BIOS 512- Homework 2

Alexis Bryant

2025-09-04

```
library(tidyverse);
 tinytex::install_tinytex(force = TRUE)
```

Question 1

What are three ways that we can assign the value of 25 to y?

```
y <- 25
y=25
y <<- 25

y
```

```
## [1] 25
```

Question 2

How would I print the following line, including the quotes, in R?
He said, "I'm Garth Marenghi. Author. Dreamweaver. Visionary. Plus actor."

```
cat("He said, \"I'm Garth Marenghi. Author. Dreamweaver. Visionary. Plus actor.\"")
```

```
## He said, "I'm Garth Marenghi. Author. Dreamweaver. Visionary. Plus actor."
```

Question 3

What if I wanted to add backslashes into the statement from Question 1 to make the statement below?
He said, "I'm Garth Marenghi. Author\Dreamweaver\Visionary. Plus actor."

```
cat("He said, \"I'm Garth Marenghi. Author\\Dreamweaver\\Visionary. Plus actor.\"")
```

```
## He said, "I'm Garth Marenghi. Author\Dreamweaver\Visionary. Plus actor."
```

Question 4

Show two ways to get the following array: 1 2 3

```r
1:3
```

```
## [1] 1 2 3
```

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

```
## [1] 1 2 3
```

Question 5

What does R call things like +, -, sin(), c(), etc? What about <-?

+,-,sin(),c() are functions and <- is an assignment operator or function/special operator that creates variable bindings

Question 6

What's the difference in the way R processes the while() and the for() below?

```r
x <- 0;
while (x <= 3) {
    x <- x + 1;
}
x
```

```
## [1] 4
```

```r
for(y in c(1,2,3)) {
     y <- y + 1;
}
y
```

```
## [1] 4
```

In the while loop R reevaluates the function condition each time and updates x in place until the condition fails. The for loop iterates over the values given in the sequence and inside each iteration, y is binded to the current element and then it changes the loop variable. After the for loop, y just keeps the last changed value. So, while loops depend on a loop condition and continuously updates the variable binding and for loops iterate across a sequence of values where it binds the loop variable each time.

Question 7

Create the Pythagorean formula and evaluate it with a=3 and b=4. Print the output.

```r
a <- 3
b <- 4
c <- sqrt(a^2 + b^2)
c
```

```
## [1] 5
```

Question 8

Load the help for the built in sin() function.

```
?sin
```

```
## starting httpd help server ... done
```

Question 9

Which version of the counter function works? What is the difference in the way R processes the two functions?

```
counter1 <- function(start, step){
    val <- start;
    function(){
        old_val <- val;
        val <- val + step;
        old_val;
    }
}
counter_from_1 <- counter1(1,1);
counter_from_1()
```

```
## [1] 1
```

```
counter_from_1()
```

```
## [1] 1
```

```
counter2 <- function(start, step){
    val <- start;
    function(){
        old_val <- val;
        val <<- val + step;
        old_val;
    }
}
counter_from_1 <- counter2(1,1);
counter_from_1()
```

```
## [1] 1
```

```
counter_from_1()
```

```
## [1] 2
```

Counter 2 works because it uses «- the super assignment so it modifies the closest binding in the original environment so the value gets updated Counter 1 doesn't work because it uses <- in the inside function, so it creates a new variable binding instead of updating the other val variable so it keeps returning 1

Question 10

a) Use read_csv the cars.csv and store it in a data frame.
b) Then, group the data frame by Make, get averages across the numeric variables, and then sort by Volume in descending order. Hint: Use summarise(across(c(), mean)) to get the averages.

```
library(tidyverse)

df.cars <- read_csv("cars.csv")
```

```
## Rows: 36 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Make, Model
## dbl (3): Volume, Weight, CO2
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
cars_summary <- df.cars %>%
    group_by(Make) %>%
  summarise(
      across(c(Volume, Weight, CO2), mean))%>%
  arrange(desc(Volume))

cars_summary
```

```
## # A tibble: 17 x 4
##    Make       Volume Weight   CO2
##    <chr>       <dbl>  <dbl> <dbl>
##  1 Mazda        2200   1280   104
##  2 Mercedes     1940   1439   106.
##  3 Audi         1867.  1455   106.
##  4 Volvo        1867.  1561.  108.
##  5 BMW          1733.  1487.  107
##  6 Opel         1733.  1388.  102.
##  7 Honda        1600   1252    94
##  8 Hundai       1600   1326    97
##  9 Ford         1540   1274.  100
## 10 Mini         1500   1140   105
## 11 Skoda        1400   1143    97
## 12 Suzuki       1300    990   101
## 13 Mitsubishi   1200   1160    95
## 14 Hyundai      1100    980    99
## 15 Toyoty       1000    790    99
## 16 VW           1000    929   105
## 17 Fiat          900    865    90
```

Question 11

Make a function that returns the Fibonacci sequence, then call it 7 times to return the first 7 values of the sequence. Use the correct counter function from question 9 for inspiration.

```r
fibonacci_Sequence <- function(a0 = 0, b0 = 1) {
  a <- a0; b <- b0
  function() {
    output <- a
    next_b <- a + b
    a <<- b
    b <<- next_b
    output
  }
}

fib_seq <- fibonacci_Sequence(0, 1)
vals <- vapply(1:7, function(i) fib_seq(), numeric(1))
vals
```

```
## [1] 0 1 1 2 3 5 8
```