 4.40633 22.48715  
## [274] -0.74012 -23.65023 12.87229 -2.52980 -7.41965 -0.13134 19.74712  
## [281] 0.73837 -0.42219 2.47717 8.70335 -2.12301 -13.76655 -4.32011  
## [288] -1.25278 -5.10006 8.46351 -6.07078 -5.17875 3.98025 11.59403  
## [295] 18.91415 -6.60080 -10.12138 -15.10733 6.30572 -8.71966 4.18455  
## [302] -2.50959 -14.02003 21.50709 -9.54006 -21.88099 2.58871 23.63980  
## [309] 8.23191 -12.32857 -7.13698 5.03863 -4.90932 3.77054 3.28678  
## [316] 0.73286 -2.26005 10.83864 -8.05998 -1.72845 2.12551 8.53038  
## [323] -0.39224 -5.45725 1.67101 7.11809 -0.21612 -5.39350 -10.71362  
## [330] 11.65272 10.67172 -4.01802 3.47846 1.10782 8.07546 -3.70333  
## [337] -24.19344 17.14165 -7.46869 2.83910 -9.24940 -17.47782 -23.26891  
## [344] 9.56038 2.92160 2.32324 2.44949 -8.42771 -2.76020 6.46605  
## [351] 4.39223 8.16952 5.83557 4.80150 -2.31540 -3.37879 -9.14478  
## [358] 9.99993 -6.79440 5.54716 -2.44011 1.29575 3.14001 -3.30290  
## [365] -16.18932 -2.72032 -8.90866 22.39521 -2.56214 8.11346 -6.96910  
## [372] -2.09005 -16.04303 14.42847 6.00994 5.81826 -5.70160 18.42597  
## [379] 3.60302 -2.22960 0.88213 -0.84032 -12.14771 -5.59169 3.36448  
## [386] -10.15751 -13.92462 -24.07480 -4.35219 -8.39451 -6.39069 13.52571  
## [393] 11.42357 11.59484 7.19373 5.13373 16.76331 13.34414 9.11178  
## [400] -7.80190 3.65543 -0.76366 4.88789 0.09023 0.92039 -4.29564  
## [407] 6.56111 4.67945 -1.33865 -10.41449 -3.52147 -0.04734 -2.16858  
## [414] 0.71080 -13.24401 -13.79757 6.84761 0.45579 3.38313 -12.57647  
## [421] -4.49748 17.70905 3.35250 19.46373 3.03815 18.63156 13.15648  
## [428] -8.23460 11.49040 -15.25374 -5.15700 14.43549 -11.39780 -6.67937  
## [435] 14.81158 -2.09635 -12.69309 11.43396 9.01925 3.11443 2.49092  
## [442] 10.63050 14.82258 4.24545 21.20267 -8.28008 -18.10338 -23.79338  
## [449] 4.53715 1.54668 13.92399 -3.71732 4.67433 -4.10764 -11.77813  
## [456] -6.43809 -7.90284 -4.19177 16.68868 14.14998 3.15147 -16.30496  
## [463] 12.49592 -9.32186 -18.41415 1.18198 16.76543 -2.26368 15.41880  
## [470] -13.21430 -11.74136 -2.28967 17.84392 -1.24586 15.21522 3.09794  
## [477] -0.88996 -1.11315 7.44733 1.48602 -2.24007 -0.53823 -8.53000  
## [484] -9.44480 -0.25790 -6.01718 -9.13605 3.96278 3.23470 18.25197  
## [491] -5.03759 9.12430 -8.12864 -3.60602 10.32273 8.14364 17.03843  
## [498] 5.54908 2.69686 4.99950

Lembre-se: quando você usa uma função, o R retorna o resultado mas não salva o resultado. Para salvar um resultado, lembre-se de criar um objeto!

Vamos criar objetos para guardar os resultados:

a <- rnorm(n=15,mean=-1,sd=12)  
b <- rnorm(n=2,mean=300,sd=3)  
c <- rnorm(n=500,mean=0,sd=10)

Observe que, como a função rnorm() cria números aleatórios, cada vez que ela é chamada cria um conjunto diferente de números. Mas o objeto criado armazena uma sequência específica.

rnorm(n=10,mean=0,sd=500)

## [1] 146.60 279.56 -236.42 27.42 512.44 357.85 569.69 -634.34  
## [9] 534.53 -356.69

rnorm(n=10,mean=0,sd=500)

## [1] 2.877 361.673 -266.702 320.505 -272.884 246.150 -280.900  
## [8] 337.669 -232.903 4.274

rnorm(n=10,mean=0,sd=500) #Três sequências diferentes

## [1] 630.65 142.42 -1043.68 -32.37 139.87 242.86 175.55  
## [8] -262.36 -131.79 -577.60

d <- rnorm(n=10,mean=0,sd=500)  
d

## [1] 158.33 524.50 623.12 265.10 589.10 -19.69 377.77 -425.49  
## [9] -238.06 -71.73

d

## [1] 158.33 524.50 623.12 265.10 589.10 -19.69 377.77 -425.49  
## [9] -238.06 -71.73

d #A mesma sequência três vezes

## [1] 158.33 524.50 623.12 265.10 589.10 -19.69 377.77 -425.49  
## [9] -238.06 -71.73

Não acredita na função rnorm()? Use as funções mean() e sd() para conferir se a média e desvio padrão dos números está correta.

mean(a)

## [1] -4.162

sd(a)

## [1] 7.636

mean(b)

## [1] 300.4

sd(b)

## [1] 0.01651

mean(c)

## [1] -0.3618

sd(c)

## [1] 10.05

mean(d)

## [1] 178.3

sd(d)

## [1] 361.1

#### Mais algumas funções úteis

Às vezes, é necessário saber quais objetos estão salvos.Outras vezes, é necessário excluí-los. Para isso servem as funções ls() e rm()

ls() #Retorna o nome dos objetos criados.Esta função não tem argumentos.

## [1] "a" "b" "c" "d" "metadata" "x"   
## [7] "y"

rm("objeto") #Exclui o objeto chamado objeto.

## Warning: objeto 'objeto' não encontrado

rm(list=c("a","b")) #Exclui os objetos "a" e "b".  
rm(list=ls()) #Exclui todos os objetos criados.

#### Obtendo ajuda

A COISA MAIS IMPORTANTE QUE EU POSSO ENSINAR A VOCÊS É COMO BUSCAREM AJUDA SOZINHOS!!!

Uma maneira de buscar ajuda é com a função help().

help(rnorm)

## starting httpd help server ... done

Outra forma mais simples, que tem o mesmo resultado é: ?função

?rnorm

Se você não lembra o nome da função, use dois pontos de interrogação:

??sequence

Outras funções interessantes para pedir ajuda:

help.start() #Ajuda geral

## If nothing happens, you should open  
## 'http://127.0.0.1:13797/doc/html/index.html' yourself

apropos("nor") #Lista todas as funções que contém "nor"

## [1] "dlnorm" "dnorm" "norm" "normalizePath"  
## [5] "plnorm" "pnorm" "qlnorm" "qnorm"   
## [9] "qqnorm" "rlnorm" "rnorm"

args(rnorm) #Argumentos da função rnorm

## function (n, mean = 0, sd = 1)   
## NULL

example(mean) #Mostra um exemplo da função mean

##   
## mean> x <- c(0:10, 50)  
##   
## mean> xm <- mean(x)  
##   
## mean> c(xm, mean(x, trim = 0.10))  
## [1] 8.75 5.50

Além disso, não deixe de procurar ajuda na internet se for necessário. Alguns sites recomendados:

* <http://www.statmethods.net/> Contém diversos tutoriais
* <http://www.rseek.org/> Muito útil. Pode ser considerado o google do R
* <http://stackoverflow.com/>; <http://pt.stackoverflow.com/> Site de perguntas e respostas para programadores profissionais ou amadores
* <https://www.google.com.br/> Google. Dispensa apresentações