Mid-course assessment revisions

Sawyer Balint

Fall 2024; Marine Semester Block 3

Question 3

Chunk 1:

```r
a <- 1

my_fun <- function(x){
  a <- 2
  return(x)
}

print(a)
```

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

Chunk 2:

```r
a <- 1

my_fun <- function(x){
  a <- 2
  return(x)
}

b <- my_fun(3)
print(a)
```

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

Chunk 3:

```r
a <- 1

my_fun <- function(x){
  a <- x
  return(x)
}

a <- my_fun(3)
print(a)
```

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

Explanation

This question is illustrating the difference between variables created in the *global environment* and variables created in the *local environment*. Variables created in the global environment can be accessed by any part of the code, including within a function. Variables created within a function only exist within that function's local environment, and will supersede a variable from global environment if it shares the same name. However, this change will only apply within the local environment, and outside of the function the variable will remain unchanged. This behavior is because R uses *lexical/static scoping*, meaning that when a variable is called R will first search within the local environment, and then any parent environments (e.g., nested functions), and then finally the global environment. If the variable is not defined in any environment, R returns an error.

In **Chunk 2**, `a = 1` in the global environment and `a = 2` in the local environment of `my_fun()`. Because `print(a)` occurs outside of `my_fun()`, a value of `1` is returned even though `my_fun()` was evaluated and used to define another variable `b`.