

Creating functions

Jai Jot Kaur

11/8/2022

Introduction to creating functions

Why is creating your own functions in R a good practice?

Creating your own functions in R is a good practice because it allows the code to be more organized and neat so that it is easy to understand not only for yourself, but also for your readers. It is also a good practice because creating the functions on your own allows you to have a deeper understanding of what is being done, so if any issues were to arise, they may be easier to debug. It also allows your code to be used for further projects that may arise in the future since the functions have been created by you.

In class discussion:

- It allows reusing code for other parts of a project or a future project
- It is less error prone than copy pasting code
- Code is more organized for you and others, so it is easier to read
- Allows deeper understanding of what our code is doing
- Useful to add data to data frames
- Allows getting specific values and requirements
- Invites you to be intentional on the code you are writing
- Allows to be more in control of the outputs

Exercise 1: Writing Functions

Copy the following function into an R chunk of your Rmd file and replace the spaces _____ with variables names for the input and output.

```
convert_pounds_to_grams <- function(pounds) {  
  grams = 453.6 * pounds  
  return(grams)  
}
```

Use the function to calculate how many grams there are in 3.75 pounds.

```
convert_pounds_to_grams_ex <- function(pounds = 3.75) {  
  grams = 453.6 * pounds  
  return(grams)  
}
```

```
convert_pounds_to_grams_ex() # this requires a default value in the function # without default value
```

```
## [1] 1701
```

```
convert_pounds_to_grams_ex(3.75)
```

```
## [1] 1701
```

```
convert_pounds_to_grams_ex(pounds = 3.75)
```

```
## [1] 1701
```

Exercise 2: Use and Modify Functions

The following function estimates the mass of an organism in kg based on its length in meters for a particular set of parameter values a and b, those for Theropoda dinosaurs, where a has been estimated as 0.73 and b has been estimated as 3.63 (Seebacher 2001).

```
get_mass_from_length_theropoda <- function(length){  
  mass <- 0.73 * length ^ 3.63  
  return(mass)  
}
```

Use this function to print out the mass of a Spinosaurus that is 16 m long based on its reassembled skeleton.

```
get_mass_from_length_theropoda(length = 16)
```

```
## [1] 17150.56
```

Create a new version of this function called `get_mass_from_length()` that estimates takes length, a and b as arguments and uses the following code to estimate the mass $mass <- a * length^b$. Use this function to estimate the mass of a Sauropoda ($a = 214.44$, $b = 1.46$) that is 26 m long.

```
get_mass_from_length <- function(length){  
  mass <- a * length ^ b  
  return(mass)  
}
```

```
a = 214.44
```

```
b = 1.46
```

```
get_mass_from_length(26)
```

```
## [1] 24955.54
```

```
get_mass_from_length <- function(length){  
  mass <- 214.44 * length ^ 1.46  
  return(mass)  
}
```

```
get_mass_from_length(26)
```

```
## [1] 24955.54
```

*** Investigate why R

Exercise 3: Default Arguments

Rewrite your `get_mass_from_length()` function from Use and Modify so that its arguments have default values of $a = 39.9$ and $b = 2.6$ (the average values from Seebacher 2001).

```
get_mass_from_length <- function(length, a = 39.9, b = 2.6){  
  mass <- a * length ^ b  
  return(mass)  
}
```

Use this function to estimate the mass of a Sauropoda ($a = 214.44$, $b = 1.46$) that is 22 m long (by setting a and b when calling the function).

```
get_mass_from_length <- function(length, a = 39.9, b = 2.6){  
  mass <- a * length ^ b  
  return(mass)  
}  
get_mass_from_length(22, 214.44, 1.46)
```

```
## [1] 19554.33
```

```
# 19554.33
```

Use this function to estimate the mass of a dinosaur from an unknown taxonomic group that is 16m long. Only pass the function `length`, not a and b , so that the default values are used.

```
get_mass_from_length <- function(length, a = 39.9, b = 2.6){  
  mass <- a * length ^ b  
  return(mass)  
}  
get_mass_from_length(16)
```

```
## [1] 53911.93
```

```
# 53911.93
```

Exercise 4: Combining Functions

Write a function called `convert_kg_to_pounds` that converts kilograms into pounds (pounds = $2.205 * \text{kg}$).

```
convert_kg_to_pounds <- function(kg) {  
  pounds = 2.205 * kg  
  return(pounds)  
}
```

Use that function and your `get_mass_from_length()` function from Default Arguments to estimate the weight, in pounds, of a 12 m long Stegosaurus with $a = 10.95$ and $b = 2.64$ (The estimated a and b values for Stegosauria from Seebacher 2001).

```
library(magrittr)
weight_in_pounds <- get_mass_from_length(12, 10.95, 2.64) %>%
  convert_kg_to_pounds()
weight_in_pounds
```

```
## [1] 17055.37
```

```
# 17055.37
```