Especially since I had to start working as a freelancer, it became essential to keep an accurate record of how much time I spend on each task and project. I checked out some software tools available online (namely TimeCamp, which I've used for a while), but I decided to see if I could write my own R functions for tracking time use, especially when **working offline** (some of us still like or need to work offline sometimes!), and for producing some related plots and analyses afterwards. Here's what I came up with.

I first need to have a file named “*PROJECTS.csv*“, with a column named “*project*” and a column named “*task*“, and the names of the projects and tasks I'll be working on, like this:

type	**project**	**task**
consulting	elephants	analysis
consulting	elephants	writing
consulting	elephants	revising
training	R-GIS	preparing
training	R-GIS	teaching
experimenting	timeman	programming
experimenting	timeman	debugging

I can manually add projects and tasks to this table at any time. Then I use the ‘*start_task*‘ and ‘*stop_task*‘ functions below to add entries to another file, which is named “*DONT-TOUCH_timeman_data.csv*“, has five columns (*proj*, *task*, *start*, *stop*, *time*) and should **not** be moved or modified manually:

```
projects <- read.csv("PROJECTS.csv")
tail(projects)

timeman_data <- read.csv("DONT-TOUCH_timeman_data.csv", as.is = TRUE)
tail(timeman_data)

start_task <- function(proj, task, time = NULL) {
  if (nrow(timeman_data) > 0 && is.na(timeman_data[nrow(timeman_data),
ncol(timeman_data)]))
    stop ("previous task must be stopped before starting another")
  if (!(proj %in% unique(projects$project)))
    stop ("proj must exist in 'PROJECTS.csv'")
  if (!(task %in% unique(projects$task)))
    stop ("task must exist in 'PROJECTS.csv'")
  #if (!(proj %in% unique(timeman_data$projecto)))
    #timeman_data[nrow(timeman_data) + 1, "projecto"] <- proj
  if (is.null(time))
    time <- Sys.time()
  timeman_data[nrow(timeman_data) + 1, 1:3] <- c(proj, task, time)
  timeman_data <<- timeman_data
}

stop_task <- function(time = NULL) {
  if (nrow(timeman_data) > 0 && !is.na(timeman_data[nrow(timeman_data),
ncol(timeman_data)]))
    stop ("task must be started before being stopped")
  if (is.null(time))
    time <- Sys.time()
  #this_row <- which.max(timeman_data$projecto == proj)
  this_row <- nrow(timeman_data)
  timeman_data[this_row, "stop"] <- as.character(time)
  timeman_data[this_row, "time"] <- difftime(timeman_data[this_row, "stop"],
timeman_data[this_row, "start"])
```

```
  timeman_data <<- timeman_data
  write.csv(timeman_data, "DONT-TOUCH_timeman_data.csv", row.names = FALSE)
  cat("Showing last rows in tasks table:\n\n")
  tail(timeman_data)
}

current_task <- function() {
  timeman_data[nrow(timeman_data), ]
}
```

I need to use ‘*start_task*‘ every time I start working on a task, and then ‘*stop_task*‘ every time I stop. This adds an entry with the starting, stopping, and total time spent in each instance:

```
unique(projects$proj)
unique(projects$task)
unique(timeman_data$proj)

# ADD START:
start_task(proj = "timeman", task = "debugging")
tail(timeman_data)

# ADD STOP:
stop_task()
tail(timeman_data)
```

I can then apply all sorts of R analyses and plots to the time-tracking data, which is recorded in the *timeman_data* data frame and CSV file. However, these functions are certainly **far from ideal**, as they use “unsafe” operators (like ‘`<<-`‘) and they rely on a file remaining untouched in its folder. It’s plain to see how things could easily go wrong at any moment…. Suggestions welcome on how this could be improved and made safer! …