For a long time, I was wondering how to get R to automatically load packages I use *every* time I open the program. For example, for a variety of reasons, I use the ‘Cairo’ package to save all my figures as PDFs. On macs, the Cairo output does funky things, like making all fonts in the figures bold and italics, so I also have to run a command resetting the fonts in Cairo to look normal. This is fine to do once, but when you do it every time you open R, it gets to be a pain (especially when you open and close R multiple times a day). Other people I know have custom functions saved as R scripts that want to load automatically. So the question was: How do you get R to do this?

The short answer: Set up an R profile. Good luck figuring this out, it took me a while. However, eventually I did manage to do it and it’s much easier than any of the help files make it sound. I only know how to do this for a mac, by the way, so if you run Windows (gross) or Linux (sweet), this article might help you in principle but not specifics.

The first thing you need to know is that you should make a text document. Call it whatever you want, I call mine ‘Rprofile.txt’. Put it in your home directory (the directory that houses the ‘Applications’, ‘Documents’, etc. folders). You can tell R to do things on startup and exit, which can be handy for automatically saving history to designated file. To do this, you need two functions called .First and .Last, which do exactly what they sound like: .First is a function of commands you want run on startup, .Last is a function of commands to run on exit. My profile looks like this:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | .First <- function(){  library(Cairo)  CairoFonts("Arial:style=Regular","Arial:style=Bold","Arial:style=Italic","Helvetica","Symbol")  cat("\nWelcome at", date(), "\nLoaded Cairo Package With Correct Fonts\n")    }    .Last <- function(){   cat("\nGoodbye at ", date(), "\n")  } |

Let’s go line by line:

1) .First <- function() { – this line says “make a function called .First”. Once you’re done, you fill the function with commands you want R to run on startup. Any command within the {} will then be run on startup.

2) library(Cairo) – load the library Cairo.

3) CairoFonts(…) – Fix the Cairo fonts

4) cat(…) – Tell me hello and remind me what I’ve loaded.

5) } – end the function

If you want R to load scripts, include a source(‘path\_to\_script’) line inside the .First function.

The same goes for .Last: make a function called .Last, insert whatever commands you want R to run on exit, and end the function.

So now you have an Rprofile text file of all the things you want R to do. Now, you have to get R to find it. R automatically searches your Home directory for a file called ‘.Rprofile’ (it’s hidden, so you won’t see it unless you enable viewing hidden files). We use the terminal to convert Rprofile.txt to .Rprofile.

Open up the terminal application and type ‘cp Rprofile.txt .Rprofile’. It’s simple: cp is ‘copy’, so your command is ‘copy Rprofile.txt to .Rprofile’.

Done! It’s as simple as that. Note that you can actually keep the Rprofile.txt file anywhere. As long as you are within your home directory in terminal, all you need to change is ‘cp path\_to\_file/Rprofile.txt .Rprofile’, where path\_to\_file is the file path to the txt file (e.g. if I stored it in my Documents folder I’d type cp Documents/Rprofile.txt .Rprofile).

I hope this saves you some time. I also won’t be held responsible if you get access to your friend’s mac and make R leave them strange messages whenever it starts up.