

# Setting up your system for Geo-scripting with R

Loïc Dutrieux, Jan Verbesselt

November 1, 2013

## 1 Introduction

Having your system properly set-up is a crucial step before starting any geo-processing with R. This short tutorial will guide you through some important steps that you should follow in order to have the required software installed and appropriately set-up. Figure 1 provides an overview of the different components of the system and how they are connected with each others.

## 2 On windows

### R and R Studio

For information on how to install R and R Studio, please refer to the [R installation manual for windows](#) and the [R Studio download page](#).

### Rtools

RTools is essential if you want to build R package. It is relatively easy to install, simply download the binary associated with your R version from [this page](#), and follow the instructions from the installer. When prompted for "Edit the system path", you must say yes.

### GDAL/OGR/PROJ4 libraries

Installing these libraries on Windows is not an absolute necessity, however they will come in handy if reprojecting data represent a large part of your work or if you plan on doing a lot of file format conversions. They are an absolute necessity if you have to work with hdf4 format data.

Many options are available for installing these libraries on Windows systems.

- FWTools.
- Installation via OSGeo4W installer.
- By installing Quantum GIS.

The two latter option should be preferred over FWTools since they contain the most up to date versions of gdal.

Note that if you want to call some gdal utilities (i.e.: `gdalwarp`, `gdal_translate`, `gdal_merge.py`, etc) via `system()` command from R, you will need to append the bin directory containing gdal utilities to your user or system path.

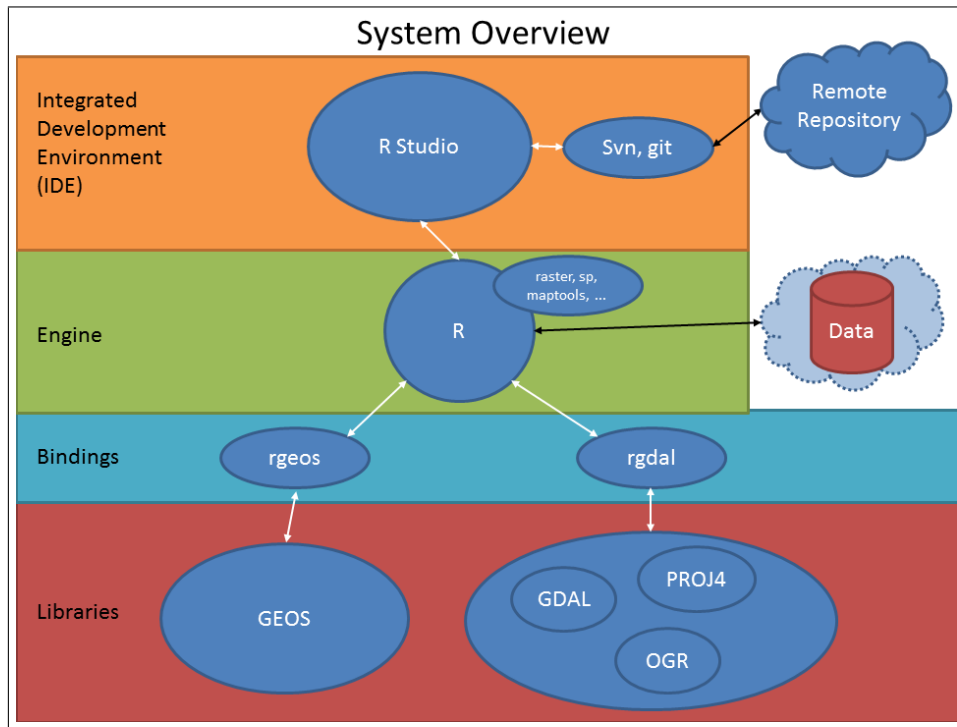


Figure 1: System architecture overview

## Version control software

Installing these software is not an absolute necessity, however it might come handy if you plan to implement some of the good practices discussed in Lesson 2. Refer to Lesson 2 for more details.

### Git and SlickSVN

Most likely you need one OR the other, depending on whether your repository is hosted on a git or Subversion version control system. Only git is required if you plan to follow the full tutorial of Lesson 2.

These software are very straightforward to install, simply download the executables and follow the instructions from the installer. [Git](#) download page, and [slickSVN](#) download page.

### Integration with the R Studio IDE

It is possible that R Studio automatically detects Git and SlickSVN on your system, however this is not systematic. In case the Git/Svn buttons in R Studio are inactive (for instance under project - Create Project - Version Control), after installing either Git or SlickSVN, follow the steps described bellow.

Once SlickSVN (or git) has been installed, in order to allow it to interact with RStudio, its path has to be set. Conveniently that can be done directly from within RStudio. In R Studio, navigate to Tools - Options - Git/SVN - SVN (or Git) - Browse and select the SVN.exe (or git.exe) executable in C:/Program Files/SlickSVN/bin/ (or C:/Program Files/git/bin/) Restart RStudio.

## Specific R packages

Not all packages are hosted on the CRAN package repository, this is the case for instance of the MODIS package that is maintained on R Forge, and requires an extra argument in the `install.packages()` function to be installed. The install line, to be copy pasted in the R shell is often given on the package R forge home page.

### 2.1 Installing the rasta package in Windows

We maintain a windows binaries version of the rasta package under <http://rasta.r-forge.r-project.org/rasta.zip>. Simply download it and install it on your system by pointing to the file you just downloaded in the package installation menu of R Studio.

## 3 Linux

### R and R Studio

Information on how to install R and RStudio on linux systems can be found [here](#) and [here](#).

### GDAL/OGR/PROJ4/GEOS libraries

#### Debian/Ubuntu

The easiest solution is probably to install these library directly from an apt repository. Recent versions of these libraries are maintained on the Ubuntu GIS ppa. First add it to your list of ppa.

```
sudo add-apt-repository ppa:ubuntugis/ppa
sudo apt-get update
```

Then install the libraries with apt-get.

```
sudo apt-get install gdal
sudo apt-get install proj
sudo apt-get install geos
```

Your system should be ready to install `rgdal` and `rgeos` R packages.

#### Redhat/CentOS

## 4 Mac

### 4.1 R and R Studio

Download and install the latest R version available from: <http://cran.r-project.org/bin/macosx/>. Download and install the latest Rstudio Desktop from the [R Studio download page](#).

### 4.2 GDAL/OGR/PROJ4 libraries

Installing these libraries on Mac is not required for the Geo-scripting course as the `rgdal` and `rgeos` R package contain the GDAL C++ library. However, they will come in handy if reprojecting data represent a large part of your work or if you plan on doing a lot of file

format conversions. They are an absolute necessity if you have to work with hdf4 format data.

The best option for install RGDAL on mac is via the [Kynchaos website](#). At the time of writing this description install the **GDAL 1.10 Complete** set if you the GDAL library (not required for the coarse). Check out a [tutorial](#) if you need more info.

### 4.3 Version control software

Installing these software is not an absolute necessity, however it might come handy if you plan to implement some of the good practices discussed in Lesson 2. Refer to Lesson 2 for more details.

Most likely you need one OR the other, depending on whether your repository is hosted on a git or Subversion version control system. Only git is required if you plan to follow the full tutorial of Lesson 2. See here for an excellent and clear description for how [to install SVN and Git on Mac so that you can use it within RStudio](#). RStudio IDE should automatically be able to find SVN and Git.

See here for more info if needed: (1) [extra info for OSX](#). Git should be available to via Xcode, or (2) [this question](#) helps.

### Specific R packages

Not all packages are hosted on the CRAN package repository, this is the case for instance of the MODIS package that is maintained on R Forge, and requires an extra argument in the `install.packages()` function to be installed. The install line, to be copy pasted in the R shell is often given on the package R forge home page.

### 4.4 Installing the rasta package on Mac

```
> install.packages("rasta", repos="http://R-Forge.R-project.org",  
+                  type = "source")
```

### 4.5 Extra help

Other links where you can find help:

- [Problem when install in Git on Mac Lion os x 10.8](#)