



R Programming For Ecologists: Week 2



Rick Reeves: NCEAS Scientific Computing Team

Meeting 2 Objectives



- **Overview: R Data Types**
 - The R data object hierarchy
 - Operations defined on each object type
- **R Data Import and Export Capability**
 - Moving data between R and other applications
- **Elements of R Data Visualization**
 - Overview: Plotting and graphing using R
- **Case Study: End-To-End Data Exploration**
 - Import data to the R Workspace
 - Four data analysis case studies
 - Drawn from *Verzani: SimpleR (for intro statistics)* text
 - Demonstrate data handling and simple analytical methods



R Object Hierarchy and Operations

- R operates on entities known as **objects**
 - Intrinsic object attributes *mode* and *length*
 - **Mode**: type of data stored in the object
 - **Length**: number of elements stored in the object
 - The foundation object is the **vector**
 - **Collection of elements, all of same type (mode)**:
 - numeric, complex, logical, character, raw
- R object hierarchy
 - Mathematics operations are defined on vectors
 - **Simple examples**: *Paradis: R for Beginners*, pp 7 – 11
 - Notable: **index vectors** select/modify a subset of a (vector) data set
 - vector -> array -> matrix -> factor -> list -> data frame -> function
 - **Each object inherits the attributes from 'parent', adds new ones**
 - **Brief definitions**: *Venables*, pp. 11 – 12
 - `class()` method reports the *type* of an object



R Object Hierarchy and Operations (ctd)

- R object hierarchy (ctd)
 - **Example: array is vector with 'dim' attribute (row/column order)**
 - Assign using vector's **dim()** attribute or use the **array** method()
 - Operate on specific parts of array using an ***index array***
 - *Venables*, pp. 18 - 19
 - **matrix**: array with two or more dimensions
 - Special matrix mathematics operations available: *Venables*, pp. 22 - 25
 - **factor**: discrete categorical variables
 - Simple example: *Venables*, pp. 16 - 17
 - **list**: an ordered collection of objects known as the list ***components***
 - The components can be of different types
 - **data frame**: collection of (possibly) any other R object
 - Accessed and displayed as a matrix: each column can be different mode
 - List and data frame fundamentals: *Venables* Chapter 6
 - **function**: user-developed program modules integrate into R



R Object Hierarchy: Focus on Vectors

■ Vectors

- The foundation class for other data objects
- Many examples: first chapters of *Paradis*, *Venables*, *Venzani*
- Widely-used intrinsic functions on vectors: *Paradis*, pp 21 - 25
- Time Series: Special Case of Vector
 - A regular sequence of numbers, user-defined range and interval
 - `ts()` method: *Paradis*: pp 19 – 20
- Other regular sequence generators:
 - `seq()`, `rep()`: *Venables*: pp 8 – 9
 - `rnorm()`, `runif()`: pseudo-random number generators
 - *Venzani*: pp 41 – 42

■ Logical Vectors

- Manipulation of logical (TRUE, FALSE, NA) values
- Examples: *Venables* pp 9 – 10, *Venzani* pg 6



R Object Hierarchy: Focus on Vectors (ctd)

■ Index Vectors

- Used to select the subset of a vector
- Contain any of four data types, see *Venables*, pp. 10 – 11
 - Results of a logical expression
 - Positive indexes (inclusion)
 - Negative indexes (exclusion)
 - Character Strings (surrogate for numeric indexes)

■ Common use: efficiently winnowing large datasets

- matrixBenchmark.r: Sample R script ‘filters’ large 2-dim matrix

■ Character Vectors (strings)

- Operations: conversion / contatenation / parsing / substring / search
- *Venables*: pp 10 -12
- Web site handout: ‘Handling text in S+ and R’



Examples: R Data Objects

- **Vectors:**

- Venables, Chapter 2, Venzani: pp 8 – 10

- **Factors:**

- Venables: pp 16 – 17, 24 - 25
- Venzani: pp 8 – 10

- **Arrays and Matrices:**

- Venables: pp. 18 – 20
- Paradis: pp. 17 - 18

- **Lists:**

- Venables: pp. 26 – 27
- Sample Program: <TBA>

- **Data Frames:**

- Venables: pp. 27 – 28
- Sample Program: <TBA>



Importing / Exporting Data in R

- **R saves contents of your session as part of workspace**
 - ..and it is read back at the start of the next session
 - Drawback: Data are in internal R format
 - Use import/export facilities to exchange with 'external' format
- **Generally, two import/export options**
 - Exchange via delimited ASCII file
 - R method `read.table()` (and variants)
 - Exchange with external file formats via add-on R package
- **ASCII Data import: the `read()` method**
 - `read.table()`: reads comma-delimited ASCII file, creates data frame
 - `read.csv()`, `read.delim()`... also create data frame
 - But have different default input parameters
 - `read.fwf()`: reads fixed-width format ASCII file
 - `scan()`: Read data into a vector or list from the console OR file.



Importing / Exporting Data in R (ctd)

- **ASCII Data export**

- **write.table()**: writes data frame to ASCII text file.
- **write()**: writes (matrix) data to an non-delimited ASCII text file

- **Import/Export with common external file formats**

- **MS Excel**: read.table() using .csv version of Excel spreadsheet
- **RDBMS**
 - ORACLE: ROracle package
 - ODBC-enabled DBMS (and other apps): RODB
- **Commercial Statistics Packages**
 - SAS: *RODBC* or *Design* add-on packages OR user-developed method
- **GIS**
 - **GRASS**: several options, e.g., *SPGRASS* package
 - **ArcMap GIS**: R reads and writes **ESRI Shape Files**
 - *sp* AND *spmaptools* packages required
 - **Demonstrations with Excel and ArcMap GIS**



Details: R List and Data Frame Manipulation

- R syntax has many productivity – enhancing features
 - Two examples are **attach()** and **(un)stack()**
- **attach(list or frame)**: connects the dataset to R search path
 - Dataset elements are directly accessible through their names
 - No need to use the '\$' notation “list\$varname”, just “varname”
- **unstack(dataset,formula)**: produces a list of columns according to the dataset’s formula
 - Breaks dataset up into groups based on the internal structure..
 - Which can then be explored separately..
- **stack(dataset)**: concatenates columns into a single vector
 - Reverses the effect of **stack()**
 - Makes note of the internal data organization
 - Output can be passed to other R functions requiring a single vector
- These are only two examples of R data handling ‘tricks’
 - Many, many more to learn once you get started....



Sample R Data Analysis sessions

- From *Verzani* : *SimpleR: Using R for Introductory Statistics*
 - See how R implements simple statistical procedures
- **Handling Categorical Data**
 - *Verzani*, pp 21- 24
 - R plotting fundamentals
- **Linear regression**
 - *Verzani*, pp 24 - 27
 - Direct via `lm()`, indirect via `abline()`
- **Multivariate data in Data Frames**
 - *Verzani*, pp 32 - 40
 - `attach()` and `(un)stack()` functions: examples of productivity tools
 - R built-in model and formula notation
- **Exploratory Data Analysis**
 - *Verzani*, pp 54 - 59
 - The elements of graphing and plotting in R
 - ALSO Subject of next class session