Funções e programação para iniciantes

Vitor Rios

11 de novembro de 2017

Que raios é uma função?

Funções são objetos que realizam uma tarefa, como ferramentas. Retomando nossa analogia da cozinha, se o script é a receita, o R é a cozinha e a análise é o jantar, os dados seriam os ingredientes e as funções seriam os utensílios e eletrodomésticos, usados para transformar os ingredientes crus, processar, espremer, cozinhar, e decorar para apresentação.

(você é o(a) cozinheiro(a))
(e faxineiro(a), e garçom, e cliente)

Em termos técnicos

Uma função é uma série comandos que realizam uma determinada função / análise / operação, empacotados com um nome seguido de parênteses, que pode ser chamada várias vezes no código

```
read.table()
sum()
```

Quando uma função gera um resultado na forma de objeto, dizemos que ela *retorna* um objeto

Geralmente as funções no R ou retornam um valor, ou retornam uma mensagem com o resultado (erro ou sucesso)

Princípio do não se repita (regra de três)

Se você realiza a mesma operação ou série de operações mais de três vezes num mesmo script, considere transformar isso numa função

Vantagens de empacotar o código em funções:

- Não se repetir: diminui erros de digitação e de copiar/colar
- Encapsulamento: cada parte do código existe separadamente, ficando mais fácil de encontrar onde está o erro
- Portabilidade: maior facilidade de compartilhar código e usar em outras análises
- Velocidade: se você tem suas funções de análise, gráficos ou transformação de dados prontas e funcionando, não precisa escrever elas novamente
- Legibilidade: usando nomes de funções que descrevam as operações, fica mais fácil de ler o código

Partes da função

A função é composta por um bloco de código, criado usando a função `function()`

```
erro.padrao.media = function(x) {
  sd(x)/sqrt(length(x))
}
```

- `erro.padrao.media` é o nome do objeto que contém a função, ou o nome da função. Use um nome descritivo para os seus objetivos
- `= function(x)`esta parte tem 3 propósitos:
- Primeiro, o operador de atribuição = indica que tudo que vier depois vai ser jogado dentro do objeto `erro.padrao.media`.
- Segundo, `function` indica que o que vier dentro das chaves a seguir é um função.
- Terceiro, `(x)` indica que a função irá receber um objeto, e que o nome desse objeto dentro da função será `x`. Dizemos a função recebe um `argumento`, ou `parâmetro`, ou que `x é o argumento da função`.

Argumentos ou parâmetros são informações que você dá à função para que ela possa fazer seu serviço. No nosso caso, `x` é um vetor que vai ser usado nas funções `sd()` e `length()`, que são chamadas de dentro de `erro.padrao.media`

Funções podem ter mais de um argumento. Toda informação que a função precisa para fazer seu trabalho deve vir na forma de argumentos

Argumentos podem ter valores padrão. Se o usuário não definir o valor do argumento, a função usa o valor padrão. Argumentos sem valor padrão são obrigatórios, isto é, o usuário deve definir o valor na chamada da função

Para definir um valor padrão, use `nome.da.funcao = function(argumento1, argumento2 = valor.padrao)`

`aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)`

`x` an R object.

`by` a list of grouping elements, each as long as the variables in the data frame `x`. The elements are coerced to factors before use.

`FUN` a function to compute the summary statistics which can be applied to all data subsets.

`...` further arguments passed to or used by methods.

`simplify` a logical indicating whether results should be simplified to a vector or matrix if possible.

`drop` a logical indicating whether to drop unused combinations of grouping values. The non-default case drop=FALSE has been available since R 3.3.0, and may change in some cases where unused combinations are still dropped.

Retornando valores

```
erro.padrao.media = function(x) {
  sd(x)/sqrt(length(x))
}
```

Por padrão, funções no R retornam o resultado da última operação feita. Na nossa função, o valor de retorno é o resultado de `sd(x)/sqrt(length(x))`.

Se não quisermos retornar a última operação, podemos definir um valor de retorno usando a função `return()`. Nesse caso, o valor retornado será o que estiver dentro dos parênteses de `return`

```
erro.padrao.media.inutil = function(x) {
  sd(x)/sqrt(length(x))
  return(print("erro padrão calculado. Tenha um bom dia"))
}
exemplo = rnorm(n=100, mean = 50, sd = 15)
erro.padrao.media.inutil( x = exemplo)

## [1] "erro padrão calculado. Tenha um bom dia"
```

O R só pode retornar um único objeto.

Se quisermos mais de um resultado, podemos usar um vetor, dataframe ou lista (ou definir uma classe especial, mas isso foge ao escopo do curso)

```
summary.de.pobre= function(x) {
  minimo=min(x)
  primeiroquartil=quantile(x,.25)
  mediana = median(x)
  media = mean(x)
  terceiroquartil= quantile(x,.75)
  maximo=max(x)
  erro.padrao=sd(x)/sqrt(length(x))

  tudoJunto =list(
    "Min."      = minimo,
    "1st Qu"    = primeiroquartil,
    "Median"    = mediana,
    "Mean"      = media,
    "3rd Qu."   = terceiroquartil
```

```r
exemplo = rnorm(n=100, mean = 50, sd = 15)
summary.de.pobre( x = exemplo)
```

```
## $Min.
## [1] 11.6494
##
## $`1st Qu`
##      25%
## 44.47154
##
## $Median
## [1] 50.09195
##
## $Mean
## [1] 50.92708
##
## $`3rd Qu.`
##      75%
## 60.40296
```

```r
exemplo = rnorm(n=100, mean = 50, sd = 15)
exemplo2 = rpois(n = 100,lambda = 50)
exemploJunto= data.frame("preco.do.chuchu"=exemplo,"preco.do.arroz"=exemplo
summary.de.pobre( x = exemploJunto)
summary(exemploJunto)
```

Como planejar e construir sua função

Lembre-se: alguém já teve esse problema antes. Procure por um pacote ou função que você possa usar ou adaptar antes de reinventar a roda.O melhor jeito de pegar prática com funções é pegar uma função simples que já está pronta e comentar cada linha. Depois, altere a função de pouco em pouco, vendo o que ocorre (ou não ocorre) e porquê

0 - Não entre em pânico

1 - Defina o que ela tem que fazer

2 - Defina que tipo de dados ela precisa aceitar, e que informaçãoes ela precisa (em outras palavras, que parâmetros ela precisa)

3 - Defina o que ela tem de retornar

4 - Escreva o pseudocódigo / fluxograma

5 - Escreva seu código, comentando cada linha

6 - Conserte seus erros

7 - Teste com dados reais

Debug is on the table

Debugging ou debugar é o ato de procurar, encontrar e consertar os erros no seu código

Seu código vai ter erros

(a não ser que você seja Don Knuth)

(e mesmo que você seja, seu código vai ter erros)

"Code drunk. Debug sober" - Anônimo

Programação em pares: uma pessoa programa enquanto a outra observa e discute, depois trocam de lugar

Code review: envie seu código pra alguém ler e comentar

Como debugar

- Tome uma dose cavalar de café
- Leia as mensagens de erro do R e procure no Google
- Saiba o que sua função deve fazer e o que ela deve retornar
- Tome outra dose cavalar de café
- Execute seu código linha por linha, observando o resultado de cada operação. Ela está fazendo o que deveria? O valor dos objetos está correto?
- Explique seu código em voz alta para seu objeto inanimado favorito
- Explique seu código para uma pessoa, de preferência tomando doses cavalares de café
- Explique seu código para uma pessoa que saiba mais que você, de preferência tomando doses cavalares de café
- Faça uma oferenda à sua divindidade preferida (ou duas, pra garantir)
- Chore

Trabalho Final

Escrever uma função, com as seguintes condições:

Deve resolver um problema, de preferência relacionado à sua pesquisa

Deve ser acompanhada de um Help.txt, em português ou inglês, com a estrutura padrão do R (veja exemplo aqui), e um conjunto de dados com o qual ela deve funcionar

Deve ter dentro do código pelo menos ou um if(), ou um for() ou uma das funções da família apply.
Todas as linhas devem ter um comentário, dizendo o que elas fazem . Sim, todas. Sim, mesmo que seja óbvio

Prazo: duas semanas após o fim das aulas, dia 21/12 às 23:59:59, sem possibilidade de extensão

Siga as dicas do site do IB-USP para a construção da sua função Disciplina R IB-USP

Forum disciplina R IB-USP

Forum da nossa disciplina

Trabalho final

Tragam a proposta da sua função para última aula

A função deverá ser entregue via email, ou postada em um repositório GitHub e o link enviado para o email da disciplina (preferido)

Repetindo: o trabalho é composto de um script com o código da função, um arquivo de texto com o help da função, e um conunto de dados de exemplo

Após fim das aulas, os monitores não estarão mais disponíveis para dúvidas presenciais. Usem o fórum da disciplina para tirar dúvidas, pois seu problema com certeza é o mesmo de outro coleguinha

Nota final: exercícios (40%) + Função (60%)

Não basta a função rodar, ela tem de funcionar direito e cumprir o que promete

- `if(panico == TRUE){message("Não entre em pânico")}`