

R Installed

ape

ouch

geiger

laser

picante

phylobase

Getting Started in R for Biologists

Marguerite Butler
University of Hawaii

Sponsored by: NSF, NESCENT

Thanks!

the R environment

An integrated suite of software facilities:

A fancy calculator

Data Management Handling and storage

Matrix Math: Manipulating matrices, vectors, and arrays

Statistics: A large, integrated set of tools for data analysis

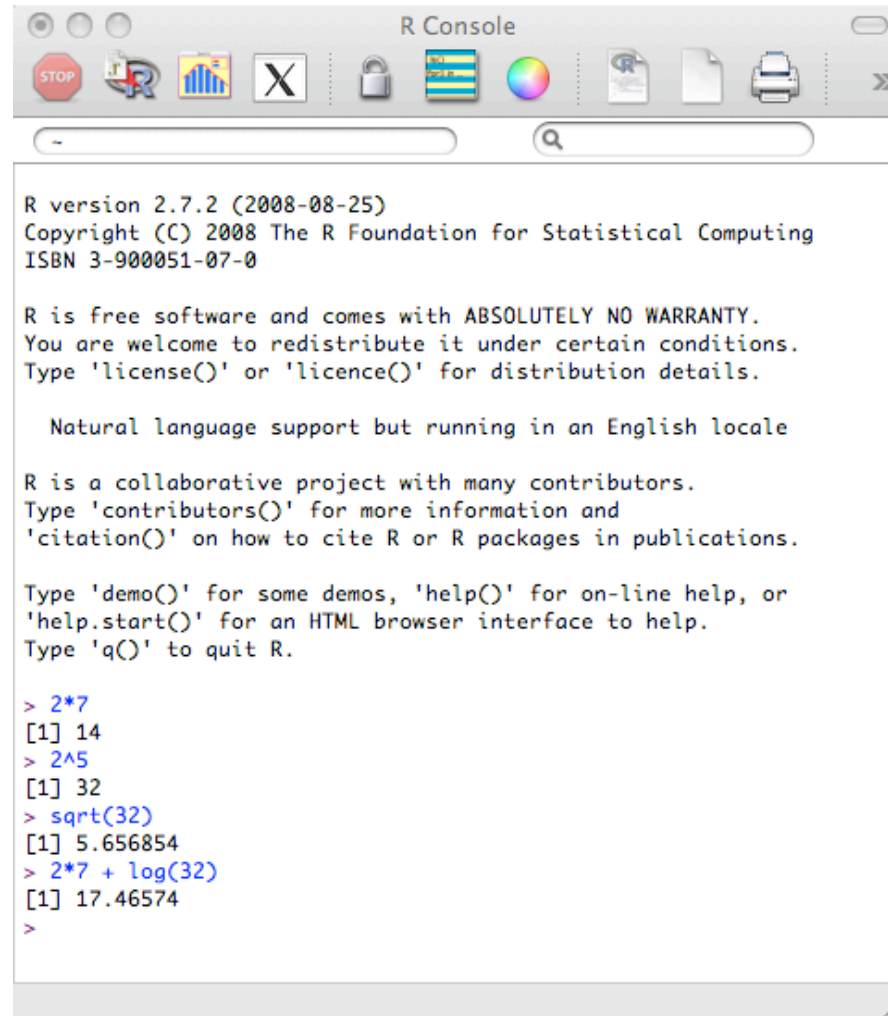
Graphics: Graphical facilities for data analysis and display

Programming: Powerful programming language ('S')

Open-Source Development Platform

the R environment

A fancy calculator



```
R Console

R version 2.7.2 (2008-08-25)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 2*7
[1] 14
> 2^5
[1] 32
> sqrt(32)
[1] 5.656854
> 2*7 + log(32)
[1] 17.46574
>
```

the R environment

an integrated suite of software facilities:

Data Handling and Storage

```
> morph <- data.frame(species=LETTERS[1:5], size = rnorm(5, mean=15))
> morph
  species      size
1      A 13.38846
2      B 14.83139
3      C 16.68702
4      D 12.42916
5      E 17.32852

> eco <- data.frame(species=LETTERS[5:1], ecology = sample(c("a", "b", "c"), 5,
replace=TRUE))
> eco
  species ecology
1      E      a
2      D      b
3      C      a
4      B      c
5      A      b

> merge(morph, eco)
  species      size ecology
1      A 13.38846      b
2      B 14.83139      c
3      C 16.68702      a
4      D 12.42916      b
5      E 17.32852      a
```

R has facilities for basic database functions:
merging, matching, string matching, file access

the R environment

an integrated suite of software facilities:

Data Handling and Storage

Matrix Math

```
> x <- matrix( data= 1:6, nrow=2)
```

matrix

```
> x
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> t(x)
```

matrix transpose

```
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

```
> y <- matrix( data = rnorm(9), nrow=3)
```

```
> y
```

```
      [,1]      [,2]      [,3]
[1,]  0.07969564 -0.04395246 -0.11727169
[2,] -0.01708504 -0.15159683  0.13944474
[3,]  0.56229980  0.25573414 -0.05902727
```

matrix multiplication

```
> x %*% y
```

```
      [,1]      [,2]      [,3]
[1,]  2.839940  0.7799277  0.005926158
[2,]  3.464850  0.8401126 -0.030928068
```

matrix inverse

```
> solve(y)
```

```
      [,1]      [,2]      [,3]
[1,]  1.779046  2.170144  1.5922018
[2,] -5.154919 -4.078422  0.6066952
[3,] -5.386179  3.003348  0.8546460
```

the R environment

an integrated suite of software facilities:

Data Handling and Storage

Matrix Math

Statistics

Linear Models

ANOVA

Non-parametric Statistics

Multivariate Statistics

Time Series

Numerical Methods

Optimization

etc. etc.

the R environment

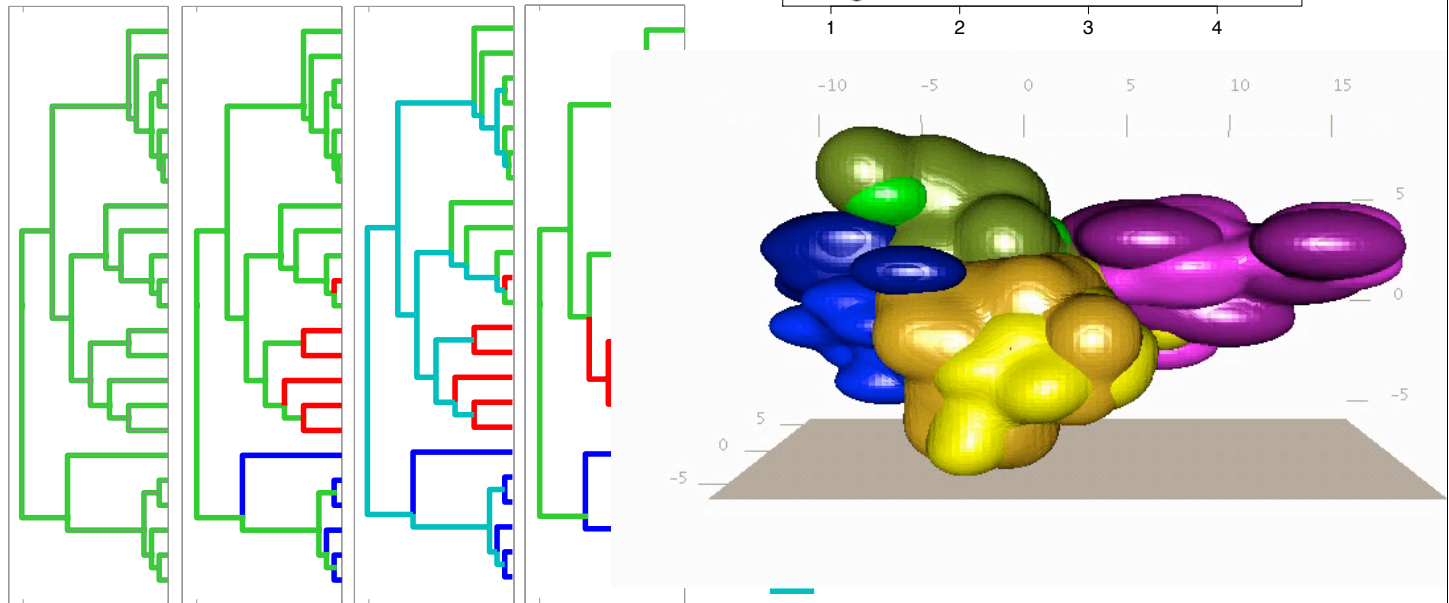
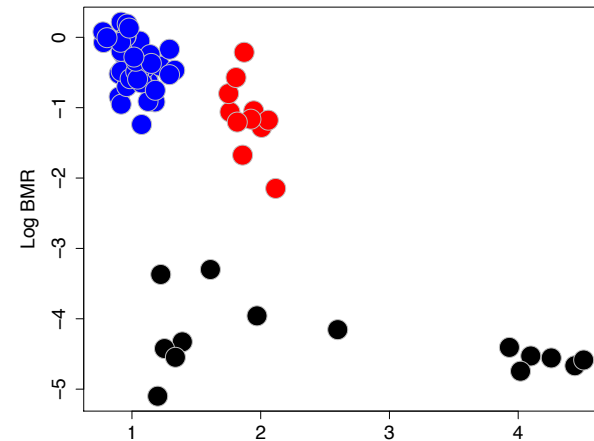
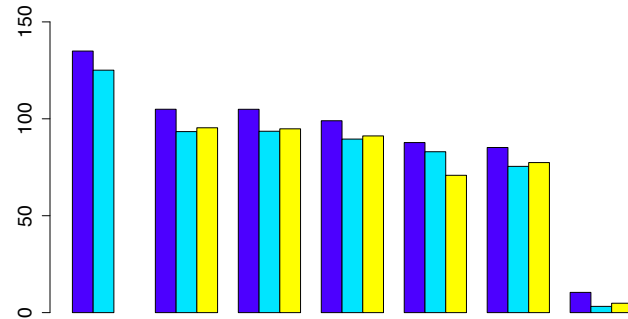
an integrated suite of software facilities:

Data Handling

Matrix Math

Statistics

Graphics



the R environment

an integrated suite of software facilities:

Data Handling and Storage

(semi) Object-Oriented Design

Matrix Math

Conditional Expressions

(Loops)

Statistics

(Recursion)

Vectorized Calculations

Graphics

Functions

Packages

Programming

Extensibility

the R environment

an integrated suite of software facilities:

Data Handling and Storage

Matrix Math

Statistics

Graphics

Programming

Open-Source Community

the R environment

an integrated suite of software facilities:

R Homepage

Data Handling and Storage

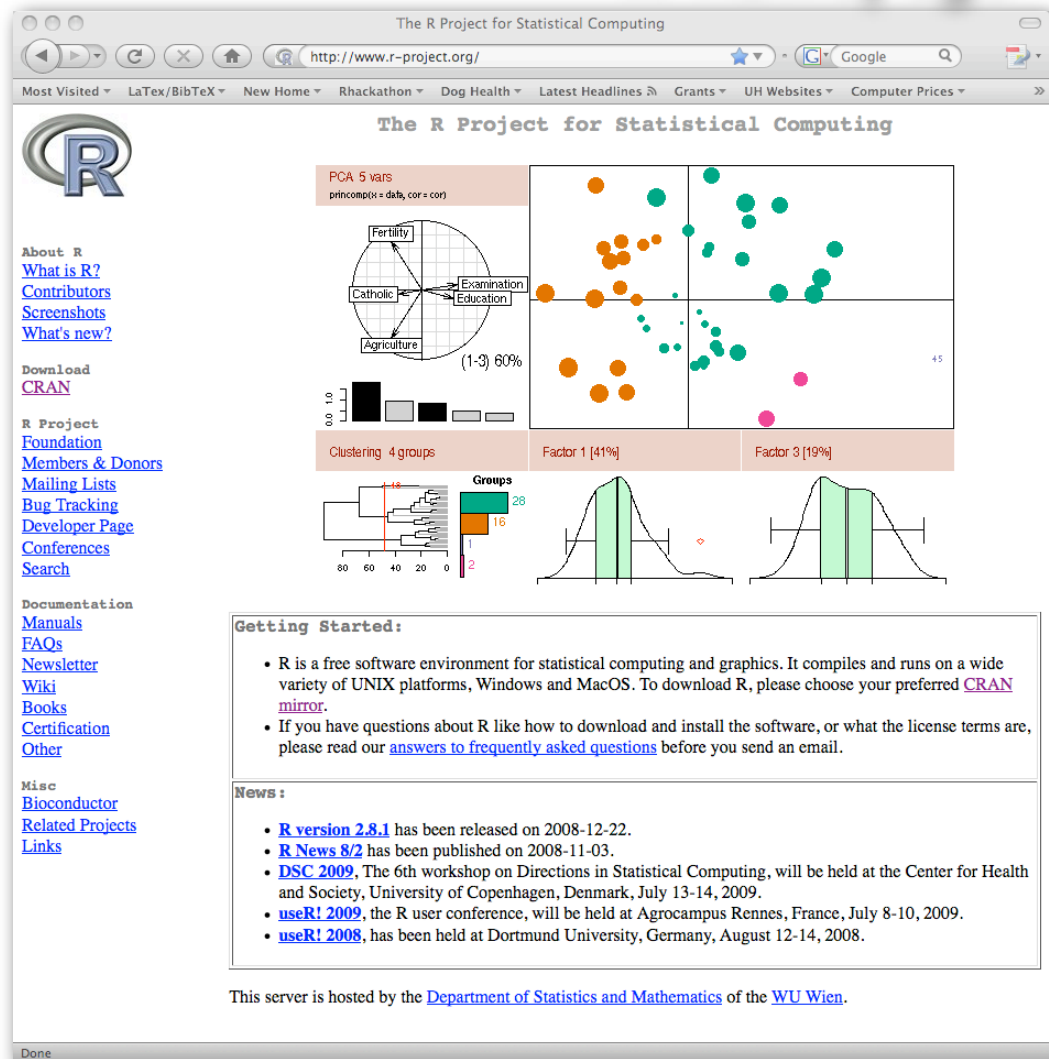
Matrix Math

Statistics

Graphics

Programming

Open-Source Community



the R environment

an integrated suite

Data Handling and Storage

Matrix Math

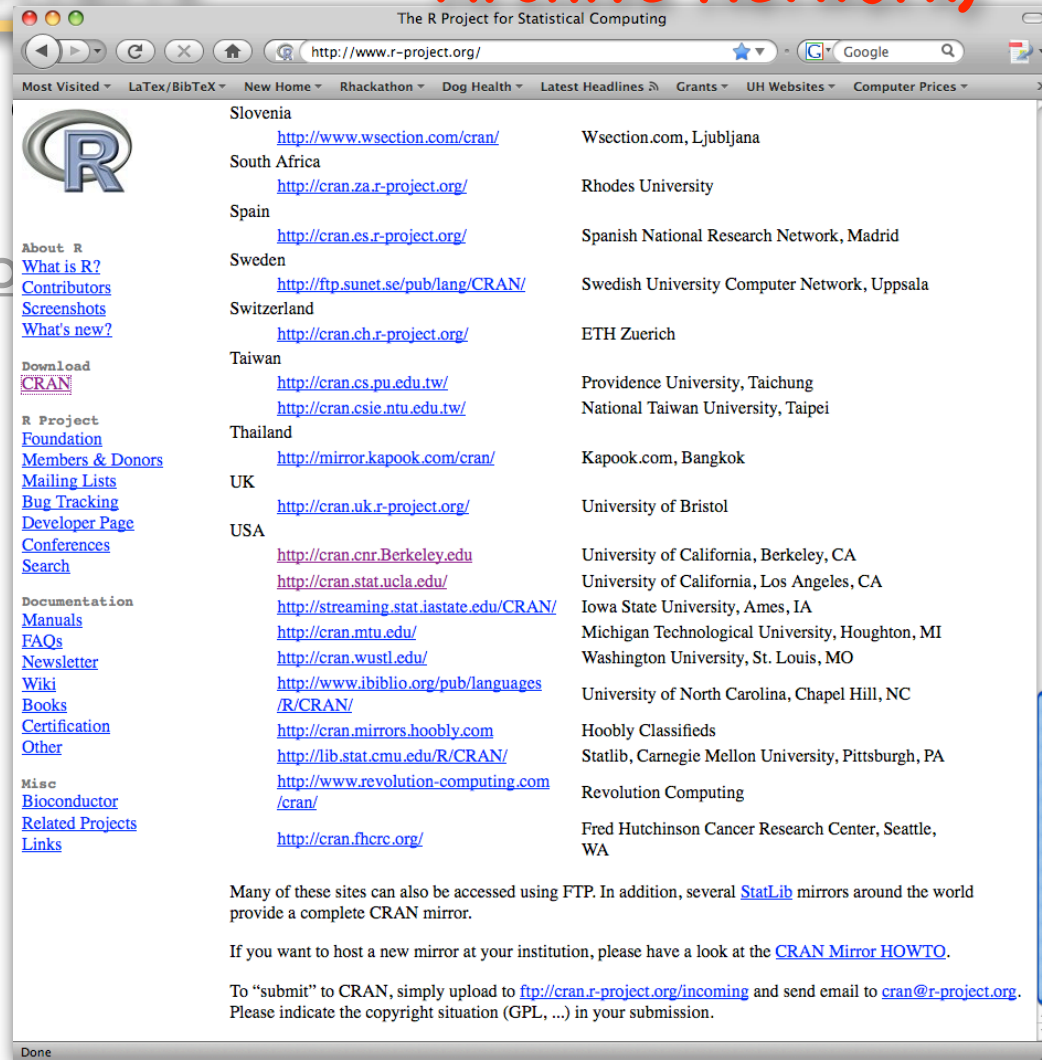
Statistics

Graphics

Programming

Open-Source Community

CRAN (Comprehensive R Archive Network)



the R environment

Support for all Major Platforms

an integrated suite

Data Handling and

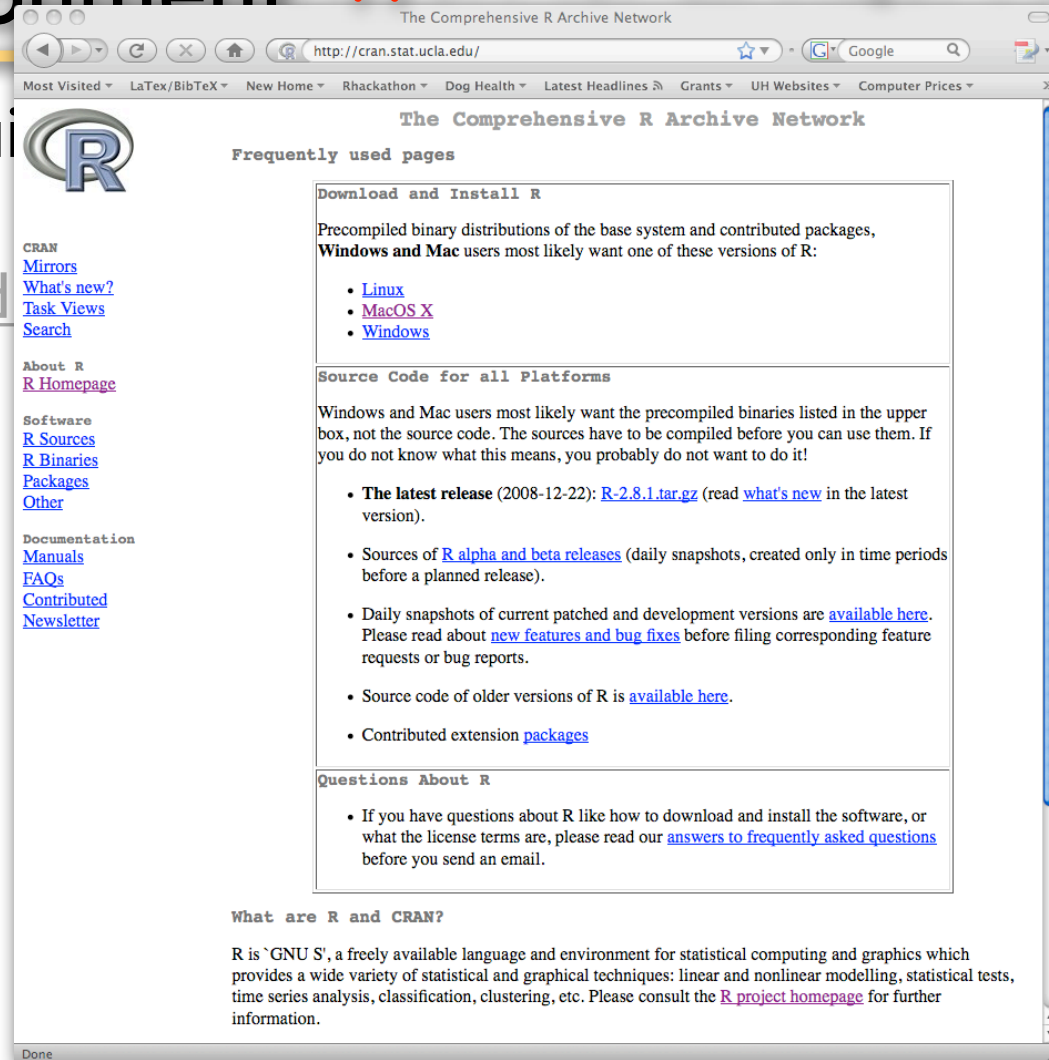
Matrix Math

Statistics

Graphics

Programming

Open-Source Community



the R environment

an integrated suite

Data Handling and Storage

Matrix Math

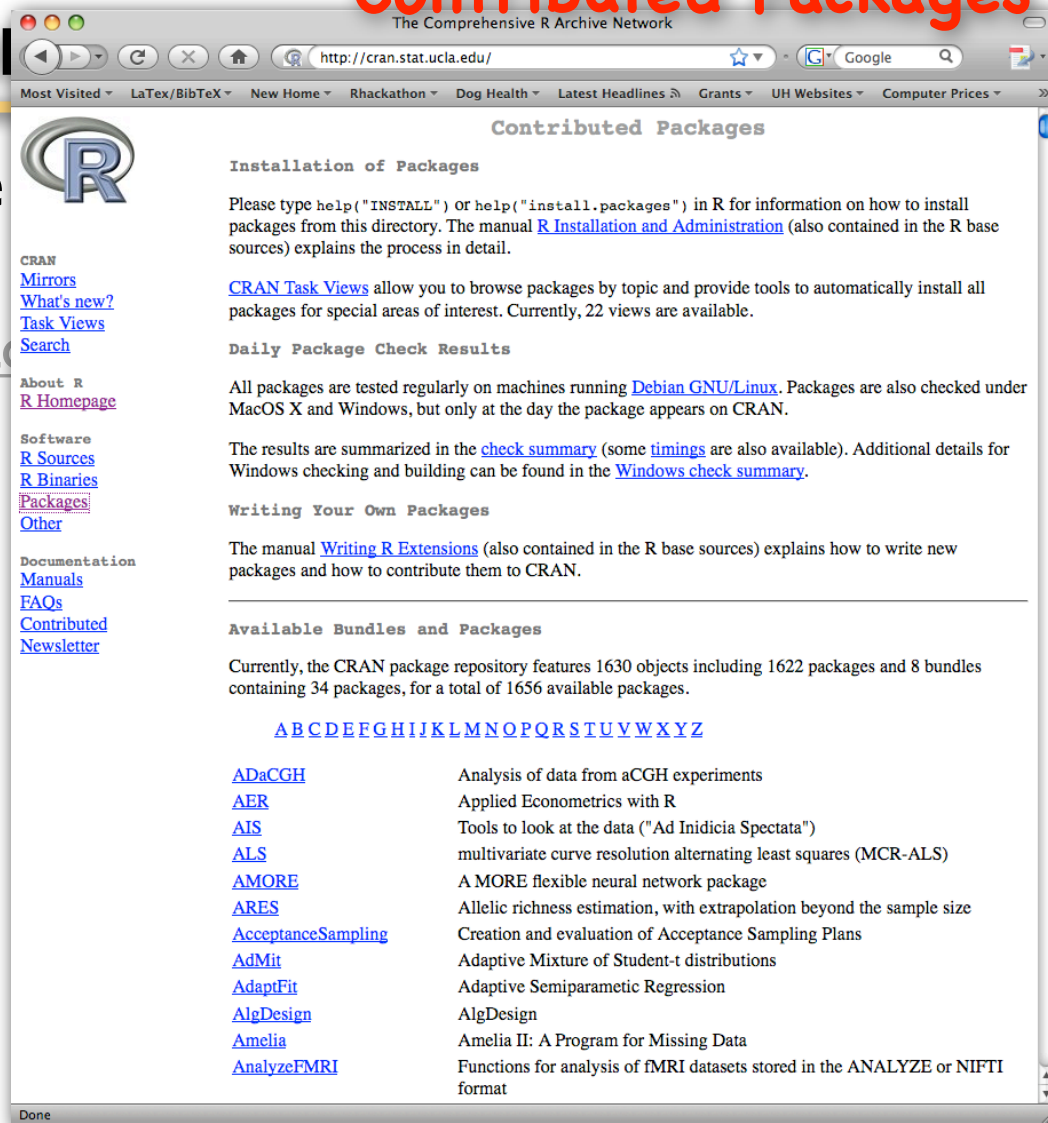
Statistics

Graphics

Programming

Open-Source Community

Contributed Packages



The screenshot shows the CRAN website in a web browser. The browser's address bar displays 'http://cran.stat.ucla.edu/'. The page title is 'Contributed Packages' and the subtitle is 'The Comprehensive R Archive Network'. The left sidebar contains a navigation menu with links: CRAN, Mirrors, What's new?, Task Views, Search, About R, R Homepage, Software, R Sources, R Binaries, Packages, Other, Documentation, Manuals, FAQs, Contributed, and Newsletter. The main content area is titled 'Contributed Packages' and includes sections for 'Installation of Packages', 'CRAN Task Views', 'Daily Package Check Results', 'Writing Your Own Packages', and 'Available Bundles and Packages'. The 'Available Bundles and Packages' section lists 1622 packages and 8 bundles, with a table showing the first few entries.

Package Name	Description
ADaCGH	Analysis of data from aCGH experiments
AER	Applied Econometrics with R
AIS	Tools to look at the data ("Ad Inidicia Spectata")
ALS	multivariate curve resolution alternating least squares (MCR-ALS)
AMORE	A MORE flexible neural network package
ARES	Allelic richness estimation, with extrapolation beyond the sample size
AcceptanceSampling	Creation and evaluation of Acceptance Sampling Plans
AdMit	Adaptive Mixture of Student-t distributions
AdaptFit	Adaptive Semiparametric Regression
AlgDesign	AlgDesign
Amelia	Amelia II: A Program for Missing Data
AnalyzefMRI	Functions for analysis of fMRI datasets stored in the ANALYZE or NIFTI format

the R environment

CRAN Package "home pages"

an integrated suite

Data Handling and S

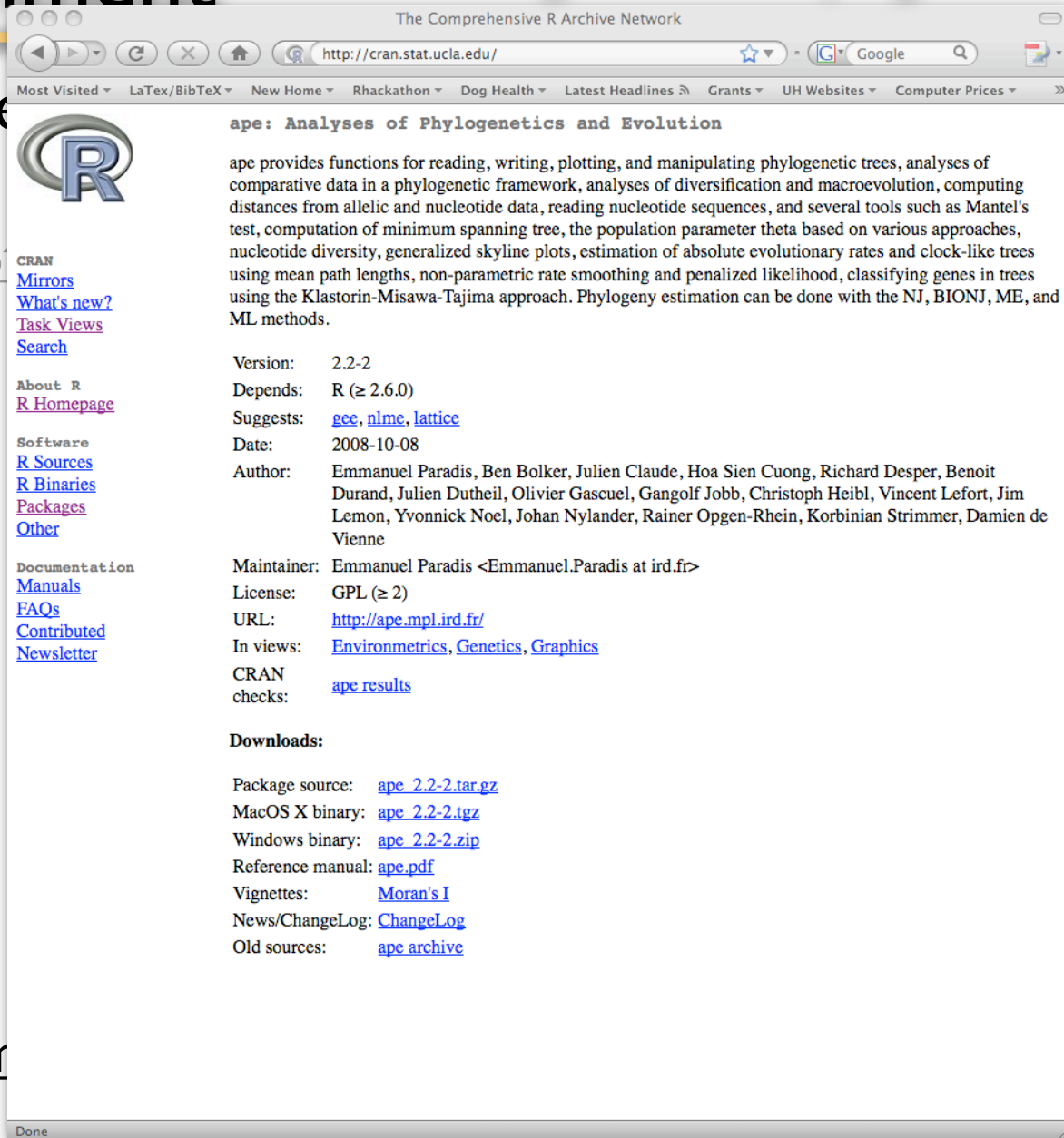
Matrix Math

Statistics

Graphics

Programming

Open-Source Comm



The screenshot shows a web browser window displaying the CRAN package page for 'ape: Analyses of Phylogenetics and Evolution'. The browser's address bar shows the URL 'http://cran.stat.ucla.edu/'. The page features the R logo on the left and a detailed description of the package on the right. The description states that 'ape' provides functions for reading, writing, plotting, and manipulating phylogenetic trees, among other tasks. It lists the version as 2.2-2, dependencies on R (≥ 2.6.0), and suggests other packages like 'gee', 'nlme', and 'lattice'. The author list includes Emmanuel Paradis, Ben Bolker, Julien Claude, Hoa Sien Cuong, Richard Desper, Benoit Durand, Julien Duthell, Olivier Gascuel, Gangolf Jobb, Christoph Heibl, Vincent Lefort, Jim Lemon, Yvonnick Noel, Johan Nylander, Rainer Opgen-Rhein, Korbinian Strimmer, and Damien de Vienne. The maintainer is Emmanuel Paradis. The license is GPL (≥ 2). The URL is 'http://ape.mpl.ird.fr/'. The package is available in various formats, including source, MacOS X binary, and Windows binary. The reference manual is 'ape.pdf'. Vignettes include 'Moran's I'. News/ChangeLog is available at 'ChangeLog'. Old sources are at 'ape archive'. The page also includes links for CRAN mirrors, what's new, task views, search, about R, R homepage, software, R sources, R binaries, packages, other, documentation, manuals, FAQs, contributed, and newsletter.

ape: Analyses of Phylogenetics and Evolution

ape provides functions for reading, writing, plotting, and manipulating phylogenetic trees, analyses of comparative data in a phylogenetic framework, analyses of diversification and macroevolution, computing distances from allelic and nucleotide data, reading nucleotide sequences, and several tools such as Mantel's test, computation of minimum spanning tree, the population parameter theta based on various approaches, nucleotide diversity, generalized skyline plots, estimation of absolute evolutionary rates and clock-like trees using mean path lengths, non-parametric rate smoothing and penalized likelihood, classifying genes in trees using the Klastorin-Misawa-Tajima approach. Phylogeny estimation can be done with the NJ, BIONJ, ME, and ML methods.

Version: 2.2-2
Depends: R (≥ 2.6.0)
Suggests: [gee](#), [nlme](#), [lattice](#)
Date: 2008-10-08
Author: Emmanuel Paradis, Ben Bolker, Julien Claude, Hoa Sien Cuong, Richard Desper, Benoit Durand, Julien Duthell, Olivier Gascuel, Gangolf Jobb, Christoph Heibl, Vincent Lefort, Jim Lemon, Yvonnick Noel, Johan Nylander, Rainer Opgen-Rhein, Korbinian Strimmer, Damien de Vienne
Maintainer: Emmanuel Paradis <Emmanuel.Paradis at ird.fr>
License: GPL (≥ 2)
URL: <http://ape.mpl.ird.fr/>
In views: [Environmetrics](#), [Genetics](#), [Graphics](#)
CRAN checks: [ape results](#)

Downloads:

Package source: [ape 2.2-2.tar.gz](#)
MacOS X binary: [ape 2.2-2.tgz](#)
Windows binary: [ape 2.2-2.zip](#)
Reference manual: [ape.pdf](#)
Vignettes: [Moran's I](#)
News/ChangeLog: [ChangeLog](#)
Old sources: [ape archive](#)

the R environment

an integrated suite of s

Data Handling and Storage

Matrix Math

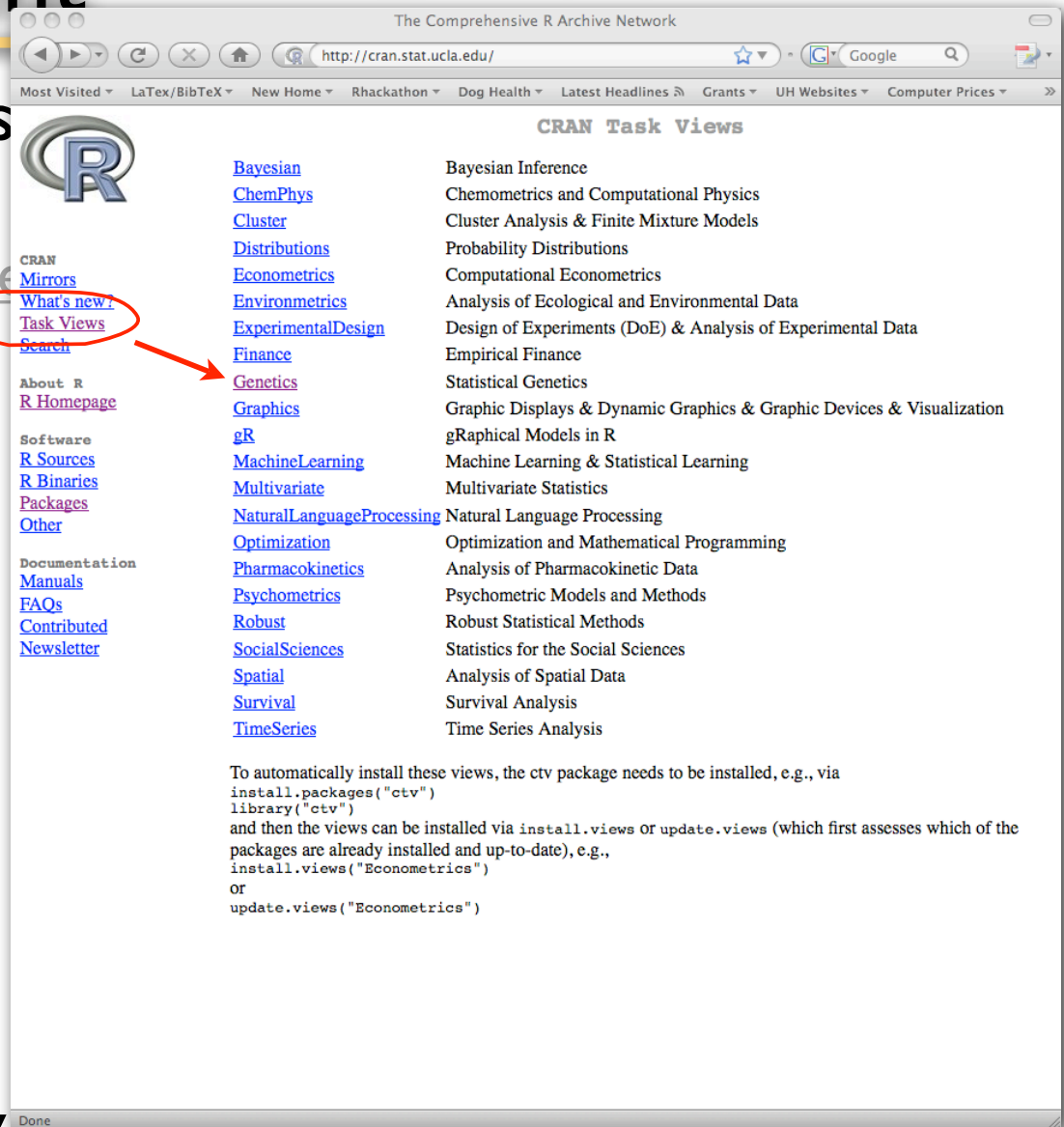
Statistics

Graphics

Programming

Open-Source Community

Task Views



The screenshot shows a web browser window titled "The Comprehensive R Archive Network" with the URL "http://cran.stat.ucla.edu/". The page displays the "CRAN Task Views" section, which lists various statistical and computational topics. A red circle highlights the "Task Views" link in the left sidebar, with a red arrow pointing to the "Task Views" link in the main content area. The main content area lists the following topics:

- Bayesian
- Bayesian Inference
- ChemPhys
- Chemometrics and Computational Physics
- Cluster
- Cluster Analysis & Finite Mixture Models
- Distributions
- Probability Distributions
- Econometrics
- Computational Econometrics
- Environmental
- Analysis of Ecological and Environmental Data
- ExperimentalDesign
- Design of Experiments (DoE) & Analysis of Experimental Data
- Finance
- Empirical Finance
- Genetics
- Statistical Genetics
- Graphics
- Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
- gR
- gRaphical Models in R
- MachineLearning
- Machine Learning & Statistical Learning
- Multivariate
- Multivariate Statistics
- NaturalLanguageProcessing
- Natural Language Processing
- Optimization
- Optimization and Mathematical Programming
- Pharmacokinetics
- Analysis of Pharmacokinetic Data
- Psychometrics
- Psychometric Models and Methods
- Robust
- Robust Statistical Methods
- SocialSciences
- Statistics for the Social Sciences
- Spatial
- Analysis of Spatial Data
- Survival
- Survival Analysis
- TimeSeries
- Time Series Analysis

Below the list, there is a section titled "To automatically install these views, the ctv package needs to be installed, e.g., via" followed by the following R code:

```
install.packages("ctv")
library("ctv")
and then the views can be installed via install.views or update.views (which first assesses which of the
packages are already installed and up-to-date), e.g.,
install.views("Econometrics")
or
update.views("Econometrics")
```


the R environment

an integrated suite of software

Data Handling and Storage

Matrix Math

Statistics

Graphics

Programming

Open-Source Community

Task Views



The Comprehensive R Archive Network

http://cran.stat.ucla.edu/

Most Visited ▾ LaTeX/BibTeX ▾ New Home ▾ Rhackathon ▾ Dog Health ▾ Latest Headlines ▾ Grants ▾ UH Websites ▾ Computer Prices ▾ ▸

CRAN Task View: Statistical Genetics

Maintainer: Giovanni Montana
Contact: g.montana at imperial.ac.uk
Version: 2008-12-08

Great advances have been made in the field of genetic analysis over the last years. The availability of millions of single nucleotide polymorphisms (SNPs) in widely available databases, coupled with major advances in SNP genotyping technology that reduce costs and increase throughput, are enabling a host of studies aimed at elucidating the genetic basis of complex disease. The focus in this task view is on R packages implementing statistical methods and algorithms for the analysis of genetic data and for related population genetics studies.

A number of R packages are already available and many more are most likely to be developed in the near future. Please send your comments and suggestions to the task view maintainer.

- **Population Genetics** : [genetics](#) implements classes and methods for representing genotype and haplotype data, and has several functions for population genetic analysis (e.g. functions for estimation and testing of Hardy-Weinberg and linkage disequilibrium, etc.). [Geneland](#) has functions for detecting spatial structures from genetic data within a Bayesian framework via MCMC estimation. [rmetasim](#) provides an interface to the metasim engine for population genetics simulations. [hapsim](#) simulates haplotype data with pre-specified allele frequencies and LD patterns. A few population genetics functions are also implemented in [gap](#) and [popgen](#). [popgen](#) has functions for clustering SNP genotype data and SNP simulation from a Multinomial-Dirichlet model. [hierfstat](#) allows the estimation of hierarchical F-statistics from haploid or diploid genetic data. [LDheatmap](#) creates a heat map plot of measures of pairwise LD. [mapLD](#) measures linkage disequilibrium and constructs haplotype blocks. [hwde](#) fits models for genotypic disequilibrium. Whilst [HardyWeinberg](#) provides graphical representation of disequilibrium via ternary plots (also known as de Finetti diagrams). [Biodem](#) package provides functions for Biodemographical analysis, e.g. `Fst()` calculates the Fst from the conditional kinship matrix. Package [kinship](#) offers some functions for analysis on pedigrees. The [adegenet](#) implements a number of different methods for analysing population structure using multivariate statistics, graphics and spatial statistics.
- **Phylogenetics** : Phylogenetic and evolution analyses can be performed via [ape](#) and [apTreeshape](#). Package [ouch](#) provides Ornstein-Uhlenbeck models for phylogenetic comparative hypotheses. [PHYLOGR](#) is a suite of functions for the analysis of phylogenetically simulated data sets and phylogenetically-based GLS model fitting. [stepwise](#) implements a method for stepwise detection of recombination breakpoints in sequence alignments. [phangor](#) estimates phylogenetic trees and networks using maximum likelihood, maximum parsimony, distance methods and Hadamard conjugation.
- **Linkage** : There are few native packages for performing parametric or non-parametric linkage analysis from within R itself, the calculations must be performed using external packages. However, there are a number of ancillary R packages that facilitate interface with these stand-alone programs and using the results for further analysis and presentation. [ibdreg](#) uses Identity By Descent (IBD) Non-Parametric Linkage (NPL) statistics for related pairs calculated externally to test for genetic linkage with covariates by regression modelling. [multic](#) also utilises IBD sharing statistics calculated externally for

Done

the R environment

Free Manuals on CRAN!

an integrated suite of



Data Handling and Storage

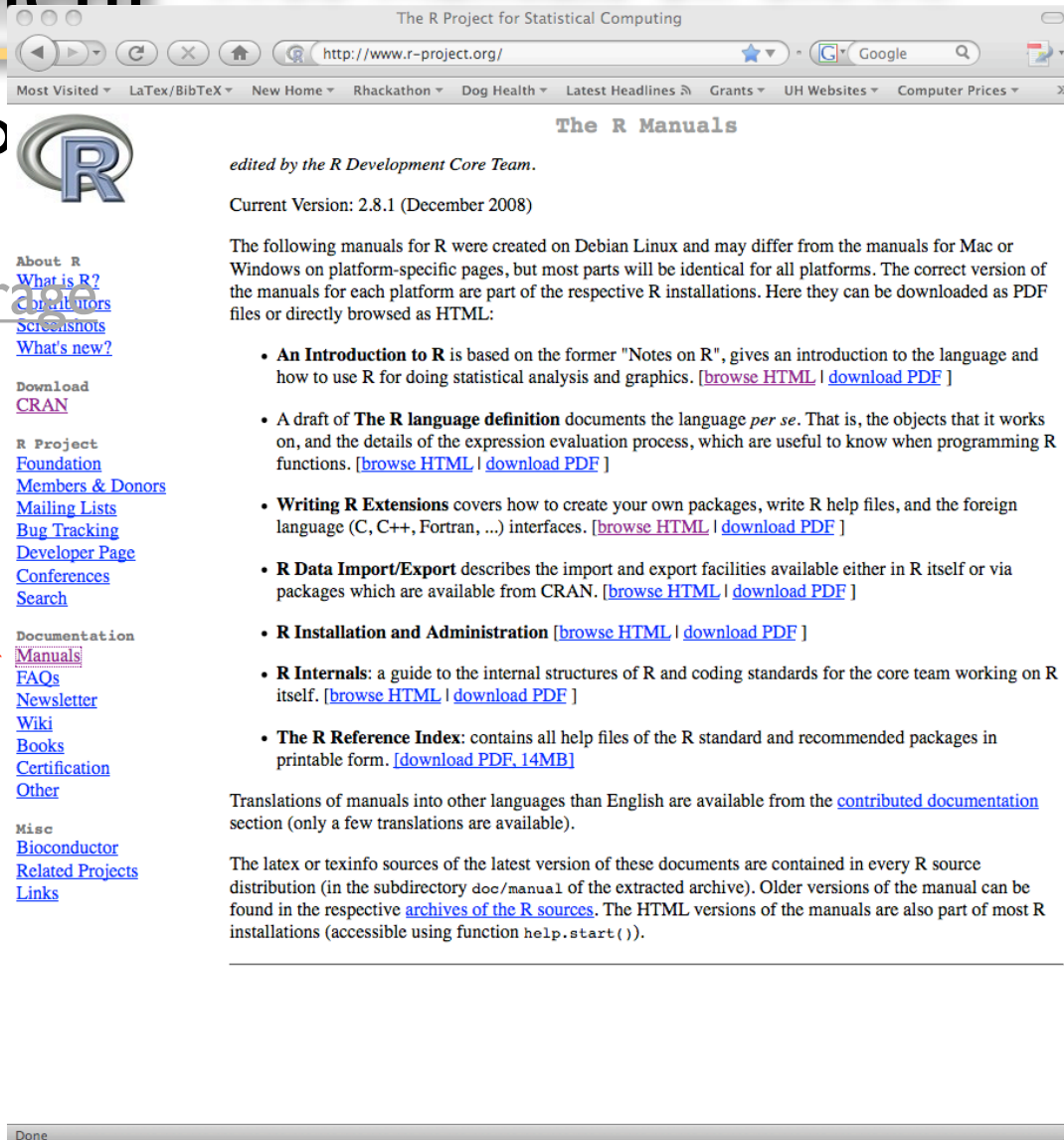
Matrix Math

Statistics

Graphics

Programming

Open-Source Community



The R Project for Statistical Computing

http://www.r-project.org/

Most Visited ▾ LaTeX/BibTeX ▾ New Home ▾ Rhackathon ▾ Dog Health ▾ Latest Headlines ▾ Grants ▾ UH Websites ▾ Computer Prices ▾ >>

The R Manuals

edited by the R Development Core Team.

Current Version: 2.8.1 (December 2008)

The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms. The correct version of the manuals for each platform are part of the respective R installations. Here they can be downloaded as PDF files or directly browsed as HTML:

- **An Introduction to R** is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics. [[browse HTML](#) | [download PDF](#)]
- A draft of **The R language definition** documents the language *per se*. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions. [[browse HTML](#) | [download PDF](#)]
- **Writing R Extensions** covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces. [[browse HTML](#) | [download PDF](#)]
- **R Data Import/Export** describes the import and export facilities available either in R itself or via packages which are available from CRAN. [[browse HTML](#) | [download PDF](#)]
- **R Installation and Administration** [[browse HTML](#) | [download PDF](#)]
- **R Internals**: a guide to the internal structures of R and coding standards for the core team working on R itself. [[browse HTML](#) | [download PDF](#)]
- **The R Reference Index**: contains all help files of the R standard and recommended packages in printable form. [[download PDF, 14MB](#)]

Translations of manuals into other languages than English are available from the [contributed documentation](#) section (only a few translations are available).

The latex or texinfo sources of the latest version of these documents are contained in every R source distribution (in the subdirectory `doc/manual` of the extracted archive). Older versions of the manual can be found in the respective [archives of the R sources](#). The HTML versions of the manuals are also part of most R installations (accessible using function `help.start()`).

Done

Let's Play with  !

How does R store data?

R can **save** information in variables or objects

Assignment works by two types of operators:

Equal sign: right side stored in left side

> x = 6 (put 6 into x)

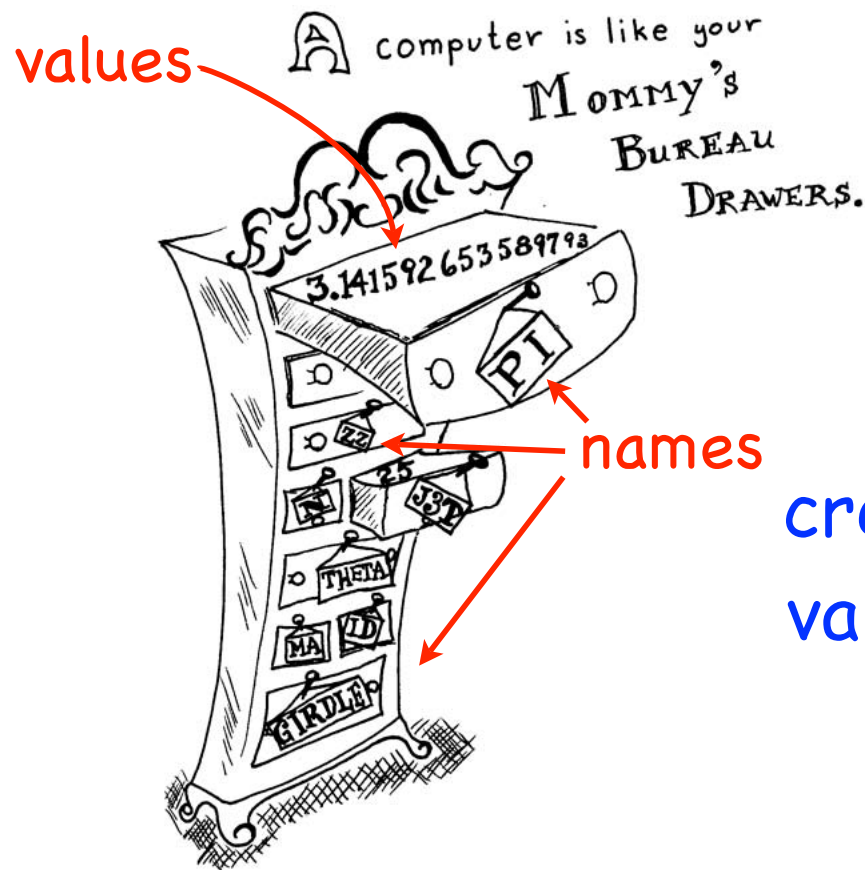
Arrow: assignment direction follows arrow

> x <- 6 (put 6 into x)

> 6 -> x (put 6 into x)

> 6 = x (error! cannot put x into 6)

How does R store data?



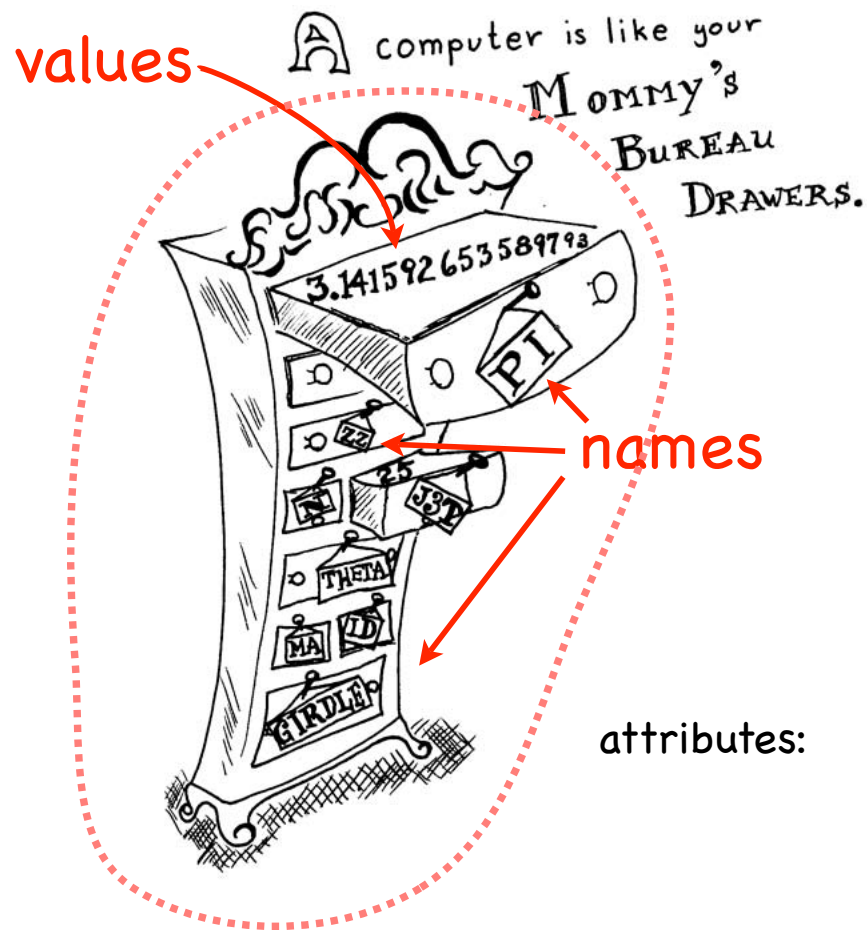
In R, there are drawers for:

- Numbers
- Characters (alphabetical strings)
- Logical (TRUE or FALSE)
- Complex Numbers (don't worry)

Create a new drawer by creating a name, and shoving a value into it - R will assign the "type" or "mode"

```
> x <- 25  
name      value  
  
> mode(x)  
[1] "numeric"
```

How does R store data?



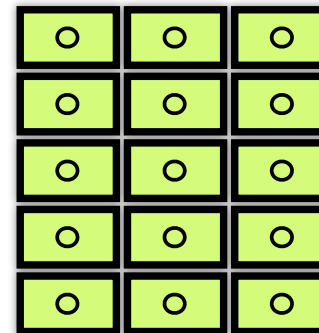
Bureaus can come in
different shapes

Vector



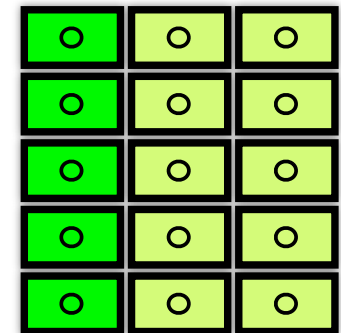
attributes: length

Matrix



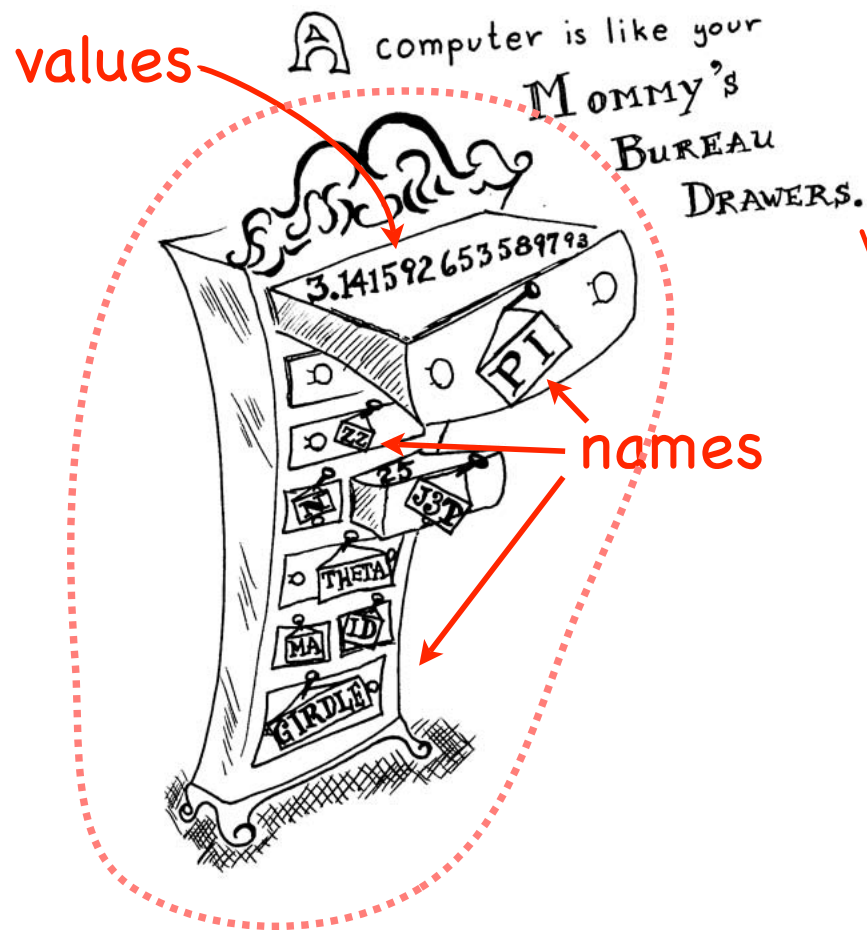
dimensions(5 rows, 3 columns)

Data Frame



Rectangular!
all columns have same length

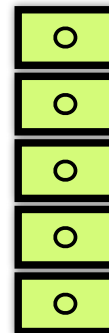
How does R store data?



Object

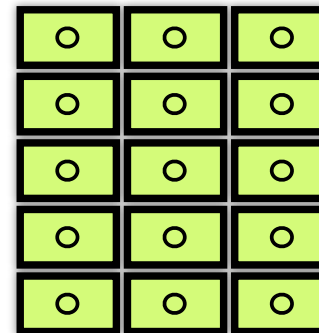
Bureaus can come in
different shapes

Vector

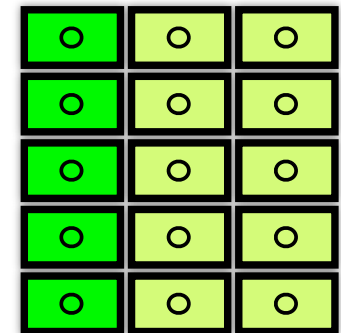


↑
All values same type
numeric vector
or
character matrix
etc.

Matrix

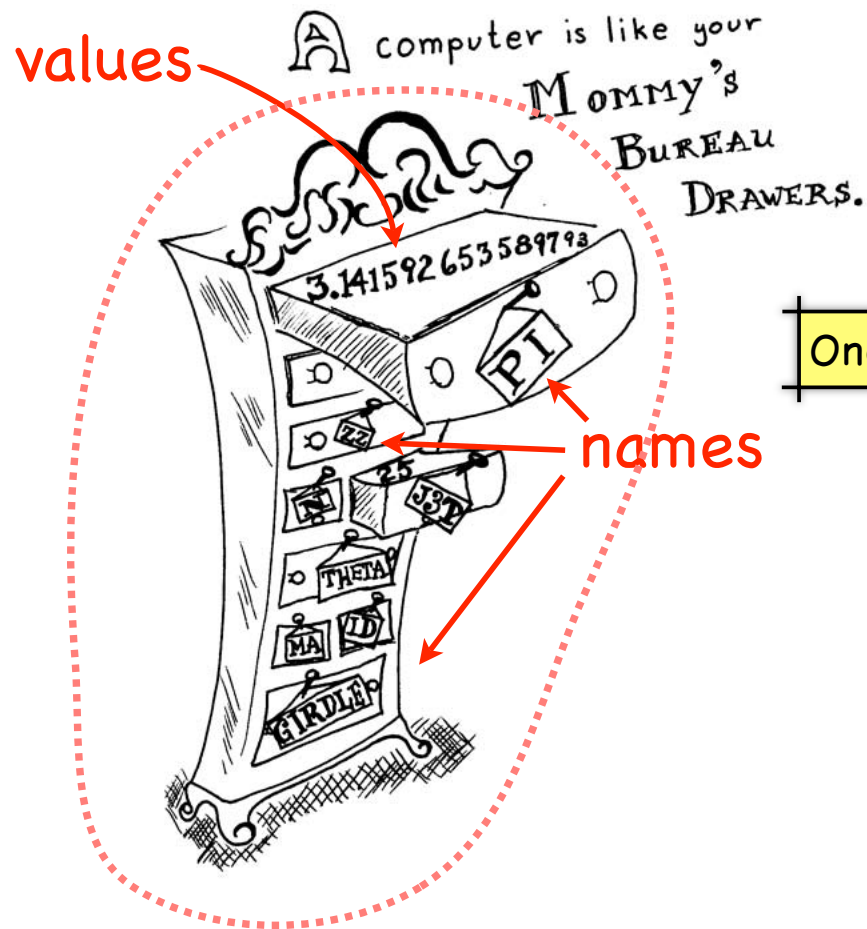


"record format"
Data Frame



↑ ↙ ↘
character numeric
e.g. species size, mass

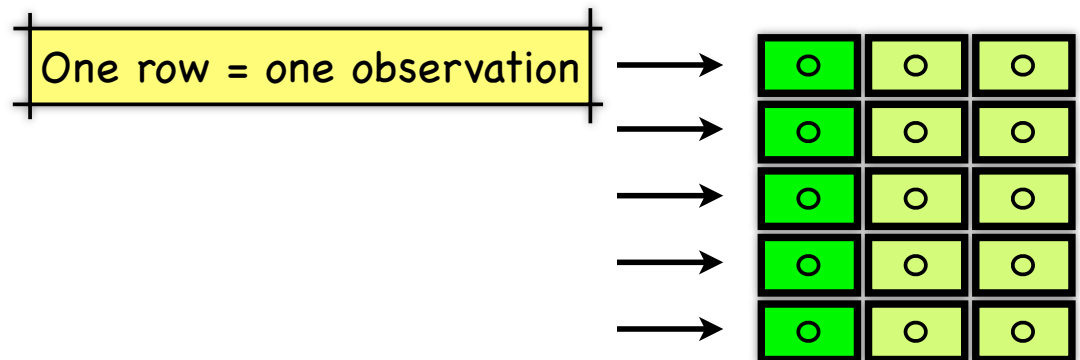
How does R store data?



Object

Bureaus can come in
different shapes

"record format"
Data Frame



For example: species name size mass
character vector numeric vector numeric vector

values in each vector
same type,
but vectors can be
different types

How does R store data?

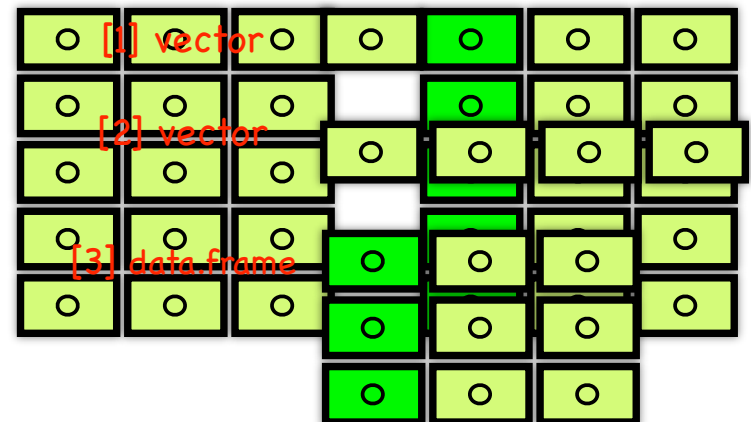
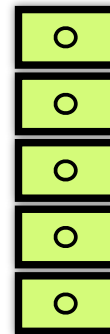
But all of these are in
different shapes
classes and objects

"record format"

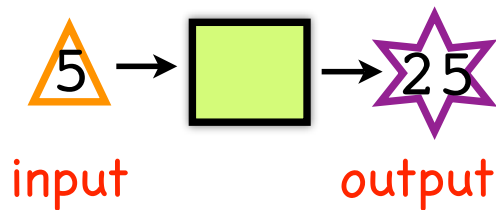
Vector

Matrix

List Data Frame



Functions



- a "list" of objects
- offers more flexibility
- often used for model output
- R has many functions that operate on lists

Common Sources of Error

1) Typos! Computers are very anal that way.

- > length = 6 # is not the same as
- > lengths = 6

2) R is case sensitive

- > length != Length

3) Using () when should use [] and vice versa

- > mean(x) # use () for functions
- > mean[x] # error
- > x[5] # select an element of a vector, matrix, data.frame, etc.
- > x(5) # error

4) No comma or comma in the wrong place

- > x[5,3] # fifth row, third column of x
- > x[5 3] # error
- > x[5,3,] # error

Common Sources of Error

5) Forgetting quotes for character strings (R will assume it's another named object or variable)

```
> treatment = c("a", "b", "c")
```

```
> treatment == a      # error - R thinks a is another object
```

What is it?

check using `class()` and `attributes()`

All objects have **class**, **mode** and **length** (the number of vector elements; everything is a vector in R)

names are optional

Derived objects can have **attributes** such as:

dim (dimensions -- number of rows, columns)

row.names

(other programmer-defined attributes)

Functions and Operators perform actions

Functions

function name
arguments

```
> c(1, 2, 3) -> x
```

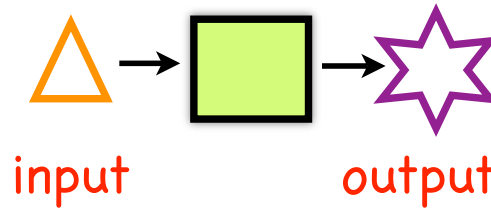
combine 1,2,3 into a vector, save as x

```
> mean(x) # mean of x
```

```
[1] 2
```

```
> ls()
```

Functions **ALWAYS** have ()
arguments are sometimes optional



Operators are used for symbolic math

(otherwise very similar
to functions)

```
> 2*2 # * multiplication symbol
```

```
> 2^2 # ^ power (2 squared)
```

```
> x == 2 # x equals 2 (element by element)
```

```
> x < 2 # x less than 2
```

R works on Objects

What is an object? An *abstraction*

- * data vector, matrix, array, data frame, list, or function
- * a definition for a type of data to be stored (object class)
- * ways of bundling parts of programs into small, manageable pieces (functions or methods)
- * an object is a component of a program that knows how to perform certain actions and to interact with other pieces of the program.
- * Functions can be described as "black boxes" that take an input and spit out an output. Objects can be thought of as "smart" black boxes. That is, objects can know how to do more than one specific task (method or behavior), and they can store their own set of data.

Objects should mimic behavior of real-world counterparts

All objects have class, mode and length

And possibly other attributes

When working with a new package,
you want to know what kind of object you are dealing with:

A common source of error is putting the **wrong class** of
object into a function

Huh?

“Atomic” datatypes basic types for the single element. (can’t break it down any further).

mode : the atomic data type (**class = mode** if object is atomic)

class : classes can be derived (i.e., programmer defined) or atomic
-- also refers to the classes that the object inherits (i.e., which
methods used are determined by the class of object)

R object types

Vector: a one-dimensional array of arbitrary length.

Matrix: a two-dimensional array with an arbitrary number of rows and columns.

Array: as a matrix, but of **arbitrary dimension** (i.e., more than 2).

Data frame: a set of data organized similarly to a matrix. However each column of the data frame may contain its own type of data. Columns typically correspond to variables in a statistical study, while rows correspond to observations of these variables.

Function: a set of commands that are packaged into a unit with defined input and output (I/O is not necessary, though).

List: an arbitrary collection of other R objects (which may include other lists).

```
> x <- cbind(a=1:3, pi=pi) # simple
                                # matrix w/ dimnames
```

```
> x
      a      pi
[1,] 1 3.141593
[2,] 2 3.141593
[3,] 3 3.141593
```

```
> class(x)
[1] "matrix"
```

```
> attributes(x)
$dim
[1] 3 2

$dimnames
$dimnames[[1]]
NULL
```

```
$dimnames[[2]]
[1] "a" "pi"
```

```
> attributes(x) <- NULL
> x # now just a vector of length 6
```

```
[1] 1.000000 2.000000 3.000000 3.141593
     3.141593 3.141593
```

```
> class(x)      # vector is default mode
[1] "numeric"
```


the R environment : Session

SESSION: A “session” is a single use of the R language, the period between starting R up and shutting it down.

Within a session, you may load packages, create R objects, produce graphics, or write/run scripts.

WORKSPACE: The collection of objects currently stored.

```
> ls()    # display the names of objects in workspace  
> rm(list=ls()) # deletes all objects from workspace
```

the R environment : Session

At end of SESSION:

Save workspace image? [y/n/c]:

If you say “y”, the objects are written to a file called “.RData” in the current directory, and the command lines used in the session are saved to a file called “.Rhistory”.

NOTE: Generally, files that start with a “.” are hidden in the directory.

IF you restart R from the same directory, it will reload the default (.Rhistory, .Rdata) history and workspace files.

use `save()`, `save.image()`, `savehistory()` to save these files with your own names specified within the parentheses.

the R environment : Packages

The R programming language is written in **modules** called **packages**, which are groups of related functions organized together in a bundle.

Packages are the means by which R is extended by the open-source community (user contributed packages)

Not all aspects of the R programming language are **loaded** every time you fire R up. -- you have to load optional packages with every new session.

the R environment : Packages

default packages at startup are **base** & a few others.

```
> search()    # gives search path for R objects
```

```
[1] ".GlobalEnv" "tools:RGUI" "package:methods" "package:stats"  
[5] "package:graphics" "package:grDevices" "package:utils"  
"package:datasets"  
[9] "Autoloads" "package:base"
```

When you type a command or object name in R, it “searches” through the “search path” for a match and then takes appropriate action (be it the name of a data object, function, operator, etc.).

NOTE: Don't create objects with the same name as R commands! (e.g., t, T, F, c all are special characters) Results are unpredictable

the R environment : Packages

```
> searchpaths() # gives path to package source code on your computer's file system #
[1] ".GlobalEnv"
[2] "tools:RGUI"
[3] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/methods"
[4] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/stats"
[5] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/graphics"
[6] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/grDevices"
[7] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/utils"
[8] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/datasets"
[9] "Autoloads"
[10] "/Library/Frameworks/R.framework/Versions/2.4/Resources/library/base"
```

INSTALLING Packages: saves the package source code to the appropriate place in your computer's file directory (usually a place that users don't mess with)

LOADING Packages: attaches the package from your computer's R library to your search path (adds it to your session).

```
> library("packagename")
```

the R environment : attach()

ATTACH: Attach Set of R Objects to Search Path

The database is attached to the R search path. This means that the database is searched by R when evaluating a variable, so objects in the database can be accessed by simply giving their names.

```
> data.frame(xx=1:10,yy=runif(10)) -> ddata
> attach(ddata)
> search( )
[1] ".GlobalEnv"  "ddata"       "tools:RGUI"  "package:methods"
[5] "package:stats" "package:graphics" "package:grDevices"
"package:utils"
[9] "package:datasets" "Autoloads"  "package:base"
```

Now ddata is in the search path, so R knows what you mean when you specify “xx” or “yy” without the data.frame.

the R environment : Working Directory

R program software is stored in a “safe” location on your computer. (usually you never want to touch this. In fact, you may not even know where this is – that's OK).

R working directory is the location where you keep your own R scripts, input files, and output files.

 **BEST PRACTICES:** Keep your software directories pristine!

Keep data directories separate.

This simplifies maintenance, prevents corruption of important software, and simplifies backup.

the R environment : Working Directory

```
> getwd()    # gets the current working directory  
[1] "C:/Program Files/R/R-2.4.0"
```

Don't use this one! -- it will dump junk into your R (program file) directory and potentially interfere with running R!! **Be especially careful on PCs!!!**

```
> setwd("/Documents and Settings/jeffrey Scales/My Documents/Data/  
Rgroup")
```

Create one instead (first you have to make a new folder called “Data” in your “My Documents” folder, and any other sub-folders you want), then use setwd() to point R to your preferred working directory

NOTE: Tinn-R is a free text editor (GPL) designed for programming in R
<http://www.sciviews.org/Tinn-R/>

References

- ☒ Jonathan Baron's R help Page <http://finzi.psych.upenn.edu/page>
- ☒ An Introduction to R <http://cran.r-project.org/doc/manuals/R-intro.pdf>
- ☒ "An Introduction to R: Software for Statistical Modelling & Computing" by Petra Kuhnert and Bill Venables http://cran.r-project.org/doc/contrib/Kuhnert+Venables-R_Course_Notes.zip
- ☒ Programming in R (a great web page!) by Vincent Zoonekynd http://zoonek2.free.fr/UNIX/48_R/02.html
- ☒ Programming in R (with R & BioConductor) by Thomas Girke http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/R_Programming.html
- ☐ (Generally, see the R manuals at <http://cran.r-project.org/manuals.html>)
- ☐ Online C++ tutorial: What is an object? http://www.intap.net/~drw/cpp/cpp06_01.htm
- ☐ The JAVA tutorials: What is an object? <http://java.sun.com/docs/books/tutorial/java/concepts/object.html>