Iterations in R: Takeaways

Syntax

- Use a for loop to loop over the *values* of a vector:

```
prices <- c(10, 20, 30, 40, 50)
profits <- c()

for (price in prices) {
  calc <- 1 * price^2 + 60 * price
  profits <- c(profits, calc)
}
```

- Use a for loop to loop over the *indices* to use with a vector or collection of vectors:

```
prices <- c(10, 20, 30, 40, 50)
profits <- integer(length(prices))
indices <- 1:length(prices)

for (i in indices) {
  calc <- 1 * prices[i]^2 + 60 * prices[i]
  profits[i] <- calc
}
```

- Use a while loop to search for a value that satisfies a condition

```
# Loop set up
nums <- 1:10
is_condition_met <- FALSE
i <- 1

# Loop execution
while (!is_condition_met) {

  current_num <- nums[i] # Grab a number using the index
```

```
  is_condition_met <- current_num > 7 # Recheck the condition
  i <- i + 1 # Increase the index
}
```

Takeaways

- Iteration is the act of repeating an operation on a set of values. `for` loops repeat code for a defined vector of values, while `while` loops repeat code until a condition is satisfied.
- Iteration saves us time by preventing us from writing out redundant code over and over again.
- When possible, use vectorization over iteration. Vectorization applies an operation over all of the values of a vector at the same time, while iteration applies an operation one at a time.
- A `for` loop should be used when we know the number of items we want to iterate through. A `while` loop is better for when we don't know this number. We may think of a `for` loop as a to-do list, whereas a `while` loop can be thought of as a milestone.

Further Reading

- The Iteration Chapter in R for Data Science by Hadley Wickham