Learning R - Session 1
Quiz 2

Viraj

June 14, 2016

Quiz Questions and Answers

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?

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

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

cube(3)

## [1] 27
```

The n is a formal argument, and not every function call in R makes use of all the formal arguments

Question 2

The following code will produce a warning in R.

```
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.

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

## Warning in if (x > 5) {: the condition has length > 1 and only the first
## element will be used
```

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?

- **10**
- 16
- 4
- 7

```
f <- function(x) {
  g <- function(y) {
  y + z
  }
  z <- 4
  x + g(x)
}

z <- 10
f(3)

## [1] 10
```

The variables declared inside a function are local to that function. So the g function considers the value of z as 4 and not 10

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?

- 5
- **10**
- 3
- NA

```
x <- 5

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

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?

- **f**
- z
- d
- L

- g *'free' variable -- a variable that is not defined in the function nor an argument of the function. The only such variable is "f"*

Question 6

What is an environment in R?

- 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**

Question 7

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

- global scoping
- dynamic scoping
- **lexical scoping**
- compilation scoping

It uses lexical scoping in which the sequence of environments to look in is determined when the function is defined, but the value is determined when the function is called.

Question 8

How are free variables in R functions resolved?

- 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 called
- **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 global environment

It uses lexical scoping in which the sequence of environments to look in is determined when the function is defined, but the value is determined when the function is called.

Question 9

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

- All objects can be stored on the disk
- R objects cannot be larger than 100 MB
- **All objects must be stored in memory**
- Functions cannot be nested

Question 10

In R, what is the parent frame?

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