Loop Speed in R

Alex Stephenson

2023-02-14

Loops in R

Loops in R have a reputation for being "slow" to compute. In general it isn't the loop itself that is a problem, but what you are doing inside of the loop. A major source of slow R code is growing an object with a loop. Whenever we use `c()`, `append()`, `cbind()`, `rbind()` or `paste()` to create a bigger object, R first allocates space for the new object and then copies the old object. Such a procedure is computationally expensive.

Simulation

To demonstrate the effect of different kinds of loops, we can do a little simulation based on Problem Set 2. We take 100000 samples with replacement of the numbers 1 to 1000 and store the mean of the sample. For each way of growing the vector, we time how long it takes the computer to complete the task.

Making a copy of the vector each time

```
startTime = Sys.time()
sampled.means = c()
set.seed(12345)
for (i in 1:100000) {
  sampled.means = c(sampled.means, mean(sample(x = 1:1000, replace = TRUE)))
}
endTime = Sys.time()
print(endTime - startTime)
```

```
## Time difference of 1.123376 mins
```

Updating via append

```
startTime = Sys.time()
sampled.means = numeric()
for(i in 1:100000){
  sampled.means = append(sampled.means, mean(sample(x = 1:1000, replace = TRUE)))
}
endTime = Sys.time()
print(endTime - startTime)
```

```
## Time difference of 1.17298 mins
```

Updating without preallocating

```
startTime = Sys.time()
sampled.means = c()
set.seed(12345)
```

```r
for(i in 1:100000){
  sampled.means[i] = mean(sample(x = 1:1000, replace = TRUE))
}
endTime = Sys.time()
print(endTime - startTime)
```

```
## Time difference of 10.56134 secs
```

Updating by preallocating

```r
startTime = Sys.time()
sampled.means = vector(mode = "logical", length = 100000)
set.seed(12345)
for(i in 1:100000){
  sampled.means[i] = mean(sample(x = 1:1000, replace = TRUE))
}
endTime = Sys.time()
print(endTime - startTime)
```

```
## Time difference of 9.591852 secs
```

Results

As we can see, using `c()` and `append()` to grow our loop object are both substantially slower than other methods. The takeaway is to avoid using them whenever possible.