

Introduction

Chris Holden

03/24/2015

Tutorial of basic remote sensing and GIS methodologies using open source software (GDAL in R).

Why R?

- Easy to learn
 - Interpreted, dynamically typed programming language
 - Test out your code in R's interpreter as you write your script (see [Rstudio](#) for a lovely IDE)
- Very large community
 - Innumerable examples on blogs, StackOverflow, Github, etc.
 - * See [R-bloggers](#)
 - New scientific algorithms or techniques are often first available as an R package
 - CRAN is huge
- Free

Dependencies

To interact with geospatial data during this tutorial, we will be using the several R packages that utilize the [GDAL](#) library. The R packages that use this library are available as a precompiled binary on Windows. On Linux, installation of GDAL is trivial using your package manager (e.g., `apt-get install gdal-bin libgdal-dev`). Installation on OSX, however, is more difficult. See the [section on OSX GDAL installation](#) for more information.

The following R libraries will be needed for this tutorial:

- raster
- rgdal
- sp
- randomForest

Install them as such:

```
install.packages(c('raster', 'rgdal', 'sp', 'randomForest'))
```

Chapters

1. [Exploring GIS packages in R](#)
2. [Your first remote sensing vegetation index](#)
3. [Plotting and visualizing your data](#)
4. [Importing and using vector data](#)
5. [Classification of land cover](#)

GDAL on OSX

`rgdal` on OSX does not come as a precompiled binary – you must build it yourself against the GDAL library framework. This means that we first must have the GDAL framework installed on OSX.

Download the GDAL, PROJ, and GEOS frameworks from [kyngchaos.com](http://www.kyngchaos.com):

- [PROJ_Framework-4.9.1-1.dmg](#)
- [GDAL_Complete-1.11.dmg](#)
- [GEOS_Framework-3.4.2-4.dmg](#)

Note that these hyperlinks are current as of April 27, 2015, but may be updated or removed from <http://www.kyngchaos.com/>.

Install all three frameworks. Default installation options should be fine. If you choose to install the frameworks in a non-default path, record this path as you will need it for the next step.

With the external frameworks installed, we can now download and compile `rgdal` and `rgeos`. Once again, the versions of `rgdal` and `rgeos` are current as of April 27, 2015, but you may want to check for an updated version.

For `rgdal`:

```
# Download file
rgdal_tgz <- 'http://cran.r-project.org/src/contrib/rgdal_0.9-2.tar.gz'
download.file(rgdal_tgz, destfile='rgdal_source.tar.gz')

# Setup configuration before we compile
# We need to point R to the locations of GDAL and PROJ
config.args <- paste(
  "--with-gdal-config=/Library/Frameworks/GDAL.framework/unix/bin/gdal-config",
  "--with-proj-include=/Library/Frameworks/PROJ.framework/unix/include",
  "--with-proj-lib=/Library/Frameworks/PROJ.framework/unix/lib",
  "--with-proj-data=/Library/Frameworks/PROJ.framework/unix/share/proj")

# Compile and install
install.packages('rgdal_source.tar.gz',
  repos=NULL, type="source",
  configure.args=config.args)

# Check it works
library(rgdal)
```

For `rgeos`:

```
# Load rgdal
require(rgdal)

# Download file
rgeos_tgz <- 'http://cran.r-project.org/src/contrib/rgeos_0.3-8.tar.gz'
download.file(rgeos_tgz, destfile='rgeos_source.tar.gz')

# Setup configuration before we compile
# We need to point R to the locations of GEOS
config.args <- '--with-geos-config=/Library/Frameworks/GEOS.framework/unix/bin/geos-config'
```

```
# Compile and install
install.packages('rgeos_source.tar.gz', repos=NULL, type='source', configure.args=config.args)

# Check it works
library(rgeos)
```