

Keywords: notes, R, functional programming

Headline

This might be interesting in combination with `lambda.tools::tryn`:

```
call_with_delay <- function(fun, delay) {  
  force(delay)  
  function(...) {  
    Sys.sleep(delay)  
    fun(...)  
  }  
}
```

What if I have a function which needs to be applied to each element and later I want to vectorize it.

```
lapply_wrapper <- function(fun, red_fun = function(x) x) {  
  force(fun)  
  force(red_fun)  
  function(...) {  
    mc <- match.call(fun)  
    mc[[1]] <- quote(lapply)  
    names(mc)[2] <- "X"  
    funPos <- length(mc) + 1  
    mc[[funPos]] <- quote(fun)  
    names(mc)[funPos] <- "FUN"  
    red_fun(eval(mc))  
  }  
}
```

And a test:

```
vec_mean <- lapply_wrapper(mean, unlist)  
vec_mean(list(1:5, 2:10))
```

```
## [1] 3 6
```