Post 1 - While Loops

Tammy Vu

Introduction

While loops have many and varied use cases. Today, I am going to talk about when they are useful, but also when they are not useful. I will also put while loops in the context of other programming languages outside of just R.

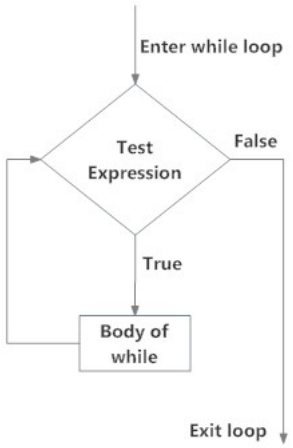
Agenda

1. What are they?
2. Use Cases
3. Some Examples
4. When are they not useful?
5. Other languages
6. Conclusion
7. References

1. What are they?

According to Programiz, “while loops are used to loop until a specific condition is met” [1]. A loop is used to repeat an action multiple times. We put an expression in the while loop that can be evaluated to true or false. The while loop checks to make sure the expression is not false before executing the action. While loops have the risk of repeating forever if we do not have any place where the expression changes to false. Therefore, we must be very careful when using a while loop to make sure that the loop will eventually stop. If we do not, there could be a timeout in R which will stop the rest of our code from running.

Flowchart:



[2]

Fig: operation of while loop

This is the syntax of a while loop:

```
while (condition is met)
{
   do this
}
```

2. Use Cases

While loops have many uses:

- Do an action `n` number of times
- Creating an infinite loop
- Until a condition is met
- While a variable is greater than zero

3. Some Examples

You can break out of the while loop using a `break` keyword. Nothing after the keyword is executed.

```
x <- 0
while (x > 0) {
  break
  print ("Made it here!")
}
```

A real use of `break`.

```
while (true)
{
    //do complicated stuff
    if (someCondition) break;
    //more stuff
}
```

How many times is a number divisible by 2?

```
n <- 20
counter <- 0
while (n%%2 == 0)
{
  counter <- counter + 1
  n <- n/2
}
print(counter)
```

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

Find the largest number greater than zero that is divisible by 5

```
x <- 29
while (x > 0)
{
  if (x %% 5 == 0) {
    break
  }
  x <- x - 1
}
print(x)
```

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

In DataCamp, they use while loops with functions like so [4]:

```
# Your User Defined Function
readinteger <- function(){
  n <- readline(prompt="Please, enter your ANSWER: ")
}

response <- as.integer(readinteger())

while (response!=42) {
  print("Sorry, the answer to whatever the question MUST be 42");
  response <- as.integer(readinteger());
}
```

Check for 2 conditions at once in an expression [5]:

```
x <- 0
while (x < 5 && x > 0)  {
  x <- x + 3
}
print(x)
```

```
## [1] 0
```

R while loops are useful for when you want to skip a step using the `next` keyword [5].

```
x <- 1
while (x < 5) {
  x <- x+1
  if (x == 3) {
    next
  }
  print(x)
}
```

```
## [1] 2
## [1] 4
## [1] 5
```

We can print things a certain number of times [6]

```r
v <- c("Hello","while loop")
cnt <- 2

while (cnt < 7) {
   print(v)
   cnt = cnt + 1
}
```

```
## [1] "Hello"      "while loop"
## [1] "Hello"      "while loop"
## [1] "Hello"      "while loop"
## [1] "Hello"      "while loop"
## [1] "Hello"      "while loop"
```

4. When are they not useful?

While loops are not that useful when we are looping over the length of a vector to perform an action on it. This is because functions in R are automatically vectorized and should be applied directly. In this way, we can save variables and time.

While loop version:

```r
arr <- c(1,2,3)
x = length(arr)
while (x > 0) {
  arr[x] <- arr[x] + 1
  x <- x - 1
}
print(arr)
```

```
## [1] 2 3 4
```

Vectorized:

```r
arr1 <- c(1,2,3)
arr1 + 1
```

```
## [1] 2 3 4
```

While loops are also not that useful to represent looping a fixed amount of times. A for loop can be used to save space and to mitigate the risk of an infinite loop in case you forget to increment/decrement.

While loop version:

```r
count <- 1
while (count < 5) {
  print(count)
  count <- count+ 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

For loop version:

```r
for(i in 1:5) {
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

5. Other languages

In other languages, there is something called the “do while” loop. According to Wikipedia, this loop checks the condition after the loop is executed [4]. This could allow for one more iteration, a crucial distinction and potentially very useful feature. In R, this can be done as the repeat function with an if statement inside of it [7].

6. Conclusion

While loops are useful for many things in R. They are a very versatile loop that can be used with any expression. This is different from the for loop which is simply for an fixed number of times and a repeat loop where you need to use an extra if statement. While loops are not useful, however, for applying functions to arrays because R automatically vectorizes functions and recycles.

7. References

7. References

[1] https://www.programiz.com/r-programming/while-loop

[2] https://cdn.programiz.com/sites/tutorial2program/files/whileLoopFlowchart.jpg

[3] https://en.wikipedia.org/wiki/While_loop

[4] https://www.datacamp.com/community/tutorials/tutorial-on-loops-in-r

[5] https://stackoverflow.com/questions/21297582/how-to-have-two-conditions-in-a-while-loop

[6] https://www.tutorialspoint.com/r/r_while_loop.htm

[7] https://stackoverflow.com/questions/4357827/do-while-loop-in-r