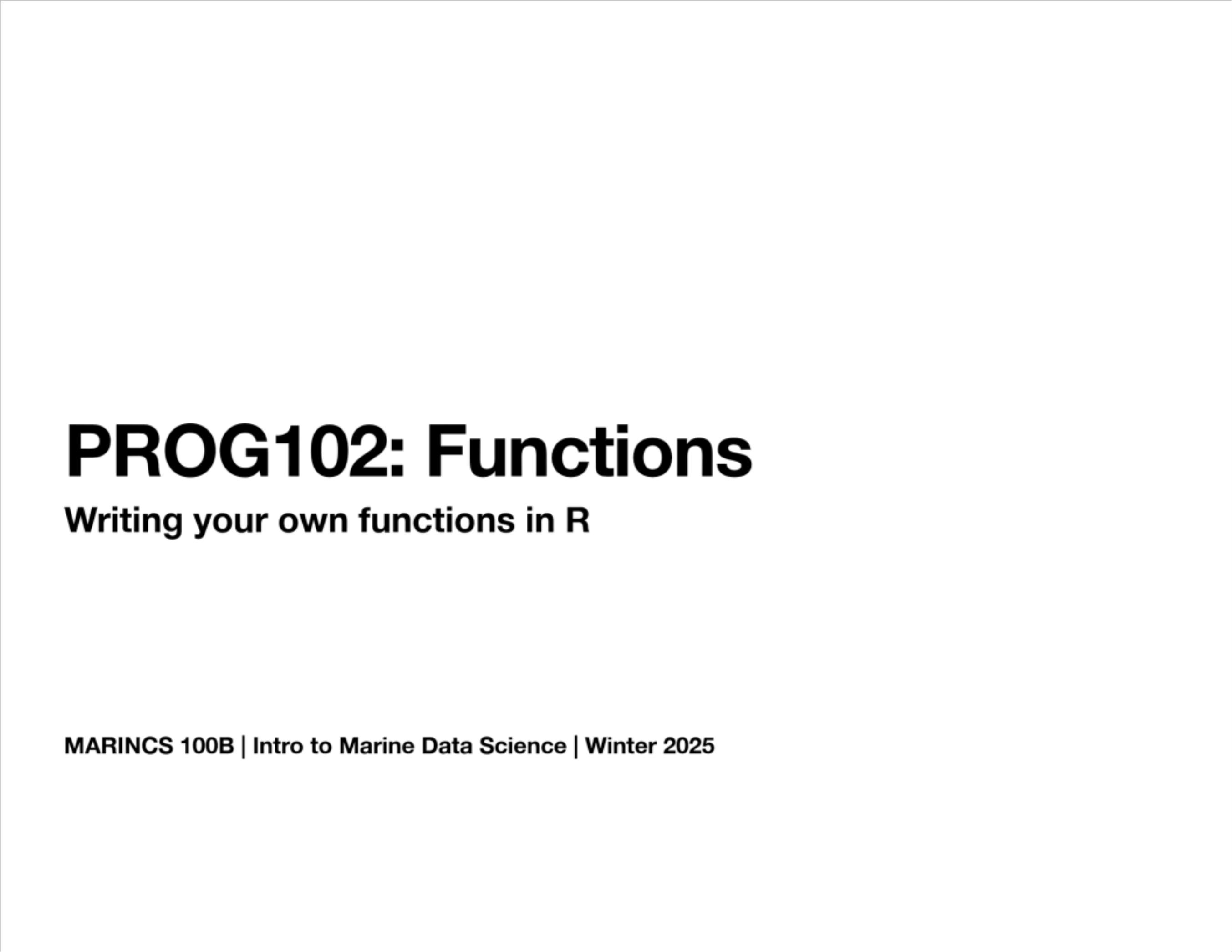
PROG102 Guided Notes

Thursday, January 16, 2025 8:26 AM

PROG102
Guided N...



PROG102: Functions

Writing your own functions in R

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

Key concepts

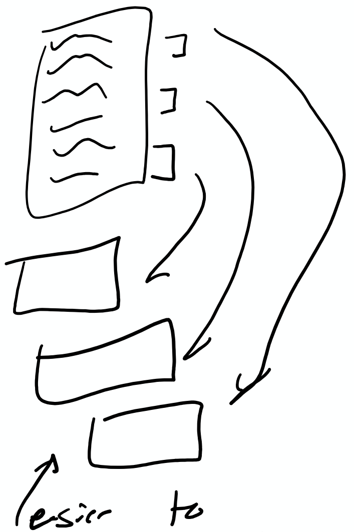
2 purposes

1) Hide details behind name
(encapsulation)

2) Apply same code over again (function)
(reusability)

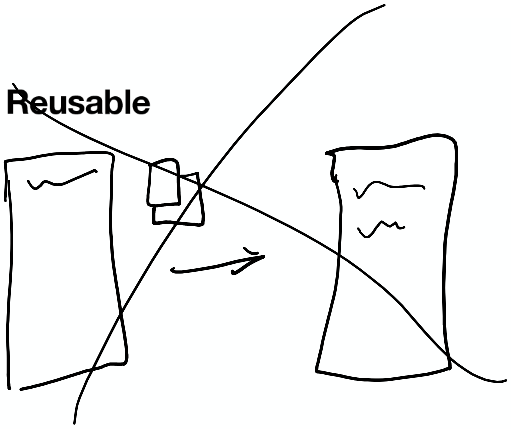
Easy to read

Human
only think
at ≤ 7
items



easier to
read/comprehend

Reusable



Use parameters

input → beep boop → outputs

(same logic, new inputs, new outputs)

Syntax

① Name ② Keyword Function (:) ③ Parameters (parenthesis

④ Body { } ⑤ Return output

Demo in R

See code

Recap

Functions make code readable by hiding details (encapsulation) yippie

Functions make code reusable by allowing diff inputs (parameters)

Syntax - every function def has 5 parts (recognize them!)

New vocabulary and lingering questions

New vocabulary	Lingering questions
Encapsulation = make something simpler by containing	N/A

Exercises

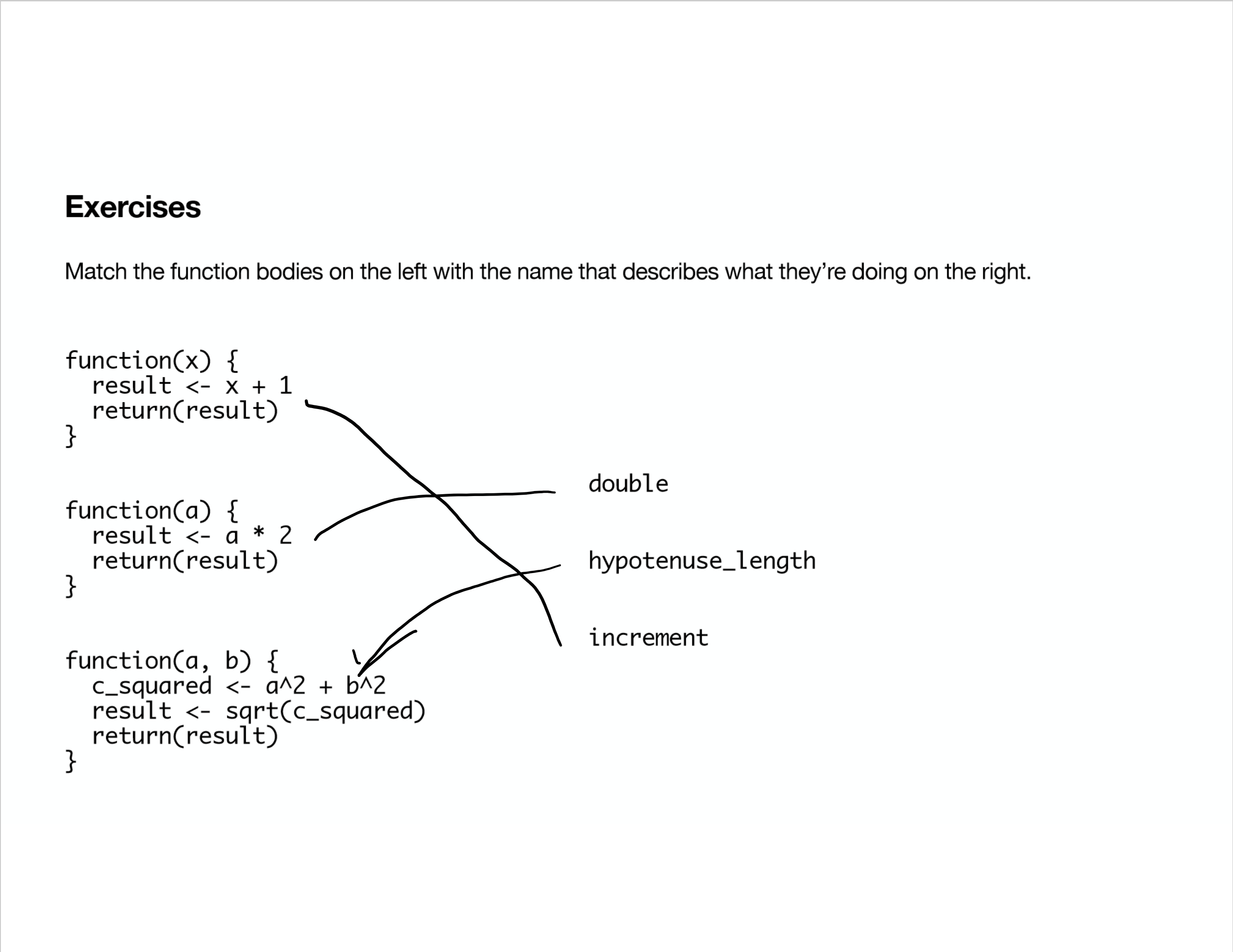
Label the five parts of this function:

Name Keyword func argument

```
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)
}
```

body

return



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

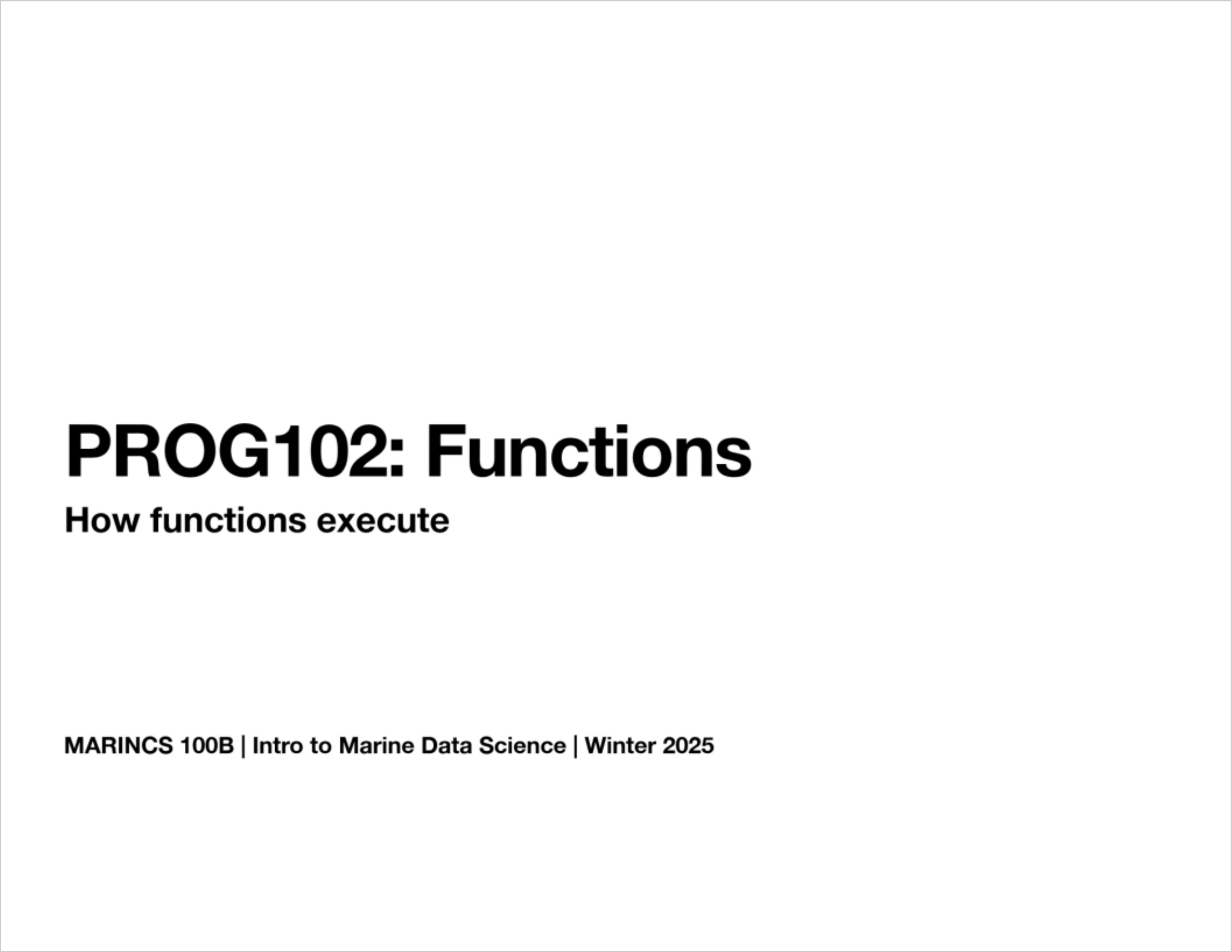
hypotenuse_length

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.

```
pali( x) {
  newx <- x + rev(x)
  return(new x)
}
```



PROG102: Functions

How functions execute

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

Key concepts

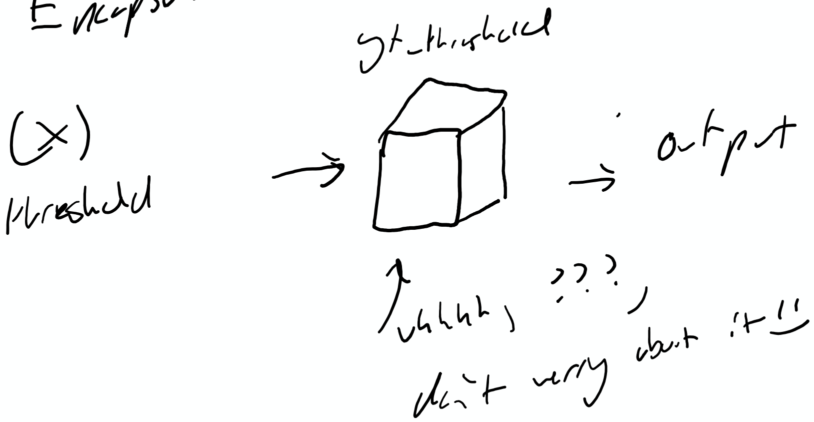
1. Functions act as black boxes (unknown)
   separate universe

2. Parameters + returns, these are our bridges
   in/out black box

3. Debugger - helps (??)

The black box

Encapsulation



Demo in R

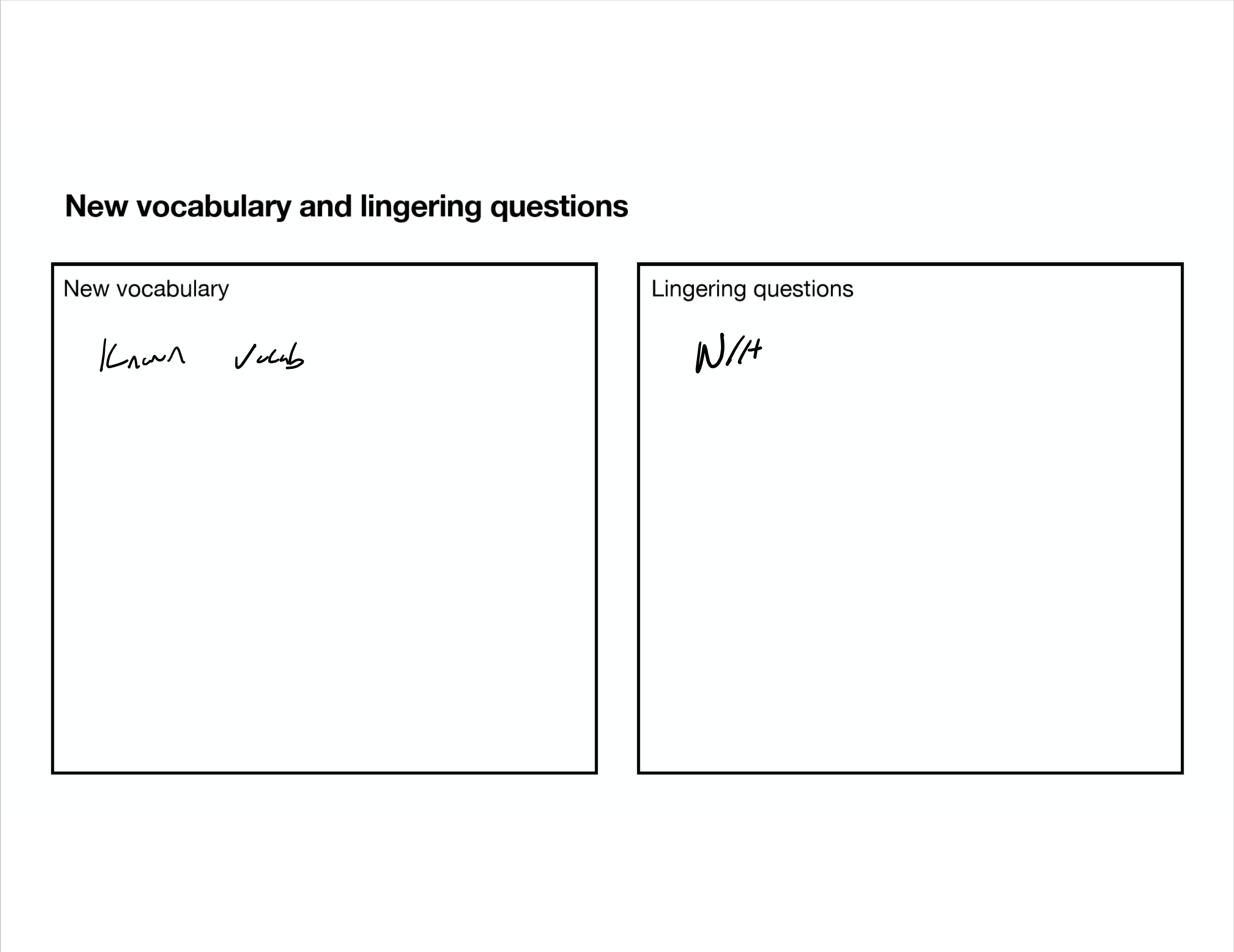
See code

✓

Recap

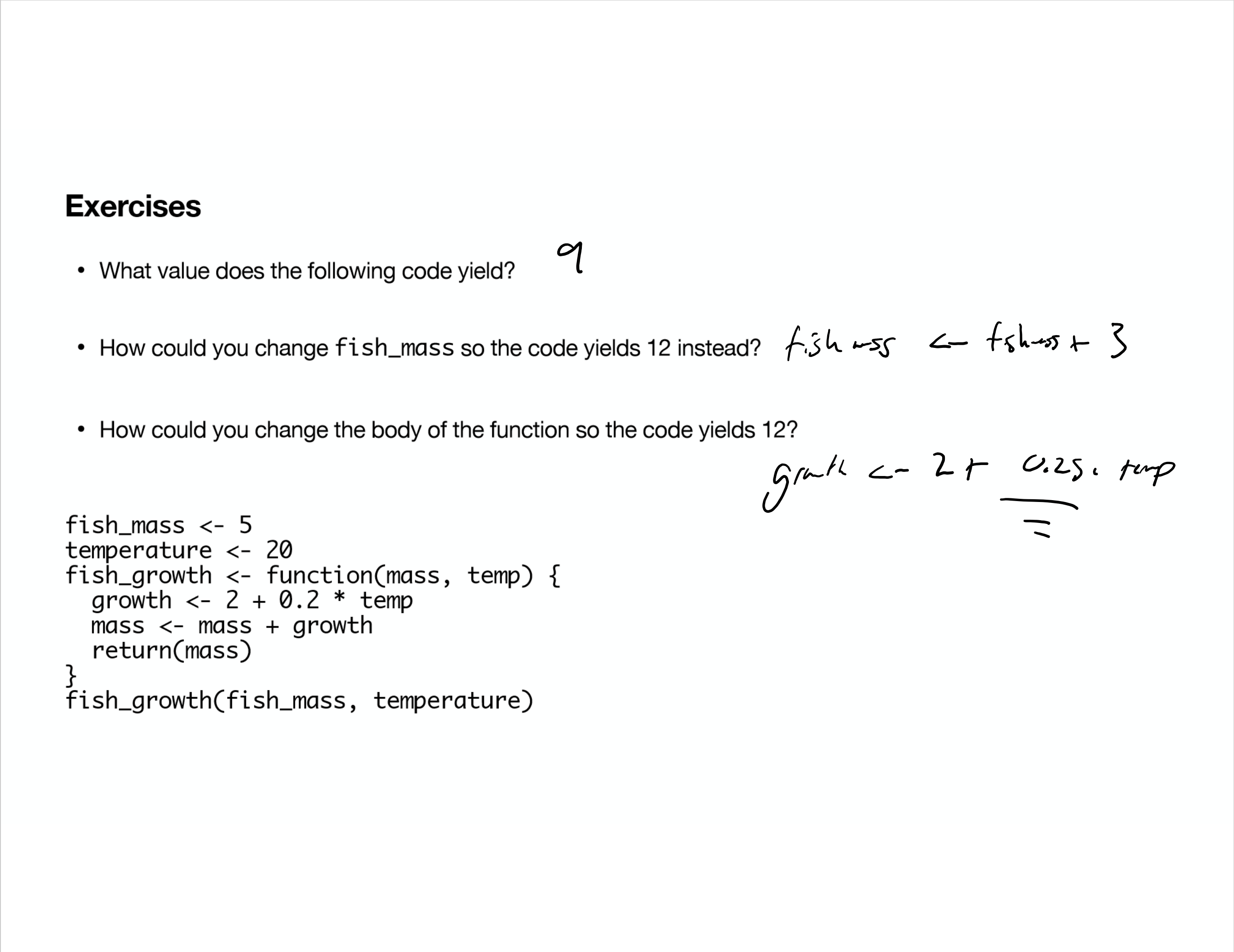
Functions own uni

Parameters go unchanged after



New vocabulary and lingering questions

New vocabulary	Lingering questions
Known vocab	N/A

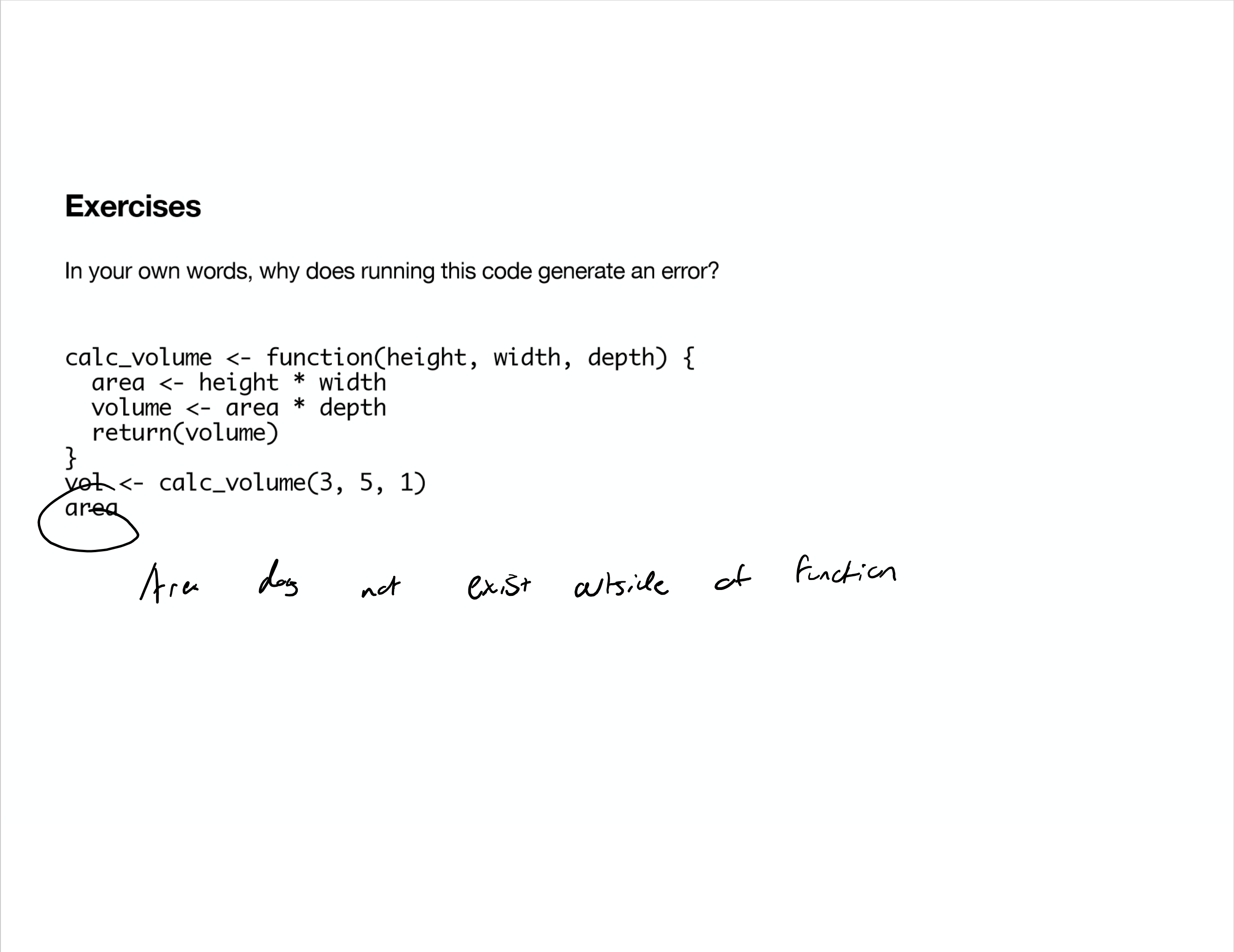


Exercises

- What value does the following code yield? 9
- How could you change fish_mass so the code yields 12 instead? fish mass <- fishmass + 3
- How could you change the body of the function so the code yields 12?

growth <- 2 + 0.25 · temp

```
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)
```

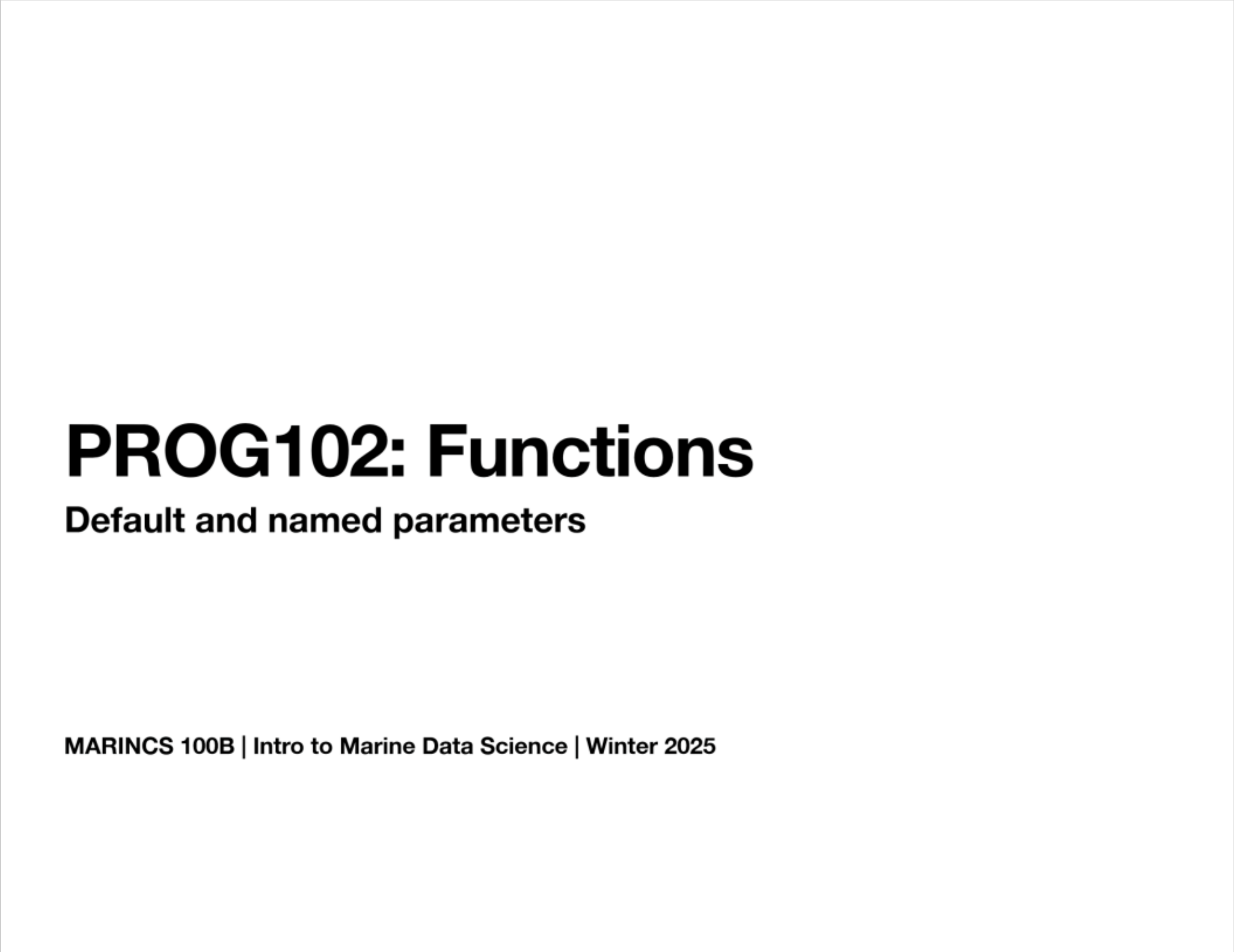


Exercises

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
```

Area does not exist outside of function



PROG102: Functions

Default and named parameters

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

Key concepts

1) Parameters usually enter in order "by pos"

2) Default parameter values allow omit values

3) Named parameters let skip in order ☺

4) Default + named parameters are usually options ☺

Default and named parameters

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

parameter name

default value

round(pi) → 3   default

round(pi, 0) → 3   by pos

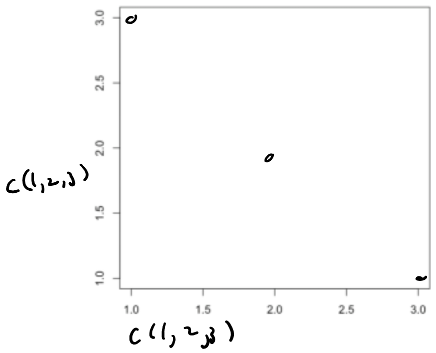
round(digits = 0, pi) → 3   all equiv, named parameter

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))
```



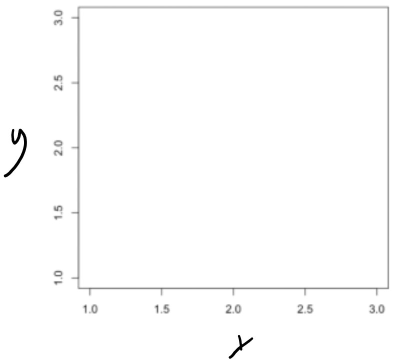
x y

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

Skip over above

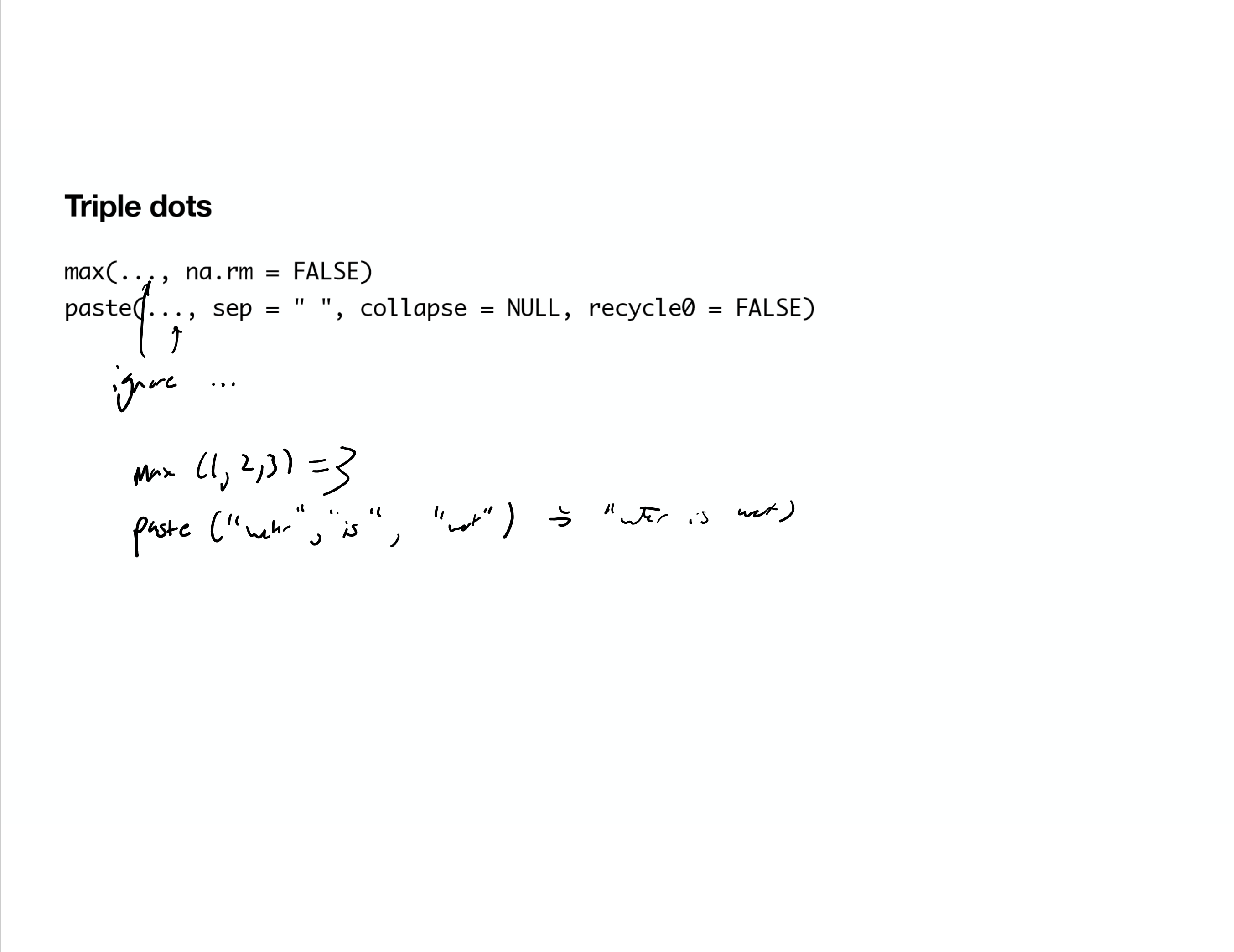
named parameters

Clearer



Demo in R

Skip



Triple dots

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

ignore ...

max(1, 2, 3) = 3

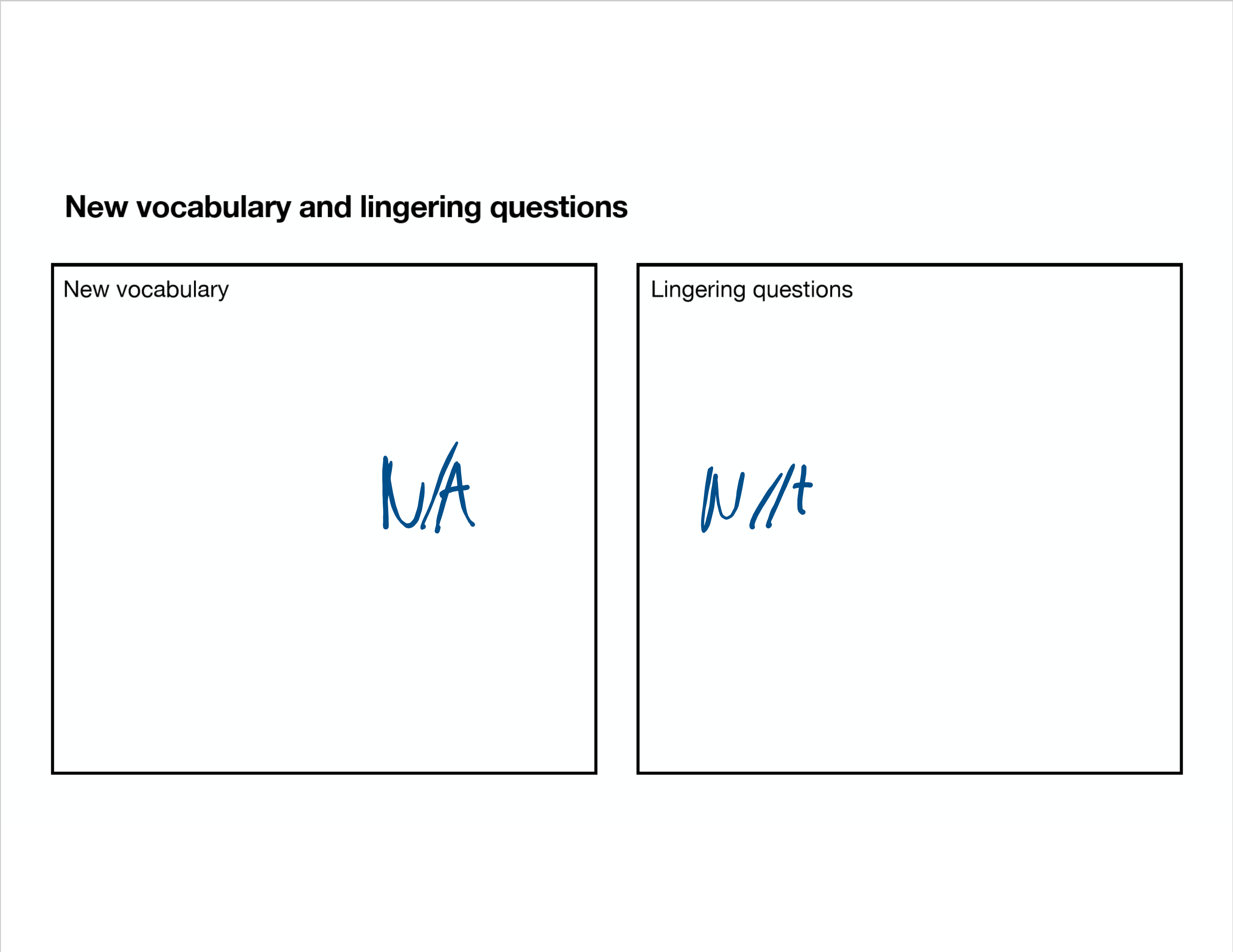
paste("water", "is", "wet") => "water is wet)

Recap

1) Named and default parameters are useful for modifying how functions work

2) Default values allow missing

3) Named allow skip around



New vocabulary and lingering questions

New vocabulary

N/A

Lingering questions

N/A

Exercises

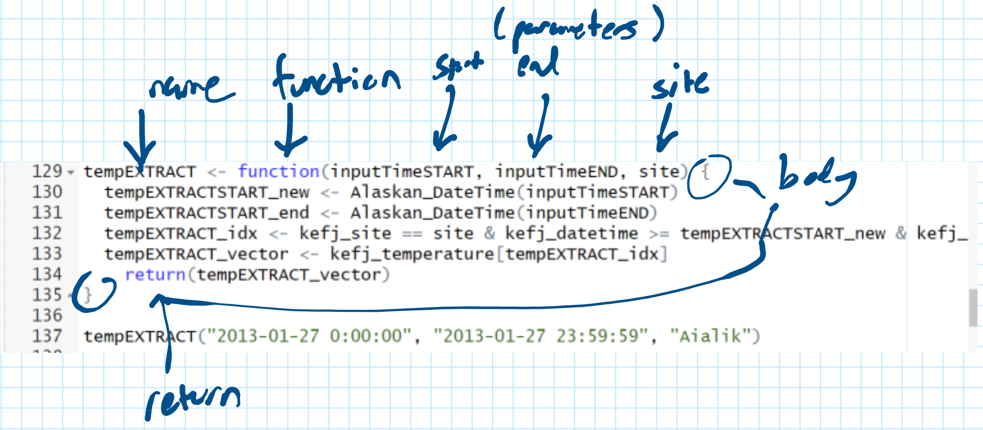
R represents *missing* data with the value NA. Say you're doing an experiment and you miss the second observation. In R you can write that as `c(1, NA, 3, 4)`.

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)`?

It removes any NA values

Defult value is execute

Remove NA, set highest as x and keep, compare to next value, return x



```
tempEXTRACT <- function(inputTimeSTART, inputTimeEND, site) {
  tempEXTRACTSTART_new <- Alaskan_DateTime(inputTimeSTART)
  tempEXTRACTSTART_end <- Alaskan_DateTime(inputTimeEND)
  tempEXTRACT_idx <- kefj_site == site & kefj_datetime >= tempEXTRACTSTART_new & kefj_
  tempEXTRACT_vector <- kefj_temperature[tempEXTRACT_idx]
    return(tempEXTRACT_vector)
}

tempEXTRACT("2013-01-27 0:00:00", "2013-01-27 23:59:59", "Aialik")
```