

## Week 2 Quiz

### Question 1

Suppose I define the following function in R

```
cube <- function(x, n) {  
  x^3  
}
```

What is the result of running

```
cube(3)
```

in R after defining this function?

1 / 1 point

- ☒ The number 27 is returned
- ☐ A warning is given with no value returned.
- ☐ The users is prompted to specify the value of 'n'.
- ☐ An error is returned because 'n' is not specified in the call to 'cube'

**Correct**

Because 'n' is not evaluated, it is not needed even though it is a formal argument.

### Question 2

The following code will produce a warning in R.

```
x <- 1:10  
if(x > 5) {  
  x <- 0  
}
```

Why?

1 / 1 point

- ☐ The syntax of this R expression is incorrect.
- ☐ There are no elements in 'x' that are greater than 5
- ☒ 'x' is a vector of length 10 and 'if' can only test a single logical statement.
- ☐ You cannot set 'x' to be 0 because 'x' is a vector and 0 is a scalar.
- ☐ The expression uses curly braces.

**Correct**

### Question 3

Consider the following function

```
f <- function(x) {  
  g <- function(y) {  
    y + z  
  }  
}
```

```
z <- 4
x + g(x)
}
```

If I then run in R

```
z <- 10
f(3)
```

What value is returned?

1 / 1 point

- ☒ 10
- ☐ 16
- ☐ 7
- ☐ 4

**Correct**

Question 4

Consider the following expression:

```
x <- 5
y <- if (x < 3) {
  NA
} else {
  10
}
```

What is the value of 'y' after evaluating this expression?

1 / 1 point

- ☐ 5
- ☐ 3
- ☐ NA
- ☒ 10

**Correct**

Question 5

Consider the following R function

```
h <- function (x, y = NULL, d = 3L) {
  z <- cbind (x, d)
  if(!is.null(y))
    z <- z + y
  else
    z <- z + f
  g <- x + y / z
  if(d == 3L)
    return(g)
  g <- g + 10
  g
}
```

Which symbol in the above function is a free variable?

1 / 1 point

- ☒ f
- ☐ z
- ☐ d
- ☐ L
- ☐ g

**Correct**

Question 6

What is an environment in R?

1 / 1 point

- ☐ a list whose elements are all functions
- ☐ an R package that only contains data
- ☐ a special type of function
- ☒ a collection of symbol/value pairs

**Correct**

Question 7

The R language uses what type of scoping rule for resolving free variables?

1 / 1 point

- ☐ compilation scoping
- ☐ dynamic scoping
- ☐ global scoping
- ☒ lexical scoping

**Correct**

8.Question 8

How are free variables in R functions resolved?

1 / 1 point

- ☐ The values of free variables are searched for in the global environment
- ☒ The values of free variables are searched for in the environment in which the function was defined
- ☐ The values of free variables are searched for in the environment in which the function was called
- ☐ The values of free variables are searched for in the working directory

**Correct**

Question 9

What is one of the consequences of the scoping rules used in R?

1 / 1 point

- ☐ Functions cannot be nested
- ☐ All objects can be stored on the disk

- ☒ All objects must be stored in memory
- ☐ R objects cannot be larger than 100 MB

**Correct**

10.Question 10

In R, what is the parent frame?

**1 / 1 point**

- ☒ It is the environment in which a function was called
- ☐ It is always the global environment
- ☐ It is the package search list
- ☐ It is the environment in which a function was defined

**Correct**