If you’re like me, when running a process through a loop you’ll add in counters and progress indicators. That way you’ll know if it will take 5 minutes or much longer. It’s also good for debugging to know when the code wigged-out.

This is typically what’s done. You take a time stamp at the start – start <- Sys.time(), print out some indicators at each iteration – cat(“iteration”, k, “// reading file”, file, “\n”) and print out how long it took at the end – print(Sys.time()-start). The problem is it will print out a new line at each time it is called, which is fine but ugly. You can reduce the number of lines printed by only printing out every 10th or 100th iteration e.g. if(k %% 10 == 0) ….

A simple way to make this better is instead of using "\n" for a new line use "\r" for carriage return. This will overwrite the same line which is much neater. It’s much more satisfying watching a number go up, or down, whichever way is the good direction. Try it out…

y <- matrix(0, nrow = 31, ncol = 5)

for(sim in 1:5){

y[1, sim] <- rnorm(1, 0, 8)

for(j in 1:30){

y[j+1, sim] <- y[j, sim] + rnorm(1) # random walk

cat("simulation", sim, "// time step", sprintf("%2.0f", j), "// random walk", sprintf(y[j+1, sim], fmt='% 6.2f'), "\r")

Sys.sleep(0.1)

}

}

## simulation 5 // time step 30 // random walk 8.97

The best way is to use the {[progress](https://cran.r-project.org/web/packages/progress/index.html)} package. This package allows you to simply add running time, eta, progress bars, percentage complete as well as custom counters to your code. First decide on what counters you want and the format of the string. The function identifies counters by using a colon at the beginning of the label. Check the doco for built-in tokens.

To add your own token add the label to the format string and add the token to tick(). To make it pretty I recommend formatting digits with sprintf(). Here’s an example.

library(progress)

pb <- progress\_bar$new(format = ":elapsedfull // eta :eta // simulation :sim // time step :ts // random walk :y [:bar]", total = 30\*5, clear = FALSE)

y <- matrix(0, nrow = 31, ncol = 5)

for(sim in 1:5){

y[1, sim] <- rnorm(1, 0, 8)

for(j in 1:30){

y[j+1, sim] <- y[j, sim] + rnorm(1) # random walk

pb$tick(tokens = list(sim = sim, ts = sprintf("%2.0f", j), y = sprintf(y[j+1, sim], fmt='% 6.2f')))

Sys.sleep(0.1)

}

}

00:00:17 // eta 0s // simulation 5 // time step 30 // random walk -12.91 [====================================================]

You can also jazz it up with a bit of colour with {crayon}. Be careful with this, it doesn’t handle varying string lengths very well and can start a new line exploding your console.

library(crayon)

pb <- progress\_bar$new(format = green$bold(":elapsedfull // eta :eta // simulation :sim // time step :ts // random walk :y [:bar]"), total = 30\*5, clear = FALSE)

...

00:00:17 // eta 0s // simulation 5 // time step 30 // random walk -12.91 [====================================================]

That’s a much neater progress bar.

**But, I didn’t stop there…**

Procrastination set in and creative tangents were followed. So, made a progress bar into a big fish which eats smaller fish … and made it green.

n <- 300

bar\_fmt <- green$bold(":elapsedfull | :icon |")

pb <- progress\_bar$new(format = bar\_fmt, total = n, clear = FALSE)

icon <- progress\_bar\_icon("fish", n, 75)

for(j in 1:n){

pb$tick(tokens = list(

icon = token(icon, j)

))

Sys.sleep(0.03)

}



Each fish represents 25% completion. Once they’re all gobbled up, the job is done.

I also threw knives at boxes. Each box represents 20% completion.

n <- 300

bar\_fmt <- green$bold(":elapsedfull | :icon |")

pb <- progress\_bar$new(format = bar\_fmt, total = n, clear = FALSE)

icon <- progress\_bar\_icon("dagger", n, 75)

for(j in 1:n){

pb$tick(tokens = list(

icon = token(icon, j)

))

Sys.sleep(0.03)

}



And my personal favourite, the Star Wars trench run.

n <- 500

bar\_fmt <- green$bold(":elapsedfull | :icon |")

pb <- progress\_bar$new(format = bar\_fmt, total = n, clear = FALSE)

icon <- progress\_bar\_icon("tiefighter", n, 75)

for(j in 1:n){

pb$tick(tokens = list(

icon = token(icon, j)

))

Sys.sleep(0.03)

}

