

# **Applied Geo-Scripting: Lesson 1**

Jan Verbesselt, et al.

October 7, 2013

# Geo-scripting learning objectives

- Learn to read, write, and visualize spatial data (vector/raster) using a script
- Know how to find help (on spatial data handling functions)
- Solve scripting problems (debug, reproducible example, writing functions)
- Find libraries which offer spatial data handling functions
- Learn to include functions from a library in your script
- Apply learned concepts in a case study: learning how to address a spatial/ecological/applied case (e.g. detect forest changes, flood mapping, ocean floor depth analysis, bear movement, etc.) with a raster and vector dataset.

# Today's topics

- Intro to basic concepts of applied scripting for spatial data
- Why geo-scripting?
- Course planning and practical issues
- Getting Us up to speed with R and the 'rasta' package

# Why geo-scripting?

- Reproducible: avoid clicking and you keep track of what you have done
- Efficient: you can write a script to do something for you e.g. multiple times e.g. automatically downloading data
- Enable collaboration: sharing scripts, functions, and packages
- Good for finding errors i.e. debugging e.g. this course is fully writing with scripting languages (i.e. R and Latex).

# Get Your R On

**Getting started with Rstudio and the rasta package** This preliminary section will cover some basic details about R. For this course we will use Rstudio as an IDE to write and run scripts. Open Rstudio! Now type the following script in the R console:

```
R> a <- 1
```

```
R> a
```

```
[1] 1
```

The first line you passed to the console created a new object named *a* in memory. The symbol '*<-*' is somewhat equivalent to an equal sign. In the second line you printed *a* to the console by simply typing its name.

**What is the class of this object?**

# Get Your R On

```
R> class(a)
[1] "numeric"
```

You now have requested the **class** attribute of *a* and the console has returned the attribute: **numeric**. R possesses a simple mechanism to support an object-oriented style of programming. All objects (*a* in this case) have a class attribute assigned to them. **R** is quite forgiving and will assign a class to an object even if you haven't specified one (as you didn't in this case). Classes are a very important feature of the **R** environment. Any function or method that is applied to an object takes into account its class and uses this information to determine the correct course of action.

# The rasta package

```
R> library(rasta) ## load the rasta library  
R> #?rasta
```

# Reading Data in and Out

The most common way to read in data is with the `read.csv()` command. Type `?read.table` in your console for some other examples.

```
R> f <- system.file("extdata/kenpop89to99.csv", package="rasta")
R> mydat<-read.csv(f)
```

We can explore the data using the `names()`, `summary()`, `head()`, and `tail()` commands (we will use these frequently through out the exercise)

```
R> names(mydat) #column names
```

```
[1] "ip89DId"      "ip89DName"    "ADMIN3"       "KEADMN3_ID"  "Y89Pop"
[6] "Y89Births"    "Y89Brate"     "Y99Pop"       "Y99Births"   "Y99Brate"
[11] "PopChg"       "BrateChg"
```

```
R> summary(mydat$Y89Pop)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
57960  222900  451500  619700  947500 1476000
```

```
R> head(mydat$Y89Births)  # see ?head
```

```
[1] 42560 27720 10980 27940 12520 17540
```

We will go over ways to index and subscript data.frames later. Lets do a basic regression so you can see an example of a list.



# Basic regression and example of a list

We use the `lm()` command to do a basic linear regression. The `~` symbol separates the left and right hand sides of the equation and we use `'+'` to separate terms and `'*'` to specify interactions.

```
R> myreg<-lm(Y99Pop~Y89Births+Y89Brate,data=mydat)
```

```
R> #Regress the Population in 1999 on the population and birthrate in 1989
```

```
R> myreg
```

Call:

```
lm(formula = Y99Pop ~ Y89Births + Y89Brate, data = mydat)
```

Coefficients:

(Intercept)	Y89Births	Y89Brate
502592.59	38.05	-14369.09

**Excercise Lesson 1: Write you own function to find out what the "class" and "average" of all objects in...**

## More information

For more information about R please refer to the following links  
<http://www.statmethods.net/index.html>. This is a great website for learning R function, graphs, and stats. Also visit  
<http://www.r-project.org/> and check out the Manuals i.e an introductions to R See also the book on Applied spatial Data analysis with R <http://www.asdar-book.org/> (Bivand et al., 2013).

Bivand, R. S., Pebesma, E. J., & Rubio, V. G. (2013). Applied spatial data analysis with R, .