



R Programming For Ecologists: A Short Course



NCEAS Scientific Computing Team



R Course Objectives

- **Install /Configure R on your computer platform**
 - MS Windows, UNIX/Linux, Mac OSX
 - Extending your R installation with additional applications packages
- **Learn and practice use of R features**
 - Command line and graphical user environments
 - R analytical components and data objects
 - Data editing and visualization tools
 - R script development and execution
- **Using R to Interact with your data**
 - Ingestion / Management
 - Exploration
 - Visualization
 - Computation
 - Export and Publication
- **Special Topic: Geospatial data and geostatistical analysis**
 - Using application-specific R packages in your work



Outline for First Class Meeting

- **At end of first meeting you will know how to:**
 - Obtain R software and documentation online
 - Download and install R on your platform of choice
 - Use the R command line to process vectors and matrices
 - Import, process, visualize, and export tabular numerical data
- **R Language Overview: The Major Components**
 - R Language Core
 - Add-on packages: extend the core
 - Online software and information sources
- **Installing R on your platform**
 - Base package + applications-specific packages
 - Running R on your desktop computer: Windows/Mac/Linux
 - Using the NCEAS Linux Servers for long-running R processing jobs



Outline for First Class Meeting (ctd)

- **Fundamentals of R usage**
 - Identifying the best R components (objects) for your tasks
 - Using the Command Environments
 - Working with 'foundation' Data Objects:
 - Data flow into, within, out of R programs
 - Review R objects for data import and export
- **Use Case 1: Excel Spreadsheet dataset processing**
 - Import Excel Spreadsheet into an R Data Frame
 - Using the R Data Frame object
 - Perform some basic statistical analysis and visualization
 - Export results to two common file formats
- **Use Case 2: Import/Export binary data**
 - A Quick Look: using the *foreign* package



What is R?

- **An open source environment for statistical computing and visualization**
 - GNU/GPL version of the S Language from Bell Laboratories
 - Provides variety of statistical and graphical techniques
 - Highly *extensible*: Add application-specific components
- **Integrated suite of software facilities for data manipulation, calculation, graphical display**
 - Effective data handling and storage facility
 - Operators for calculations on vectors/matrices
 - Intermediate tools for data analysis
 - Graphical facilities for data analysis and display
 - Well-developed programming language

* Source: Introduction to R documentation



R Technical and Educational Support

- **Comprehensive Online Help**
 - Best on the MS Windows platform
- **Documentation included with R packages**
 - MS Windows: accessible through the R online help
- **CRAN Web site**
 - <http://www.r-project.org/about.html>
 - In-depth description of R environment
 - <http://cran.stat.ucla.edu/src/contrib/PACKAGES.html>
 - Listing of approved R analytical packages
- **R Newsletter**
 - Special Interest Articles, new package announcements
- **Special-Interest Groups and Mailing Lists**
 - Example: R-sigGEO: R geospatial/geostatistical data analysis community



Selected R documents for new users

- **Emmanuel Paradis: R for Beginners**
 - Institut desSciences de l'Evolution
 - Very readable general introduction
- **Mark Venables: An Introduction to R**
 - From the Core R Development Core Team
 - Good wide ranging introduction to R
- **John Verzani: simpleR: Using R for Intro Statistics**
 - Notes designed for an intro stat course taught using R
 - Complete in-depth examples of statistical problems solved in R
- **Core R Team: R Installation and Administration**
 - How to install and maintain R on major computer platforms
- **J H Maindonald: Using R for Data Analysis/Graphics**
 - Good examples of stat analysis and graphics production
- *Links to all of these (and others) on Sci Comp Web Site*



Installing R on the Windows XP Platform

- **Download from CRAN site:**
 - <http://www.r-project.org/> : R home page
 - <http://cran.stat.ucla.edu/> : closest ftp mirror site
- **Install the 'base' R software package**
 - Self-extracting installer
- **Find, install R application extension packages**
 - Packages developed, packaged, distributed in standard format
 - Many include self-extracting installer for Windows version
 - Use the **Packages** option in the *Rgui* toolbar
 - Web site listing many (**not** all) contributed R packages:
 - <http://cran.stat.ucla.edu/src/contrib/PACKAGES.html>
 - Also use Google keyword search to find other R packages



Installing R on UNIX/LINUX and Macintosh

- **Many LINUX distros include the base 'R' system**
 - E.G., Ubuntu
- **OR, Install the 'base' R software package**
 - Option 1: Installation using R source code
 - Option 2 (some platforms): UNIX binary images
 - Details: <http://cran.r-project.org/doc/FAQ/R-FAQ.html>
- **Find, install R application extension packages**
 - R command line: `install.packages(pkgname,dependencies=TRUE)`
 - Must be root (super) user to modify the R installation
- **Macintosh (Mac OS X)**
 - Obtain self-installing package from CRAN download site
 - Source Code also available
 - Mac OS 8 – 9 versions available but no longer supported
- *Also consult the R Installation and Administration manual*



Running R Remotely on NCEAS Servers

- **R is installed on *Muse* and *Zeus* LINUX servers**

- Fast, multiprocessor machines for long-running jobs
- To use:
 - Transfer your R scripts and data via network/shared hard drive.
 - Get Secure Shell and X windows client software
 - Start SSH and X client sessions
 - Log into Linux server using your NCEAS account
 - Invoke R from UNIX command line:
 - **/usr/bin/R**

- **Suggested Strategy for using the NCEAS Servers**

- Develop R scripts on local desktop machine....
- ...Then port to Linux servers for large dataset jobs



R Architecture: Data Object-centered

- **Architecture understanding enhances use of R**
- **R is an Object-Oriented programming tool**
 - Enhances flexibility and power of the components
- **Data and Processing are Inseparable**
 - Objects consist of descriptive attributes and behaviors
- **Objects Communicate via Messaging**
 - Through predefined interfaces
- **Inner workings of Objects are hidden from users**
 - This way, they can not be inadvertently modified or 'corrupted'
- **Existing objects form the foundation for new, more capable objects**
 - R objects: components that can be assembled in many ways
 - Programmers extend existing components to do new things



Options: The R User Interface

- **R from the Command Line: `<path>/R`**
 - Works the same on all platforms
 - Powerful: do calculations, enter/view/edit/display/plot data
 - Uses platform's native graphical window
 - X window manager for Unix/Linux and SSH access
 - Command line history and editing
 - Scripting capability: Write and run R functions (scripts)
 - **Includes R-supplied Data Editor: `data.entry()`**
 - 'Grid entry' tool for entering matrices of data
 - Popular alternative: Use 3rd-party editor; import data as 'data frame'
- **MS Windows Graphical User Interface: `Rgui()`**
 - MS Windows 'look and feel' application with menu bars, etc
- **Multi-Platform Graphical Command Tool: `Rcmdr()`**
 - Add-on R package provides point-and-click control of R tools
- **Coming Soon: Web-based R User Interface: `Rweb`**
 - <http://www.math.montana.edu/Rweb/>



Using R: The Basics

■ Starting the R environment

- ***Rterm***: Text-based command environment
 - Web browser needed for Help package
 - Window manager needed for data editor and visualization
- ***Rgui***: Integrated Graphics Environment
- ***Rcmd***: Batch processing of long-running R jobs (best on UNIX)

■ Moving data into the R environment

- Keyboard Entry: the `c()` function
- Built-in R Data Editor
- Read from file: `scan()`, `read.csv()`, `read.delim()`, `read.table()`
- Read from standard binary data interfaces: ODBC, SAS, ArcGIS
- Calculate it: `rnorm()`, other distributions
- Data will be ingested into an appropriate R data object
- Example Session: **Venables**, pp. 76 - 78



Overview: The R Workspace

- **In an R session, software and data components are created and remain active in memory**
 - Data that you create during your analysis
 - User-written R functions loaded via the `source()` method
- **The collection of currently active objects is known as the *workspace*.**
 - `objects()` or `ls()` methods display the workspace contents
 - `rm(name1,name2....)` method removes objects from workspace
- **The workspace is saved at the end of your session**
 - You will be asked when you exit R via the `q()` method
 - Default name is “.RData”
 - At any time, save the workspace `save.image()` method
 - To activate any workspace, use the `load()` method



R Components: Data Objects and Functions

- R data *objects* (structures) are *data containers*
 - The *foundation* object is the *vector*
 - Entity consisting of an *ordered collection of elements*
 - *Functions* perform operations on vector contents
 - Specialized vector classes contain different data types
 - *character, complex, index, logical, numeric, raw*
 - Other R objects are *collections* of vectors
 - Good vector discussion: Venables, Chapter 2
 - R *functions* operate on the objects' contents
 - *Functions contain processing algorithms, other instructions*
 - *Functions adapt their processing behavior to different objects*
 - *Example: sort() function works differently on integer vectors than on character vectors*
 - *To adapt, functions read the objects' attributes*
 - **Intrinsic functions: built in to R environment**
 - **User-Written functions add new capability to R**



R Components: Data Objects (ctd)

- **Attributes** (properties) define objects' *behavior*
 - Good Discussion: Venables, *Chapter 3*
 - Two key Attributes of any R data object:
 - **Mode**: Atomic type of data elements in the object
 - *character, complex, logical, numeric, raw*
 - **Length**: Number of elements in the object
 - *Object length can be changed during program execution*
- **The Major R Data Objects**
 - **Vector**: A collection of elements. Foundation for others
 - **Time Series: Special Case of Vector**
 - Paradis: pp 19 – 20
 - **Array / Matrix** (vector collection): Venables: pp. 18 – 20
 - **Factor** (categorical data): Venables: pg. 16.
 - **List**: Venables: pp. 26 - 27
 - **Data Frame**: Venables: pp. 27 - 28



R Components: Functions

- **Functions: Building Blocks of the R Environment**
 - Collections of R statements executed as a module
 - Functions typically contain three types of statements
 - *Data declaration statements*
 - *R commands*
 - *Flow-of-control logic (looping, branching)*
 - Most intrinsic R commands are implemented as functions
 - *Any statement ending in '()' is a function*
 - *R functions 'loosely coupled' to data objects*
 - *Not encapsulated in the object, but read objects' attributes*
 - *Functions change their behavior based on the object type*
- **User-written functions: extend your R environment**
 - Load them into your Workspace using the `source()` command
 - They become part of the R command set
 - Functions and data objects written for one R platform are portable
 - Example: Develop a function for R under Linux....
 - ...the function should work in Windows or Mac R installation



Using R for Data Exploration

- **Break data exploration into four phases:**
 - ***Ingestion***: Move data into the R Environment
 - ***Exploration***: Preliminary analysis and visualization
 - Look for 'low frequency' trends, outliers, anomalies
 - Formulate ideas for more rigorous analysis
 - ***Analysis***: *Transformation, hypothesis testing, etc*
 - *Exploration and Analysis usually involve **Visualization** phase*
 - ***Export***: Move the transformed data out of R environment
- **R environment well-suited to Data Exploration:**
 - Includes functions (tools) for each phase
 - The R environment captures a record of all sessions in a ***workspace***
 - The workspace is saved to named disk file at end of each session
 - **Can be explicitly recalled in a future R session**



Use Case: Sample R Data Exploration

- **Demonstrate four methods for ingesting data**
 - From keyboard into a set of R vectors
 - From ASCII .csv file into R **matrix** and **table**
 - Import from .csv and .dbf files into R **data frame**
- **Explore it using R exploration tools**
 - Generate plots of input and transformed data
 - Export the results to external files
 - **ASCII**
 - **Binary**
 - Demonstrate the *foreign* and *ODBC* R packages
 - Suggested text: ***Verzani: SimpleR.pdf***
- **Save the commands used to an R script**
- **New R session: Load and execute the script**
 - Demonstrate R script development
 - Use your favorite text editor in conjunction with R



Using R: Suggested Exercise for Week 1

- **Download and Install R on your computer**
 - R environment and core method set
 - Locate and install add-on packages for your specialty
- **Select one of your research datasets**
 - Preferably one that fits the 'data frame' format
- **Use R to explore the data**
 - Refer to the suggested tutorial documents (and Internet) for examples