Back

Week 2 Quiz

Quiz, 10 questions

Question 1

1  
point

**1. Question 1**

Suppose I define the following function in R



1

2

3

cube <- function(x, n) {

x^3

}

What is the result of running



1

cube(3)

in R after defining this function?



A warning is given with no value returned.



The number 27 is 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'

Question 2

1  
point

**2. Question 2**

The following code will produce a warning in R.



1

2

3

4

x <- 1:10

if(x > 5) {

x <- 0

}

Why?



There are no elements in 'x' that are greater than 5



The expression uses curly braces.



The syntax of this R expression is incorrect.



'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.

Question 3

1  
point

**3. Question 3**

Consider the following function



1

2

3

4

5

6

7

f <- function(x) {

g <- function(y) {

y + z

}

z <- 4

x + g(x)

}

If I then run in R



1

2

z <- 10

f(3)

What value is returned?



7



16



10



4

Question 4

1  
point

**4. Question 4**

Consider the following expression:



1

2

3

4

5

6

x <- 5

y <- if(x < 3) {

NA

} else {

10

}

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



10



5



3



NA

Question 5

1  
point

**5. Question 5**

Consider the following R function



1

2

3

4

5

6

7

8

9

10

11

12

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?



f



z



d



L



g

Question 6

1  
point

**6. Question 6**

What is an environment in R?



a collection of symbol/value pairs



a list whose elements are all functions



a special type of function



an R package that only contains data

Question 7

1  
point

**7. Question 7**

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



global scoping



compilation scoping



lexical scoping



dynamic scoping

Question 8

1  
point

**8. Question 8**

How are free variables in R functions resolved?



The values of free variables are searched for in the global environment



The values of free variables are searched for in the working directory



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

Question 9

1  
point

**9. Question 9**

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



All objects must be stored in memory



All objects can be stored on the disk



R objects cannot be larger than 100 MB



Functions cannot be nested

Question 10

1  
point

**10. Question 10**

In R, what is the parent frame?



It is the environment in which a function was defined



It is the package search list



It is the environment in which a function was called



It is always the global environment

Upgrade to submit