

Writing R – packages

1. On R Studio

- 1.1 Create a new Rstudio project called `megaplotproject` and inside create a file called `code.R` and inside the file copy the code of the function below `megaplot`. This is the only function for our package.
- 1.2 Create a package skeleton in R, using the command
`package.skeleton(name="megaplot",list=c('megaplot'))`
This will create a folder called `megaplot` and inside there is a `DESCRIPTION` file and two folders called `R` with the R code and `man` with two help functions of type `.Rd`. Edit the `DESCRIPTION` file and `.Rd` help files appropriately. If you have any doubts please follow the instructions of the manual (`Help => Manuals=> Writing R Extensions`). The `DESCRIPTION` file is easy to fill up. Just need a few words about the package, your name and email and for license please use `LGPL`. For the help file you can delete many items that do not apply.
- 1.3 Inside R-studio go to `Menu => Build => Configure build Tools` and set the package directory to the folder `megaplot` and save.
- 1.4 Then build the package by going to `Menu => Build => Build Source Package`. If there are any errors usually in the `.Rd` files they will show up in the `Build` window usually in the top right corner of the screen. Then correct errors and repeat the build process until done.

2. On Windows without R Studio

These should work for not very old versions of windows like Windows 7 32 bit and 64 bit, Windows Vista, Windows XP with regular updates. It may not work for most Windows 2000 versions and older but I would try it anyway.

- 1.5 Create a package skeleton in R, using the command
`package.skeleton(name="mypackage",list=c('superplot','superplot2','hist.inv'))`
- 1.6 Then edit the `DESCRIPTION` file and `.Rd` help files appropriately as explained in 1.2 above.
- 1.7 Download and install the following software:
 1. Mike TeX: <http://www.miktex.org>
 2. Rtools: <http://www.murdoch-sutherland.com/Rtools/>.
Rtools will install: Vanilla Perl, Tool set, MinGW, Cygwin, TCL/TK
- 1.8 Check PATH VARIABLE: (Start->Settings->)Control Panel->System->Advanced. Click on the Environment Variables button, which should be in the middle. The previous software in Rtools should be in the path. If your version of windows is not very old the previous installation on step 2. should make the appropriate modifications to the path. In my windows 7 - 64 bits my path like this at the beginning of the path:
`c:\Rtools\bin;c:\Rtools\perl\bin; C:\R\bin;c:\Rtools\MinGW64\bin;C:\Program Files\R\R-2.15.1\bin\x64;C:\Program Files (x86)\MiKTeX 2.9\miktex\bin;`
and here continues with other addresses for other non R software. For a 32 bit windows is not identical either, and there will be modifications related to your installation folder addresses. You can check in the CMD window if R is already in the path by just typing `R`. You can also check if MikeTeX is in the path by typing `PATH` in the CMD window. Download the newest version of the R batch files
<http://cran.r-project.org/contrib/extra/batchfiles/>
Extract the files to the folder `C:\Rbatch`. And pasted `C:\Rbatch;` to the `PATH` as before.

1.9 OPEN cmd window with administrator permission. Then go to the folder root to where the package was created

Cd

1.10 RUN the command to create the R-package.

R CMD build mypackage

1.11 RUN the command to install the R-package.

R CMD INSTALL mypackage

1.12 Alternative to 1.11. RUN the command to make a “zip” file that can be loaded from the menu inside Rgui.

R CMD INSTALL --build mypackage.tar.gz

1.13 If there are any error messages that you can't resolve please bring them to class and we will deal with them.

```
x = pima[,2]; y= pima[,3]; superplot(x,y)

##MEGA PLOT
megaplot = function(x,y) {
xhist <- hist(x, plot=FALSE)
yhist <- hist(y, plot=FALSE)
top <- max(c(xhist$counts, yhist$counts))
xrange <- range(x)
yrange <- range(y)
nf <- layout(matrix(c(2,1,0,3),2,2,byrow=TRUE), c(1,3), c(3,1), TRUE)
layout.show(nf)
par(mar=c(1,1,1,1))
plot(x, y, xlim=xrange, ylim=yrange, xlab="", ylab="")
par(mar=c(1,1,1,1))
barplot(yhist$counts, xlim=c(top, 0), space=0, horiz=TRUE) # !
par(mar=c(1,1,1,1))
barplot(xhist$counts, ylim=c(top,0), space=0) #!ylim!
invisible()
}
```