PROG102: Functions

Writing your own functions in R

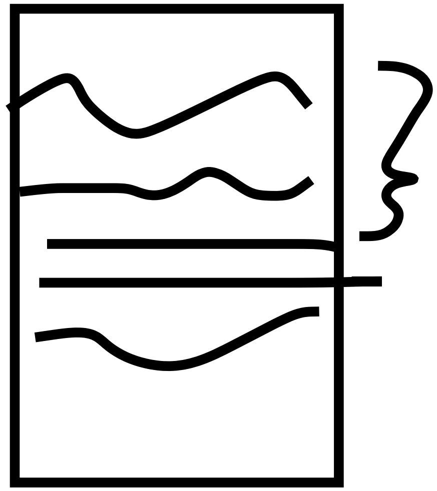
MARINCS 100B | Intro to Marine Data Science | Winter 2025

Key concepts

Functions have two purposes:
1) hide the details *encapsulation*
2) Apply the same code to new inputs *reusability*

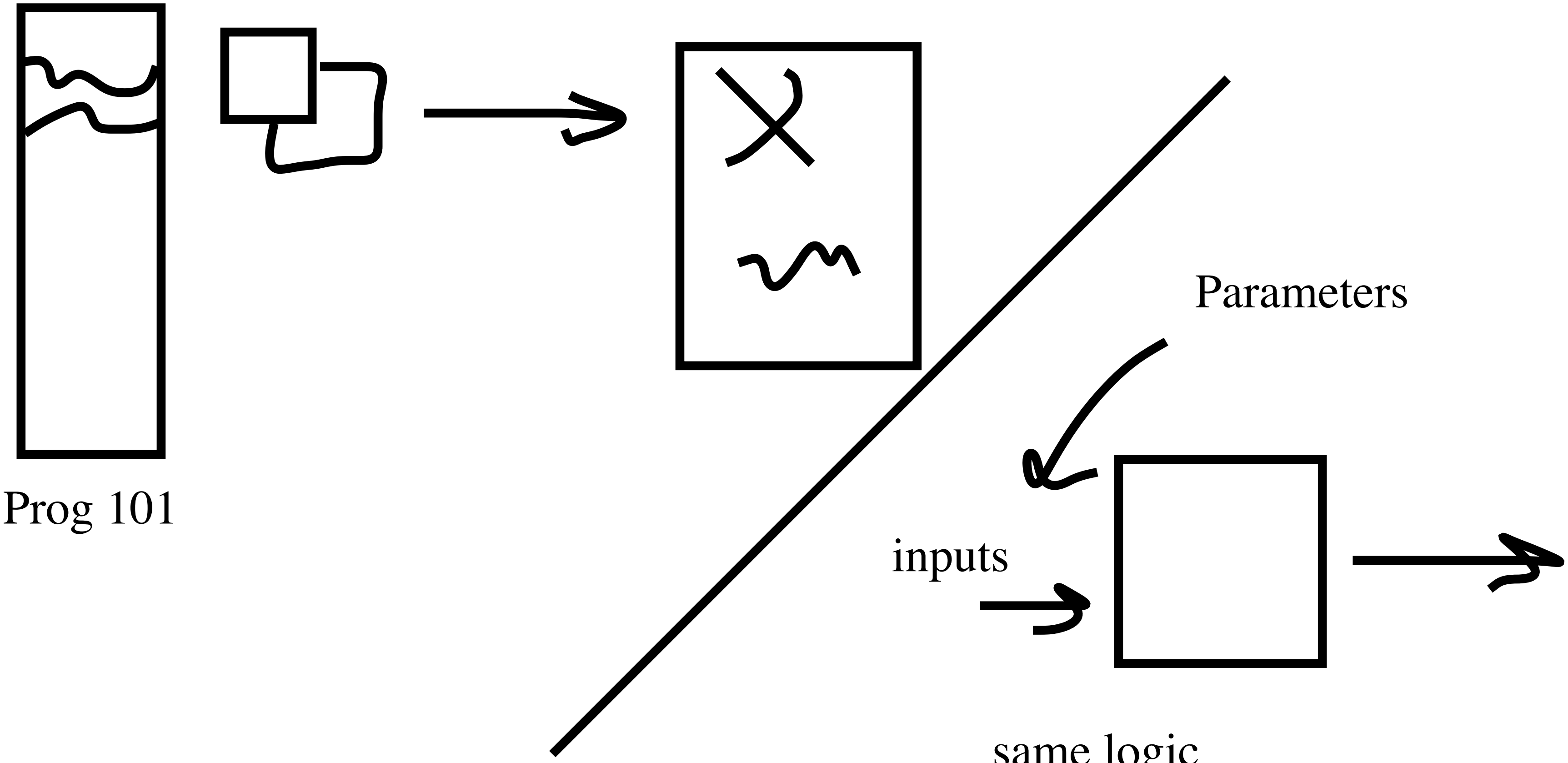
Easy to read

Cognitive load
<7 items at a time



Reusable

Copy paste edit work flow



Prog 101

Parameters

inputs

ouputs

same logic
new inputs new outputs

Syntax

Functions have five parts

1) name
2) keyword function
3) parameters (parentheses)
4) body in curly brackets
5) return output

Demo in R

Recap

Functions make code readable by hiding details (encapsulation)
Functions make code reusable by allowing different inputs (parameters)
syntax- every function definition has 5 parts

New vocabulary and lingering questions

New vocabulary

Lingering questions

Exercises

Label the five parts of this function:

```
first_and_last <- function(s) {
  first_char <- substr(s, 1, 1)
  last_char <- substr(s, nchar(s), 1)
  result <- paste(first_char, last_char)
  return(result)
}
```

1) assigns first_and_last to be the function name

2) makes first character

Exercises

Match the function bodies on the left with the name that describes what they’re doing on the right.

```
function(x) {
  result <- x + 1
  return(result)
}
```

```
function(a) {
  result <- a * 2
  return(result)
}
```

```
function(a, b) {
  c_squared <- a^2 + b^2
  result <- sqrt(c_squared)
  return(result)
}
```

```
double

hypotenuse_length

increment
```

Exercises

Write a function that turns a vector into a palindrome. For example, it should turn 1 2 3 into 1 2 3 3 2 1. Hint: you'll have to use a function called `rev()`. Choose a short but descriptive name for your function.

PROG102: Functions

How functions execute

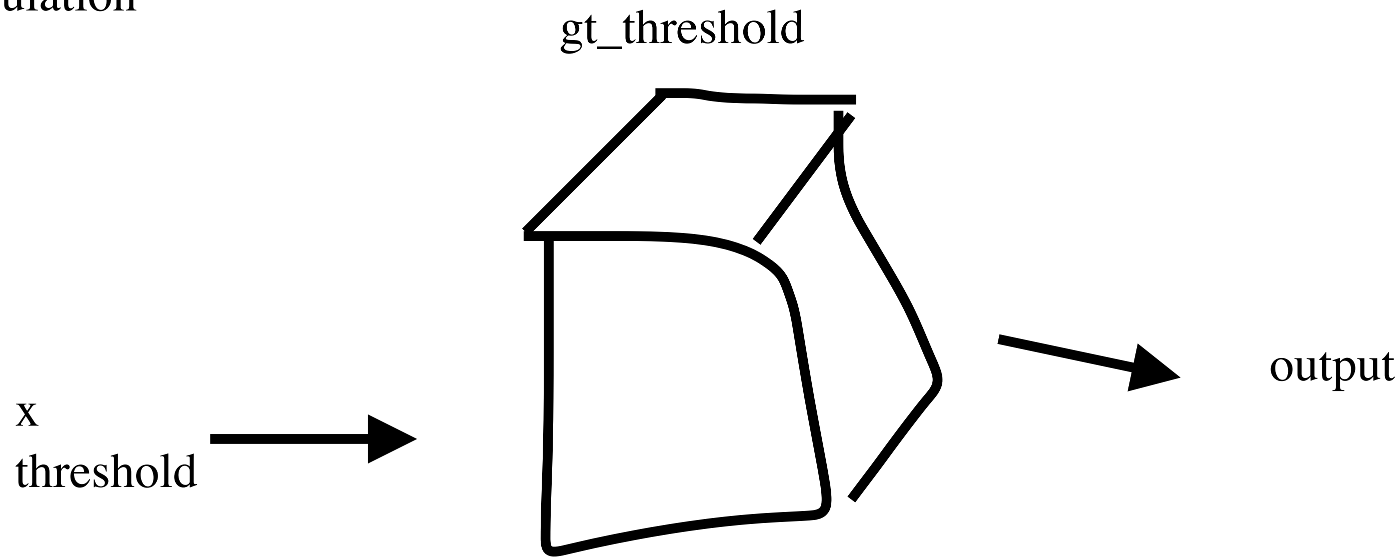
MARINCS 100B | Intro to Marine Data Science | Winter 2025

Key concepts

1) functions act as black boxes separate universe
2) parameters and returns, those are our bridges into and out of the black box
3) debugger- a useful way to peak inside the black box

The black box

Encapsulation



gt_threshold

x
threshold

output

Demo in R

Recap

Functions operate in their own little universe “black box”
Parameters are how we let information in
return() is how we let information back out

New vocabulary and lingering questions

New vocabulary

Lingering questions

Exercises

- What value does the following code yield?

- How could you change `fish_mass` so the code yields 12 instead?

- How could you change the body of the function so the code yields 12?

```
fish_mass <- 5
temperature <- 20
fish_growth <- function(mass, temp) {
  growth <- 2 + 0.2 * temp
  mass <- mass + growth
  return(mass)
}
fish_growth(fish_mass, temperature)
```

Exercises

In your own words, why does running this code generate an error?

```
calc_volume <- function(height, width, depth) {
  area <- height * width
  volume <- area * depth
  return(volume)
}
vol <- calc_volume(3, 5, 1)
area
```

PROG102: Functions

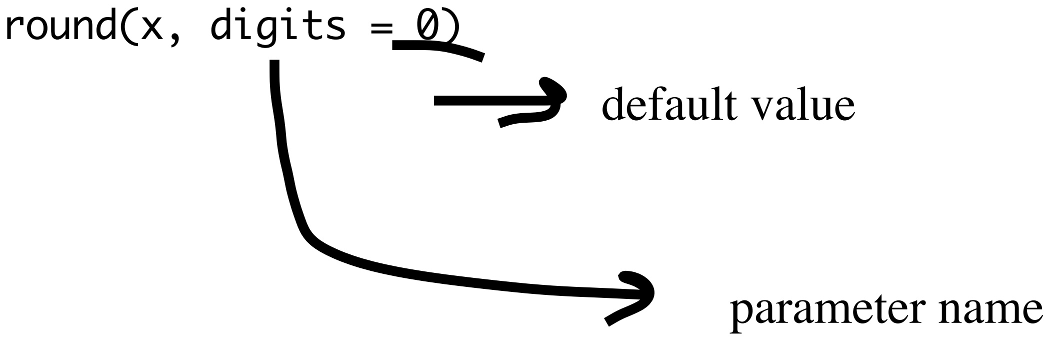
Default and named parameters

MARINCS 100B | Intro to Marine Data Science | Winter 2025

Key concepts

1) parameters usually enter in order- by position
2) default parameter values allow you to omit certain values
3) named parameters let you skip around in order
4) default and named parameters are usually options

Default and named parameters



```
round(x, digits = 0)
```

default value

parameter name

round(pi) —> 3 * use the default "by position"

round(pi,0) —>3 "by position"

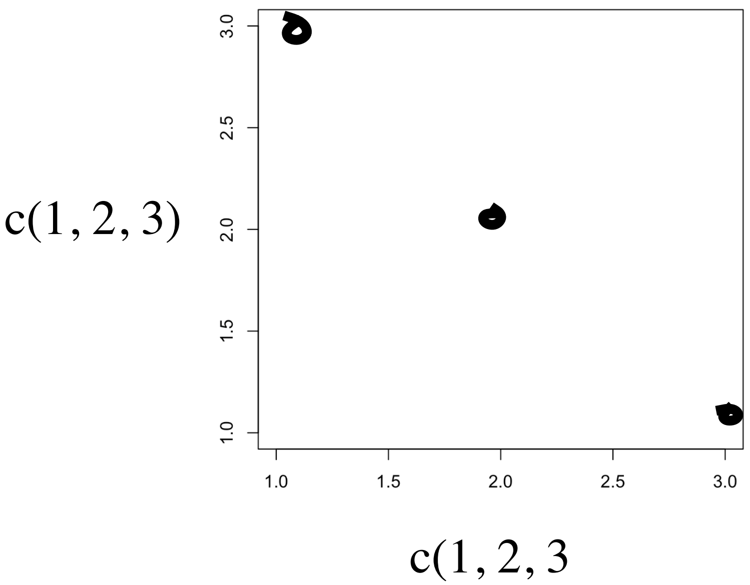
round(digits =0, pi) —>3

Long parameter lists

```
plot(x, y = NULL, type = "p",  xlim = NULL, ylim = NULL,
     log = "", main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
     ann = par("ann"), axes = TRUE, frame.plot = axes,
     panel.first = NULL, panel.last = NULL, asp = NA,
     xgap.axis = NA, ygap.axis = NA,
     ...)
```

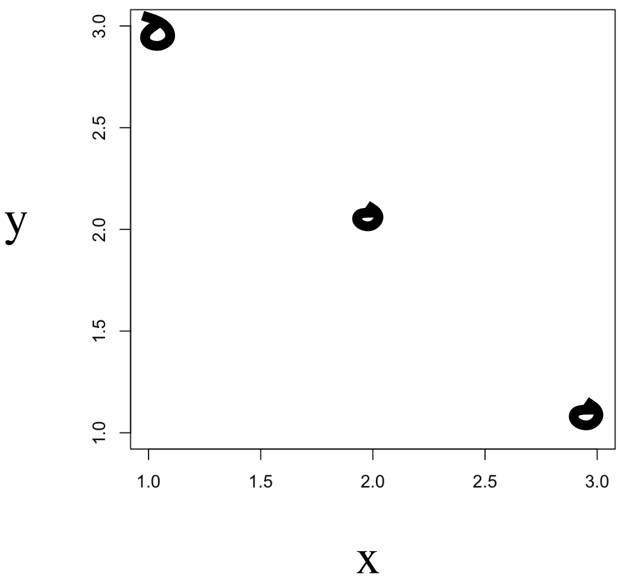
x y

```
plot(c(1, 2, 3), c(3, 2, 1))
```



x y

```
plot(c(1, 2, 3), c(3, 2, 1),
     xlab = "x", ylab = "y")
```



Demo in R

skip

Triple dots

```
max(..., na.rm = FALSE)
paste(..., sep = " ", collapse = NULL, recycle0 = FALSE)
```

ignore the triple dots and focus on the name of the function

max(1,2,3) -> 3

paste (“water”, “is”, “wet”) —> “water is wet”

Recap

1) named and default parameters are useful for modifying how functions works
2) default values allow omission
3) named parameters allow us to skip around

New vocabulary and lingering questions

New vocabulary

Lingering questions

Exercises

R represents *missing* data with the value NA. Say you're doing an experiment and you miss the second observation. In R you can write that as `c(1, NA, 3, 4)`.

Most summary functions, like `mean()`, `max()`, and `median()`, have a parameter called `na.rm`. What does this parameter do? What is its default value? How would you get the maximum value of the vector `c(1, NA, 3, 4)`?