Table of Contents

About R program

For NumEcol 2019 class (semester 107-2): make sure that you have the latest version of R and RStudion installed on your computers:

- R - version 3.5.2 or newer
- RStudio - version 1.1.463 or newer
- install manually the following packages if not yet installed: `install.packages (c('vegan', 'labdsv', 'ade4', 'adespatial', 'cluster'))`

R is a programming language and programming environment for statistical analysis and data visualization. The R language is inspired by S language, which was invented and developed by John Chambers from Bell Laboratories and which is the base for commercial software S-PLUS. In contrast to S-PLUS, R is a freeware application, introduced by Ross Ihaka and Robert Gentleman (the R in their first names gave the name to the program).

How to install R

1. Download the installation file of the newest R version from here: CRAN[1] (choose your operating system and download the installation file).
2. Install the program - you may keep the default location of installation files in Program Files/R. On Windows, the R folder in your Program Files folder may contain several subfolders with different R versions installed in parallel, e.g. R-2.10.1, R-2.11.1 etc.
3. When you run the program, you may change some basic setting (e.g. the size of the font, a colour of the background (if you feel it is more comfortable to have it e.g. orange) in menu Edit > GUI preferences. If you change anything in this window, you need to click on Save button and save the file Rconsole to the folder /Program Files/R/R-version/etc.

How to update R

If you have R already installed on your computer and you want to update to a newer version, the easiest way is simply to install the new version along with the old one (and later possibly delete the old one if you don't need it or you want to save space on hard disk). All R versions are, if not changed, installed beside each other in *R* subfolder of *Program Files* folder (in case of Windows). You may need to **install also all the libraries** you used in the previous version. You can simply copy the folder `library` from the old R version to new R version and run an update procedure in new R version using the command `update.packages` (however, I do not recommend to do this if you are updating to new major version, e.g. from version 3.4.x to 3.5.0; in that case, the manual installation of packages (using `install.packages` function) is preferred). For this, do the following:

1. **Close all running R programs (or RStudio) before starting!**
2. Open some file manager (like Total Commander or Windows Explorer) and locate the `library`

folder with R libraries from the previous version (usually located in *c:\Program Files\R\R-x.x.x\library*, where *R-x.x.x* is the number of the previous R version, e.g. R-3.1.2).

3. Copy all subfolders in `library` folder into `library` folder of the new version. **Do not overwrite the newer versions of files in the `library` folder of the new R version**.
4. Open the new R version, and update all libraries using command `update.packages(ask = F)`[2]. This should update all the libraries to their latest version compatible with current version of R.

How to install RStudio

Download the latest version from RStudio website and follow instructions for installation. If you already have RStudio installed in your computer, you may check whether you have the latest version and update if you don't (in RStudio menu, go to *Help > Check for Updates*).

RStudio is a convenient software, which combines R program with a text editor and graphical interface (and offers much more, like the organization of scripts and outputs into projects within a single folder, and for advanced users also convenient building of packages, document markup etc.). One thing I found a bit not handy is RStudio's native graphical output - it offers its' own unique sizeable graphical output into one of the subpanels, but this sometimes produces troubles (especially in the case when you draw more complex figures). The workaround is to use RStudio, but open the original drawing engine - just type `windows ()`, and a new active window will open.

Optionally, there is a number of other text editors, which can be associated with R, e.g. Tinn-R.

How to install R packages

R has a great option to extend the basic functionality for other functions wrapped into packages. There are thousands of packages in central repositories like CRAN or Bioconductor, and also in developers repositories like R-Forge or GitHub. For an overview of documentation for packages aggregated from these repositories check the website R Documentation.

If you have administrator rights for the computer, the packages will be installed into the `library` subfolder, usually located in *c:\\Program Files\R\R-x.x.x\* [3]. If you don't have administrator rights, you will be asked to install the libraries into some writeable location of the hard disc, by default into the `Documents` subfolder of your account (here, the following structure of subfolders will be created: *Documents\R\win-library\3.2*).

Installing from CRAN

Most of the commonly used R packages for analysis of ecological data are available from CRAN, and to install them, use simply `install.packages` command in R (note that the name of the package should be separated by quotation marks, and if more than one library should be installed, wrap the names into character vector):

```
install.packages ("ade4")
```

If you are using RStudio, you may use built-in function to install the packages, which is (in default setting of RStudio) located in bottom right panel, in the folder *Packages* - click the button *Install*, in *Install from* choose *Repository (CRAN)*, and in *Packages* type the name of the package.

There are, however, examples of packages which are not available from CRAN for some reason - e.g. because they are still under development, or because they do not fit rather strict CRAN rules for hosted packages. Alternative places are e.g. R-Forge or GitHub.

Installing from GitHub

GitHub doesn't offer precompiled package versions - you need to compile the packages from source on your computer. However, there is a convenient R function in R package `devtools` developed by Hadley Wickham, which can help you with installing packages hosted on GitHub: `install_github`. To install some package (here called `MyPackage`, first install `devtools` from CRAN, and then apply the function `install_github`:

```
install.packages ("devtools")
devtools::install_github ("MyPackage")
```

(note that `::` means that `install_github` function will be read from the `devtools` namespace without need to actually upload the whole `devtools` library). If the GitHub package contains C/C++/Fortran code which needs to be compiled, you still need to install the set of compilation tools called Rtools on your computer (see Install Rtools (for Windows users only) below).

Installing from R-Forge

R-Forge contains both source packages (files with extension `*.tar.gz` containing the original R code, which needs to be compiled during installation) and also already compiled binaries for Windows (`*zip` files, do not need to be built). The binaries are available only for the latest R version available, so if you are using the older version of R, you need to build the library from source. For example, to install package `packfor`[4] for forward selection of variables directly from R-Forge, type

```
install.packages("packfor", repos="http://R-Forge.R-project.org")
```

If you are using the latest R version, the package will probably install without problems. If not, you may receive the following warning message:

```
Warning in install.packages :
  package ‘packfor’ is not available (for R version 3.1.2)
```

In that case, you need to install the package from source. Manually download the binary file from R-Forge (with `*.zip` extension) to your computer, and use the command `install.packages` with argument `repos = NULL`:

```
install.packages ("C:/Users/Downloads/packfor_0.0-8.zip", repos = NULL)
```

(note that this works if your downloaded file is the version of packfor 0.0-8 and it is saved in Downloads folder - modify this for your current needs). If even this option doesn't work, you may need to install the package from the source - download `*.tar.gz` version to your computer and use `install.packages` function with `repos = NULL` and `type = “source”`:

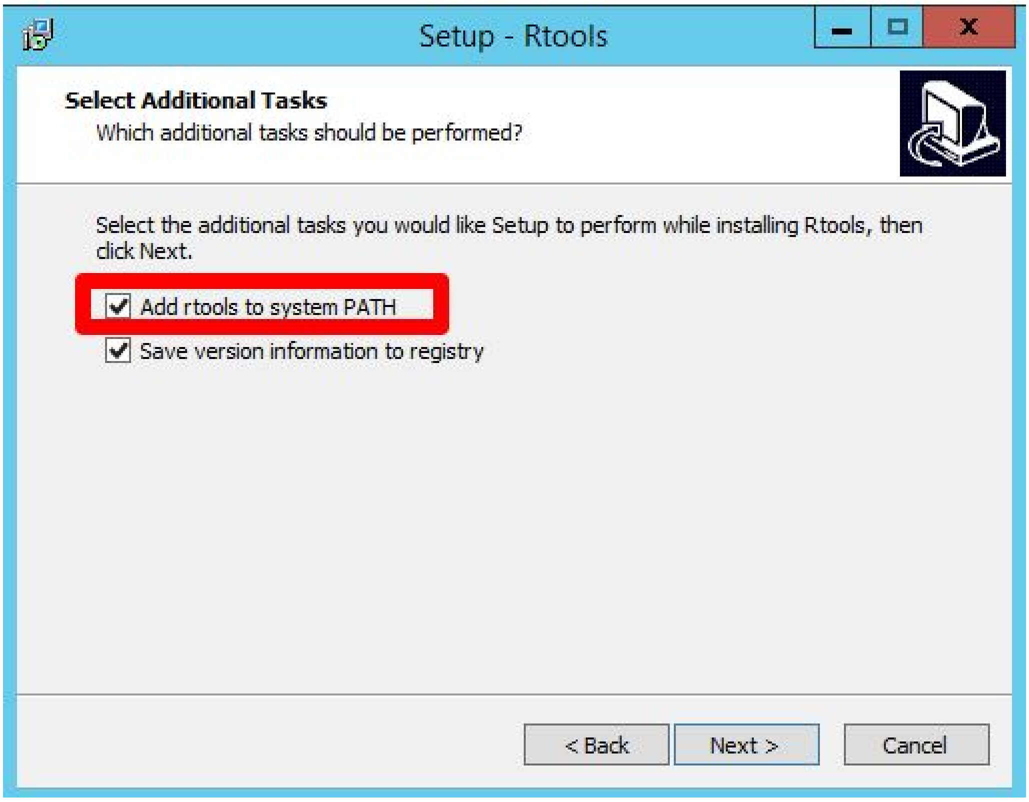
```
install.packages ("C:/Users/Downloads/packfor_0.0-8.tar.gz", repos = NULL,
type = "source")
```

However, installing R packages from source is nontrivial task and may require that your computer has installed set of compilation tools, so called Rtools, because installing from source consists of several steps (compiling C/C++/Fortran codes, making documentations, checking for code errors, running examples etc.) - see instructions below how to install.

Install Rtools (for Windows users only)

Packages hosted on CRAN usually contain package binaries - you can directly install packages without the need to compile them from the source code. If you install packages from GitHub, R-Forge or other repositories, you may need to compile them from source. To do it, do the following:

- download Rtools.exe from Rtools
- if you already have installed an older version of Rtools on your computer, uninstall it (Rtools are usually installed in the folder c:\Rtools)
- follow the installation wizard of Rtools; you can mostly accept the default setting, with one important exception - at the page `Select additional tasks`, please check the box with `Add rtools to system PATH` (screenshot below). In the next step, check whether the folder with Rtools have been added on the top of the list with system paths; if it is and it occurs twice (e.g. because it has been added there by the previous version of Rtools), delete one of them.


Setup - Rtools

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Rtools, then click Next.

☑ Add rtools to system PATH

☑ Save version information to registry

You do not need to install Rtools if

- you use `devtools::install_github` function to install package from GitHub and the package hosted on GitHub does not contain any compiled code;
- you install package from R-forge for the latest version of R, for which R-forge usually offers compiled binaries; however, if you are using older version of R, you still need to build the package from the source (R-forge offers binaries only for the current version of R, not older ones; this is different from CRAN, which offers package for a number of R versions backwards).

[1] This is a link to cloud storage, which will automatically redirect you to the local repository according to your geographical position. If doesn't work, use the manual selection in https://cran.r-project.org/mirrors.html.

[2] The argument `ask = F` in the function `update.packages` makes sure that the update function will not ask you to prompt updating each package separately.

[3] R-x.x.x stems for the R version, e.g. R-3.2.1

[4] Note that both forward selection function from this package have been included in the `adespatial` package available on CRAN.

From:
https://anadat-r.davidzeleny.net/ - **Analysis of community ecology data in R**

Permanent link:
https://anadat-r.davidzeleny.net/doku.php/en:r?rev=1548496495

Last update: **2019/01/26 17:54**