

Funções na linguagem R

Ivanildo Batista

16 de março de 2021

FUNÇÕES NA LINGUAGEM R

O R possui funções internalizou funções Built-in: `abs()`, `sqrt()`, `prod()`, `rev()`, `c()` e `contributors()`

FUNÇÕES

```
#sample() #gera uma amostra aleatória  
args(sample) #mostra o parâmetro das funções
```

```
## function (x, size, replace = FALSE, prob = NULL)  
## NULL
```

```
args(mean)
```

```
## function (x, ...)  
## NULL
```

```
args(sd)
```

```
## function (x, na.rm = FALSE)  
## NULL
```

Built-in

```
abs(-43) #valor em módulo
```

```
## [1] 43
```

```
sum(c(1:5)) #soma os valores
```

```
## [1] 15
```

```
mean(c(1:5)) #tira a média
```

```
## [1] 3
```

```
round(c(1.1:5.8)) #arredonda valores
```

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

```
rev(c(1:5))#pega um vetor e inverte a posição dos seus valores
```

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

```
seq(1:5)#gera a sequência de valores
```

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

```
sort(rev(c(1:5)))#reverte o função rev()
```

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

```
append(c(1:5),6) #insere um valor em um vetor
```

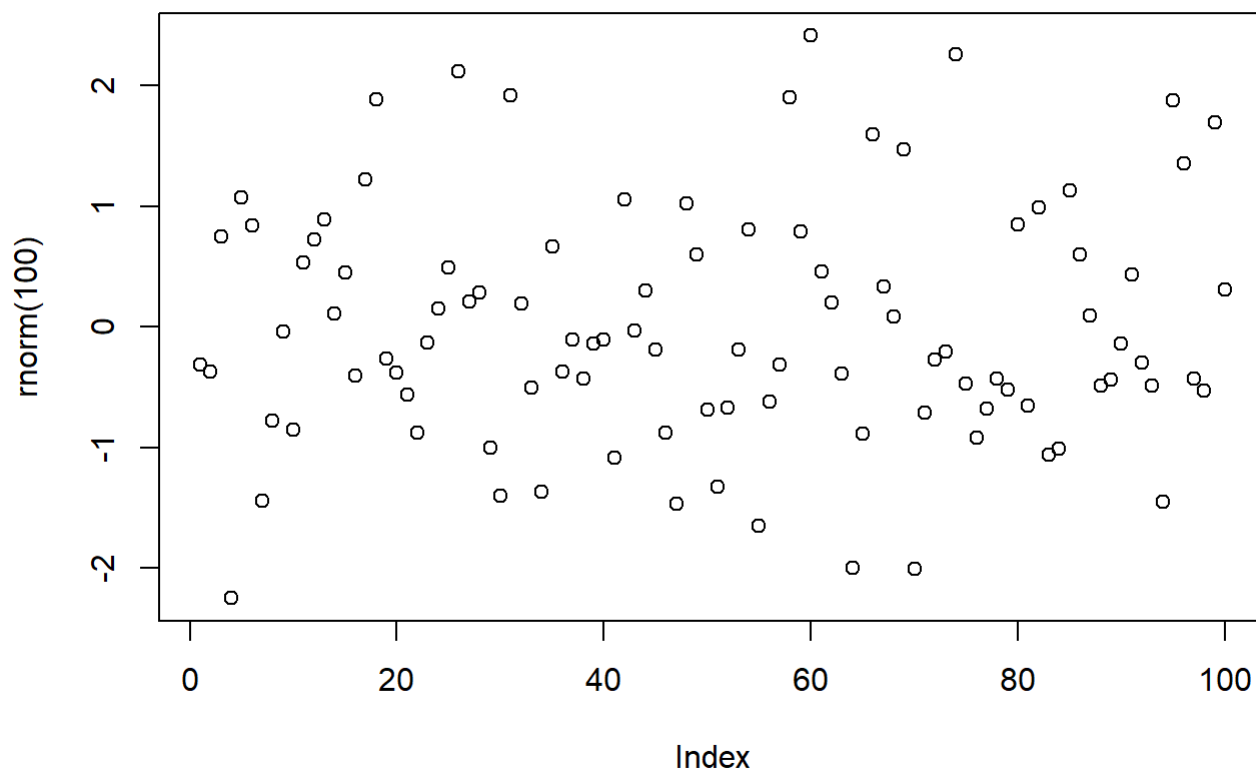
```
## [1] 1 2 3 4 5 6
```

```
vec1 = c(1.5,2.5,8.4,3.7,6.3)  
vec2=rev(vec1)  
vec2
```

```
## [1] 6.3 3.7 8.4 2.5 1.5
```

Funções dentro de funções

```
plot(rnorm(100)) #gerando valores com uma distribuição normal
```



```
mean(c(abs(vec1),abs(vec2))) #média entre o módulo de vetores
```

```
## [1] 4.48
```

Criando funções

```
myfunc = function(x){x+x} #criando uma função que soma valores
myfunc(10) #usando a função
```

```
## [1] 20
```

```
class(myfunc) #verificando a classe do objeto gerado
```

```
## [1] "function"
```

```
myfunc2 = function(a,b){
  valor=a^b
  print(valor)
} #gerando outra função com potência

myfunc2(13,2)
```

```
## [1] 169
```

Jogando dados

```
jogando_dados=function(){  
  num = sample(1:6, size=1)  
  num #essa é uma variável local  
}  
  
jogando_dados()
```

```
## [1] 5
```

Escopo

```
num=c(1:6)  
  
print(num)
```

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

```
num #essa é uma variável global
```

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

Funções em argumentos

```
vec1 = (10:13)  
vec2 = c('a','b','c','d')  
vec3 = c(6.5,9.2,11.9,5.1)  
  
myfunc3 = function(...){  
  df=data.frame(cbind(...))  
  print(df)  
}  
  
myfunc3(vec1)
```

```
##   vec1  
## 1   10  
## 2   11  
## 3   12  
## 4   13
```

```
myfunc3(vec1,vec2)
```

```
##   vec1 vec2
## 1   10   a
## 2   11   b
## 3   12   c
## 4   13   d
```

```
myfunc3(vec1,vec2,vec3)
```

```
##   vec1 vec2 vec3
## 1   10   a  6.5
## 2   11   b  9.2
## 3   12   c 11.9
## 4   13   d  5.1
```

```
myfunc3(vec1,vec3)
```

```
##   vec1 vec3
## 1   10  6.5
## 2   11  9.2
## 3   12 11.9
## 4   13  5.1
```

```
myfunc3(vec2,vec3)
```

```
##   vec2 vec3
## 1    a  6.5
## 2    b  9.2
## 3    c 11.9
## 4    d  5.1
```