functions-prep

Davinder Singh

2023-03-15

Writing your own function in R

writing a function how it works

```
function_name <- function(inputs) {
  ouput_value <- do_something(inputs)
  return(output_value)
}
```

example of brackets

```
{
  a =2
  b =3
  a + b
}
```

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

example of how function work to find volume

```
calc_shrub_vol <- function(length, width, height) {
  area <- length * width
  volume <- area * height
  return(volume)
}
calc_shrub_vol(0.8, 1.6, 2.0)
```

```
## [1] 2.56
```

```
shrub_vol <- calc_shrub_vol(0.8, 1.6, 2.0)
```

How Functions Excute

function creates its own internal environment once code is complete you can't use anything from the function

```
calc_shrub_vol <- function(length, width, height) {
  area <- length * width
  volume <- area * height
  return(volume)
}
calc_shrub_vol(0.8, 1.6, 2.0)
```

```
## [1] 2.56
```

```
shrub_vol <- calc_shrub_vol(0.8, 1.6, 2.0)
```

Setting Default Values for Arguments

if you want to set a value for a function set it a value. ex (a,b, c =)

```
calc_shrub_vol <- function(length, width, height =1) {
  area <- length * width
  volume <- area * height
  return(volume)
} # in video they did height = 1 to set the height to 1 meter
calc_shrub_vol(0.8, 1.6, 2.0)
```

```
## [1] 2.56
```

```
calc_shrub_vol(0.8,1.6) # since height is not added the default function is multuplied in this case 1
```

```
## [1] 1.28
```

When to use named and unnamed arguments

when assigned it gets same value because of function

```
calc_shrub_vol <- function(length, width, height =1) {
  area <- length * width
  volume <- area * height
  return(volume)
}
calc_shrub_vol(0.8, 1.6, 2.0)
```

```
## [1] 2.56
```

```
calc_shrub_vol(0.8,1.6)
```

```
## [1] 1.28
```

```
calc_shrub_vol(length = 0.8, width = 1.6, height =  2.0)
```

```
## [1] 2.56
```

```
calc_shrub_vol(height =  2.0, length = 0.8, width = 1.6) # cause function is l * W * H
```

```
## [1] 2.56
```

```
calc_shrub_vol(0.8, 1.6, height = 2.0) # change height
```

```
## [1] 2.56
```

Combining Functions

another way to get mass using intermediate variables

```
est_shrub_mass <- function(volume) {
  mass <- 2.65 * volume^0.9
  return(mass)
}
shrub_volume <- calc_shrub_vol(0.8, 1.6, 2.0)
shrub_mass <- est_shrub_mass(shrub_volume)
```

pipes another way

```
shrub_mass <- calc_shrub_vol(0.8, 1.6, 2.0) %>%
  est_shrub_mass()
```

nesting functions

```
shrub_mass <- est_shrub_mass(calc_shrub_vol(0.8, 1.6, 2.0))
```

Calling functions from inside other functions

using functions inside other functions example

```
est_shrub_mass_dim <- function(length, width, height = 1) {
  volume <- calc_shrub_vol(length, width, height)
  mass <- est_shrub_mass(volume)
  return(mass)
}
est_shrub_mass_dim(0.8, 1.6, height = 2.0)
```

```
## [1] 6.175354
```

Rstudio tips and tricks for working with functions

Top level to jump around code

arrows to collapse code to hide code

highlight function in tools -> global options -> display "Highlight R function calls" functions calls are gray, variables are black, key words are blue