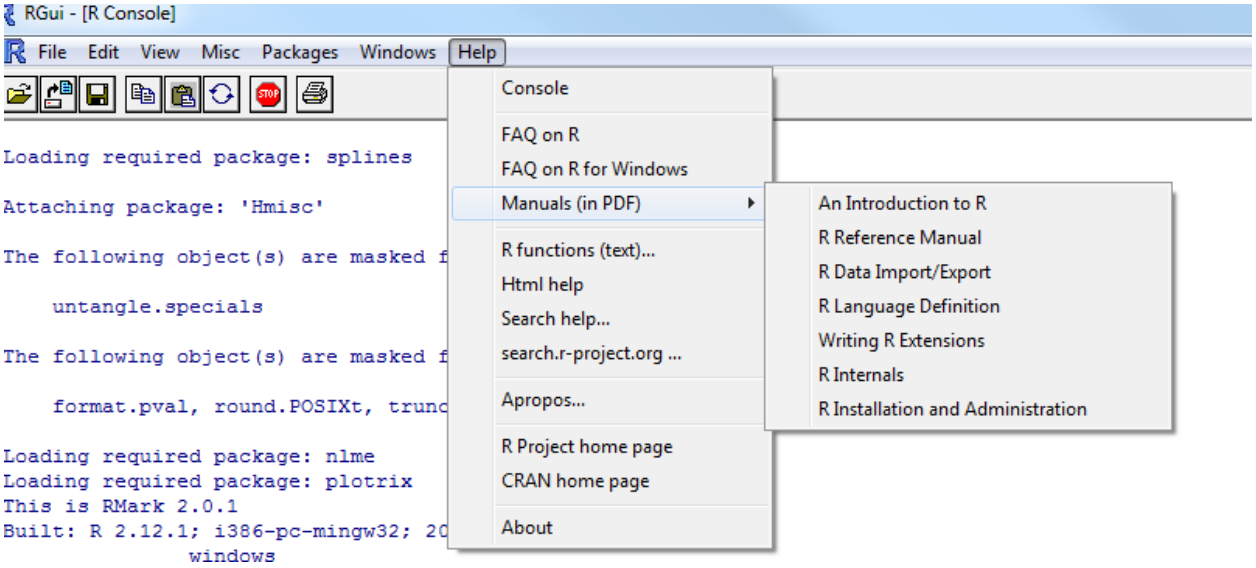
How can I learn R?

Initially it may seem like an impossible task to learn R. Fortunately it is not necessary or even possible to learn everything in R before you start to enjoy its benefits. I am continually learning new things in R and relearning things that I once knew. Because it is such a diverse environment with an enormous number of functions and arguments it is impossible (at least for me) to maintain it all in your brain. To learn and re-learn R, it is essential that you integrate that learning process into your workflow. As a new user to R you may not be familiar with many of the ways to learn R and stay up to date on new developments. Here is what I use.

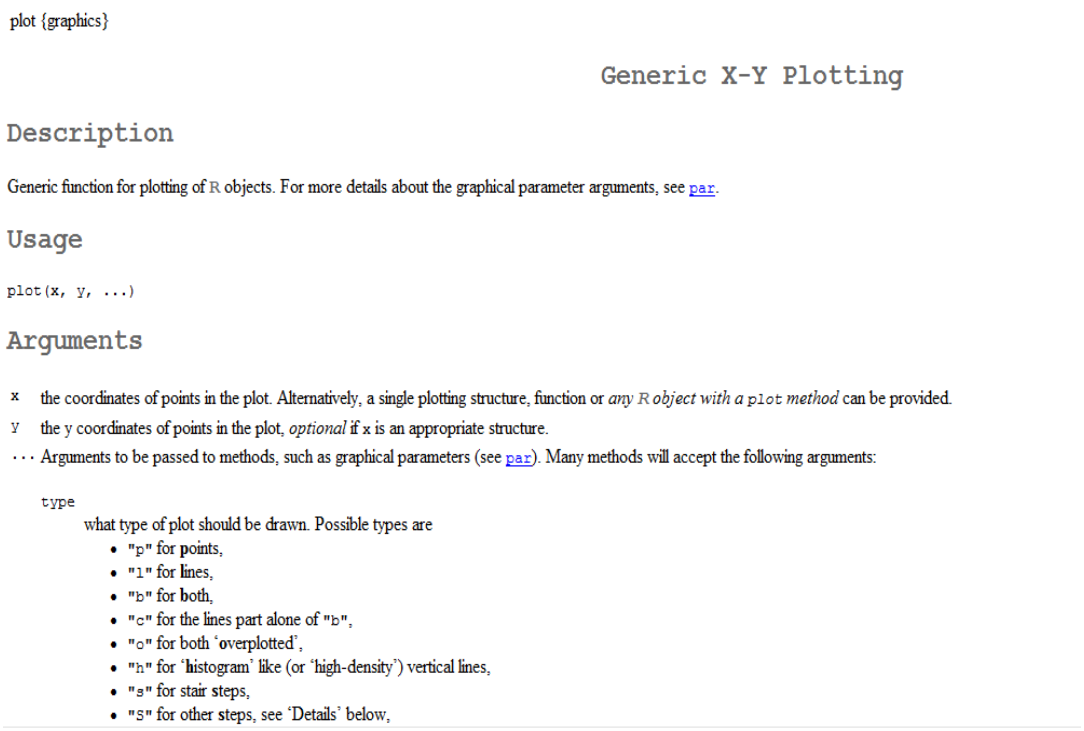
1) Books: If you are brand new user buy a book on R. There are many out there so it is hard to recommend a particular book. A good beginner book Introductory Statistics with R was written by Peter Dalgaard but it is rather brief and focuses mostly on "doing" statistics with R. More complete books include the R Book, R in a Nutshell, and R Cookbook. If you are primarily interested in programming and data manipulation then I'd recommend Data Manipulation with R by Spector. Books will get you a start but by their very nature they are limited in their coverage. Also, books are static and R is very dynamic with at least 2 new versions per year and many new packages being added all the time.
2) R Manuals: Depending on the options you chose, one or more pdf manuals are installed with R. You can find them on the help menu as shown below:



If a manual is grayed out then it was not installed. I routinely select all manuals to be installed. The Intro to R is a good starting point if you don't want to buy a book. The R Reference manual is a complete reference manual of the functions contained in base R and the standard packages that accompany base R like stats, graphics, tools etc. The R Data Import/Export covers various to import and export data and convert to and from various formats. The R Language Definition describes the R language and syntax. A programmer will find this a useful reference to read. It can be rather daunting for the beginner but still very useful because it is important to understand the syntax and objects that compose the language. Paying attention to the details of the syntax is often the most troubling part of R for new users. If you learn it well, the rest of R

will come easier. The manual Writing R Extensions is useful if you want to start developing your own R packages or want to interface to C/FORTRAN code. The other 2 manuals are quite technical and only useful to technical programmers and system administrators.

3) Help: The most current documentation for R and all of the 1000s of packages is the documentation that accompanies them. All of this documentation has a common format which helps with learning but there are a few tricks that you should know. Help comes in 2 flavors: text and html. Awhile back there were also compiled help files on Windows which was a very useful format but Microsoft abandoned their help compiler so it was dropped by the R developers. I believe now the default for R help is html which you'll be used to from the internet with hyperlinks to related pages etc. There are several ways to get help.
    a. You can select the Help/Html help menu item as shown in figure above (equivalent to typing help.start()) and then navigate to the section of the help.
    b. If you know the name of the function, you can type ?function (e.g., ?strsplit) and it will show the help page. If the help does not appear in your browser window and there are no hyperlinks, then you are getting the text version. You can get the html version by typing options(help_type="html") at the start of the session or include in your .First or RProfile file. The structure of the help file is common to all help. The figures below show the help for plot in a couple of panels going from top to bottom.



plot {graphics}

Generic X-Y Plotting

Description

Generic function for plotting of R objects. For more details about the graphical parameter arguments, see par.

Usage

```
plot(x, y, ...)
```

Arguments

x	the coordinates of points in the plot. Alternatively, a single plotting structure, function or *any R object with a* `plot` *method* can be provided.
y	the y coordinates of points in the plot, *optional* if x is an appropriate structure.
...	Arguments to be passed to methods, such as graphical parameters (see par). Many methods will accept the following arguments:

`type`
what type of plot should be drawn. Possible types are
- `"p"` for points,
- `"l"` for lines,
- `"b"` for both,
- `"c"` for the lines part alone of `"b"`,
- `"o"` for both 'overplotted',
- `"h"` for 'histogram' like (or 'high-density') vertical lines,
- `"s"` for stair steps,
- `"S"` for other steps, see 'Details' below,

At the top left hand corner is the name of the function and in the {} is the package that contains the function. This is followed by a title, description and then the 2 most important sections Usage and Arguments. Usage shows the syntax for calling the function including the names and order of the arguments. The Arguments section provides the definitions of the arguments. A few non-obvious details are important here in the Usage section. If arguments are specified as x,y like above then they do not

have a default value and must be specified and if they are not an error will be given that a particular argument is missing. If the argument is specified as argument=value (e.g., na.rm= FALSE in ?mean) in the Usage section then the specified value is the default and you only need to specify that argument if you want to use something other than the default. Also, the … is special "argument" which is really a placeholder for one or more arguments that are passed through to other functions that are called by the current function. Below the … is a list of common arguments that are specified and how they can be used, but it is not complete. For example, … in plot can be used to specify arguments to the function par such as pch=2 to specify the use of the second plotting character for points. Notice the hyperlink to par in the help file which lets you easily jump to that help file. The remaining sections of help include, Details (more specific details about what the function does and how it works), References (if any), See Also (related functions with hyperlinks to their help pages), and Examples which provide example code.

Details

For simple scatter plots, `plot.default` will be used. However, there are `plot` methods for many R objects, including `function`s, `data.frame`s, `density` objects, etc. U

The two step types differ in their x-y preference: Going from *(x1,y1)* to *(x2,y2)* with $x1 < x2$, `type = "s"` moves first horizontal, then vertical, whereas `type = "S"` move

See Also

`plot.default`, `plot.formula` and other methods; `points`, `lines`, `par`.

For X-Y-Z plotting see `contour`, `persp` and `image`.

Examples

```
require(stats)
plot(cars)
lines(lowess(cars))

plot(sin, -pi, 2*pi)

## Discrete Distribution Plot:
plot(table(rpois(100,5)), type = "h", col = "red", lwd=10,
     main="rpois(100,lambda=5)")

## Simple quantiles/ECDF, see ecdf() {library(stats)} for a better one:
plot(x <- sort(rnorm(47)), type = "s", main = "plot(x, type = \"s\")")
points(x, cex = .5, col = "dark red")
```

[Package *graphics* version 2.12.1 Index]

You can copy and paste code from the example or more simply type example(function) at the command prompt (e.g., example(plot)) and the example code will be run in your session. Something that is often missed in the help is the Index hyperlink at the very bottom of the page. Clicking on it will take you to the index of all the functions in that package. Often that is a good way to see what other functions are in a package rather than navigating the related links in See Also.

c. If you don't know the name of a particular function then you can type help.search("text") or select Help/Search help on the menu shown in the figure above. to search the help files for a particular text string. Below is the first part of the list returned for help.search("plot"). On the left is the name of the package and the name of the function after the ::

```
Help files with alias or concept or title matching 'plot' using regular expression matching:


ade4::add.scatter                   Add graphics to an existing plot
ade4::area.plot                     Graphical Display of Areas
ade4::between                       Between-Class Analysis
ade4::betweencoinertia              Between-class coinertia analysis
ade4::coinertia                     Coinertia Analysis
ade4::corkdist                      Tests of randomization between distances applied to 'kdist' objetcs
ade4::discrimin                     Linear Discriminant Analysis (descriptive statistic)
ade4::dpcoa                         Double principal coordinate analysis
ade4::dudi.acm                      Multiple Correspondence Analysis
ade4::foucart                       K-tables Correspondence Analysis with the same rows and the same
ade4::fourthcorner                  Functions to compute the fourth-corner statistic
ade4::kplot                         Generic Function for Multiple Graphs in a K-tables Analysis
ade4::kplot.foucart                 Multiple Graphs for the Foucart's Correspondence Analysis
ade4::kplot.mcoa                    Multiple Graphs for a Multiple Co-inertia Analysis
ade4::kplot.mfa                     Multiple Graphs for a Multiple Factorial Analysis
ade4::kplot.pta                     Multiple Graphs for a Partial Triadic Analysis
ade4::kplot.sepan                   Multiple Graphs for Separated Analyses in a K-tables
ade4::kplot.statis                  Multiple Graphs of a STATIS Analysis
ade4::krandtest                     Class of the Permutation Tests (in C).
ade4::mcoa                          Multiple CO-inertia Analysis
ade4::mdpcoa                        Multiple Double Principal Coordinate Analysis
ade4::mfa                           Multiple Factorial Analysis
ade4::multispati                    Multivariate spatial analysis
ade4::niche                         Method to Analyse a pair of tables : Environmental and Faunistic Data
ade4::njplot                        Phylogeny and trait of bacteria
ade4::pcaiv                         Principal component analysis with respect to instrumental variables
```

At the bottom of the list it will show the message
Type '?PKG::FOO' to inspect entries 'PKG::FOO', or 'TYPE?PKG::FOO' for entries like 'PKG::FOO-TYPE'.

As an example, if I wanted help for area.plot in the package ade4, I would enter ?ade4::area.plot. To close that list, click on the x in the right hand corner of the window and not at the very top because that will close R.

d. There are various other search engines available to get help for R and its functions. One is available on the Help/search r-project.org menu item (also available at command line as RSiteSearch e.g. RSiteSearch("plot")). It will search the R help list archives (described below) and documentation for a particular text string that you enter. It will open the search results in your browser. Another useful search engine is at www.rseek.org.

4) R Help list: Sometimes you can't find the help you need or can't understand how something does or does not work. Once you have done your homework and can't work out the problem you should turn to the R help list. Go to http://www.r-project.org/ and click on the Mailing Lists link on the left side of the page. There you'll find information on the various mailing lists including R-help. Do what it says at the top of the page:

Please read the instructions below and the posting guide *before* sending anything to any mailing list!

If you don't, you may get a nasty message back if you don't follow the guidelines. The guidelines are there for your benefit. If you follow them you are more likely than not to get help that will actually be useful. RSiteSearch and RSeek search the archives of the R help list which is why that you want to search them first because there is a good chance that someone may have already asked your question or something like it. If someone replies with, "please provide self-contained reproducible code" what they mean is to provide in the body of your message a simple example that illustrates the problem you are having. Often in creating that simple example you'll discover the error. If you provide an example that someone can copy from your

message and paste into R, they can see the problem and help you. If you can't reproduce the error easily and don't understand an error message, provide the code that gave the error and the error message that appeared. Don't send a large data file or any attachments. Don't be afraid to post to R-help, just try to help yourself first and then follow the posting guide in creating your query. You can subscribe to R-help and get the messages individually or in digest form. I get them individually and have my email client direct them to a separate folder. Then I sort them by subject and briefly look through them. If I see a topic of interest then I read it, otherwise I delete the contents of the folder. There are about 100-200 messages a day so you do want it to come to your inbox except in digest form which is another option.

5) What package should I use? The 1000s of packages can be overwhelming and you don't know which one to use or what it contains. The first place to turn is Task Views which can be found at http://cran.stat.ucla.edu/web/views/ in the UCLA repository and it looks like:

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	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
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis

Each subject hyperlink takes you to material that describes the various packages that are available for that subject area. They are usually clustered based on the kind of task the packages accomplish and a brief description may be given for each package. For example, the beginning of the task view for Environmetrics is shown on the next page. It has numerous different categories and within each a brief description and a link to each relevant package. For example, under the section describing Ordination is a link to the package vegan.

CRAN Task View: Analysis of Ecological and Environmental Data

Maintainer: Gavin Simpson
Contact: gavin.simpson at ucl.ac.uk
Version: 2011-01-28

Introduction

This Task View contains information about using R to analyse ecological and environmental data.

The base version of R ships with a wide range of functions for use within the field of environmetrics. This functionality is complemented by a plethora of packages available via CRAN, which provide specialist methods such as ordination & cluster analysis techniques. A brief overview of the available packages is provided in this Task View, grouped by topic or type of analysis. As a testament to the popularity of R for the analysis of environmental and ecological data, a special volume of the *Journal of Statistical Software* was produced in 2007.

Those useRs interested in environmetrics should consult the Spatial view. Complementary information is also available in the Multivariate, Phylogenetics and Cluster task views.

If you have any comments or suggestions for additions or improvements, then please contact the maintainer.

A list of available packages and functions is presented below, grouped by analysis type.

Modelling species responses and other data

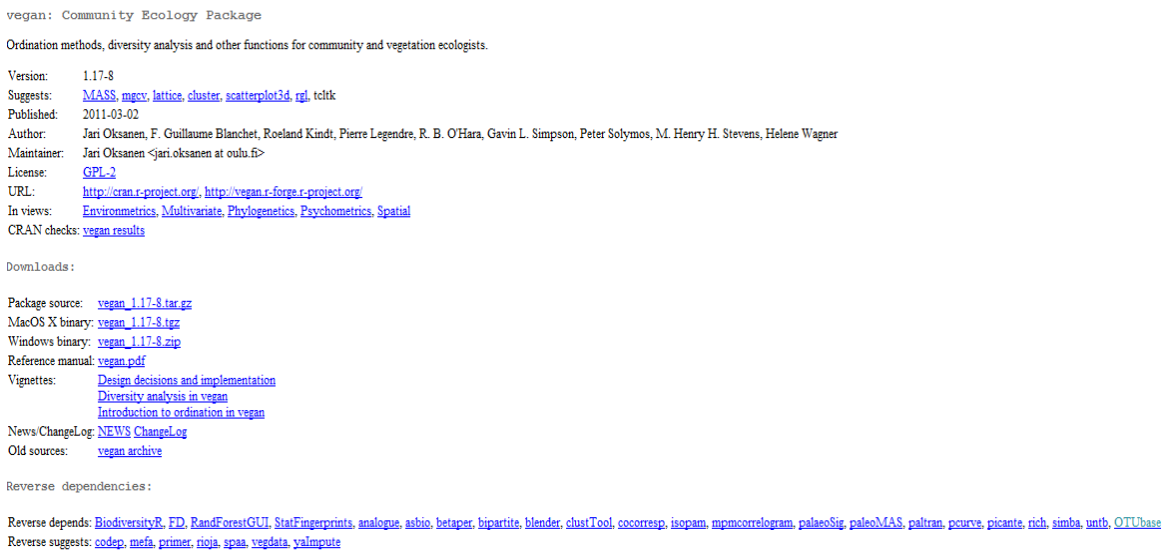
Analysing species response curves or modeling other data often involves the fitting of standard statistical models to ecological data and includes simple (multiple) regression, Generalised Linear Models (GLM), extended regression (e.g. Generalised Least Squares [GLS]), Generalised Additive Models (GAM), and mixed effects models, amongst others.

- The base installation of R provides `lm()` and `glm()` for fitting linear and generalised linear models, respectively.
- Generalised least squares and linear and non-linear mixed effects models extend the simple regression model to account for clustering, heterogeneity and correlations within the sample of observations. Package nlme provides functions for fitting these models. The package is supported by Pinheiro & Bates (2000) *Mixed-effects Models in S and S-PLUS* , Springer, New York. An updated approach to mixed effects models, which also fits Generalised Linear Mixed Models (GLMM) and Generalised non-Linear Mixed Models (GNLMM) is provided by the lme4 package, though this is currently beta software and does not yet allow correlations within the error structure.
- Recommended package mgcv fits GAMs and Generalised Additive Mixed Models (GAMM) with automatic smoothness selection via generalised cross-validation. The author of mgcv has also written a companion monograph, Wood (2006) *Generalized Additive Models; An Introduction with R* Chapman Hall/CRC, which has an accompanying package gamair.
- Alternatively, package gam provides an implementation of the S-PLUS function `gam()` that includes LOESS smooths.
- Proportional odds models for ordinal responses can be fitted using `polr()` in the MASS package, of Bill Venables and Brian Ripley.
- A negative binomial family for GLMs to model over-dispersion in count data is available in MASS.
- Models for overdispersed counts and proportions
  - Package pscl also contains several functions for dealing with over-dispersed count data. Poisson or negative binomial distributions are provided for both zero-inflated and hurdle models.
  - aod provides a suite of functions to analyse overdispersed counts or proportions, plus utility functions to calculate e.g. AIC, AICc, Akaike weights.
- Detecting change points and structural changes in parametric models is well catered for in the segmented package and the strucchange package respectively. segmented has recently been the subject of an R News article (R News, volume 8 issue 1).

Tree-based models

Tree-based models are being increasingly used in ecology, particularly for their ability to fit flexible models to complex data sets and the simple, intuitive output of the tree structure. Ensemble methods such as bagging, boosting and random forests are advocated for improving predictions from tree-based models and to provide information on uncertainty in regression models or classifiers.

If you click on the vegan link you'll see the following:



vegan: Community Ecology Package

Ordination methods, diversity analysis and other functions for community and vegetation ecologists.

Version:	1.17-8
Suggests:	MASS, mgcv, lattice, cluster, scatterplot3d, rgl, tcltk
Published:	2011-03-02
Author:	Jari Oksanen, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, Helene Wagner
Maintainer:	Jari Oksanen <jari.oksanen at oulu.fi>
License:	GPL-2
URL:	http://cran.r-project.org/, http://vegan.r-forge.r-project.org/
In views:	Environmetrics, Multivariate, Phylogenetics, Psychometrics, Spatial
CRAN checks:	vegan results

Downloads:

Package source:	vegan_1.17-8.tar.gz
MacOS X binary:	vegan_1.17-8.tgz
Windows binary:	vegan_1.17-8.zip
Reference manual:	vegan.pdf
Vignettes:	Design decisions and implementation Diversity analysis in vegan Introduction to ordination in vegan
News/ChangeLog:	NEWS ChangeLog
Old sources:	vegan archive

Reverse dependencies:

Reverse depends: BiodiversityR, FD, RandForestGUI, StatFingerprints, analogue, asbio, betaper, bipartite, blender, clustTool, cocorresp, isopam, mpmcorrelogram, palaeoSig, paleoMAS, paltran, pcurve, picante, rich, simba, untb, OTUbase
Reverse suggests: codep, mefa, primer, rioja, spaa, vegdata, yaImpute

Many of you may have installed packages but you did so through the Packages menu item in R and you may not be aware but for every R package there is one of these pages on CRAN. It is useful to know because these pages can contain a lot of information. For example, under Suggests or Depends, you'll see the other packages that are suggested or required for this package. Under In views, you'll see which task views link to this package so you can find possibly related packages. Under Cran checks are the results of the output from the examples in the package. Package Source, is a zipped file of the directory structure of the package which includes the R code and the documentation files. This is the actual source code and you can download it, see how it works if you are trying to do something similar or how you might want to modify it, if it doesn't suit your needs. This is the beauty of open source software. You can look at all of the code in R and its packages. Nothing is hidden or proprietary. I have had better luck unzipping these files with 7Zip (freeware) rather than WinZip.

The other zipped files are Binaries for Windows and Mac if the package works with those operating systems. The binary is what R uses to install the package. Following that is a reference manual which is a pdf of all the help files contained in the package and then some packages like this one contains one or more vignettes. That is not a word in my vocabulary so I had to look it up. It has many meanings but in this context it is a "short essay" that describes and demonstrates the package with one or more examples. This is useful material to read to understand the package and to see examples of its use and output.

For the more technical folks, each vignette is a pdf file which is constructed with Sweave. Sweave is a set of functions in R that interweave Latex (pronounced Lay tek) with R code to produce a document in conjunction with MikTex a Latex processor. In the vegan source directory, there is a sub-directory inst/doc which contains the files used to create the vignettes, so this is a good place to get Sweave examples.

At the bottom of the page is a Reverse Suggests and Depends. This lists the packages that use and require vegan (Depends) or suggest its use.

6) Reference card: Sometimes you just need a quick reminder of some of the syntax and basic functions and what they do. Various folks have created reference cards and these can be handy when you are first learning. The following link provides several of them. http://devcheatsheet.com/tag/r/.
7) User-supported help: One of the great aspects of open source software like R, is the user community that provides an amazing amount of tutorial material. The following link provides a variety of pdf documents http://cran.stat.ucla.edu/other-docs.html. And even more help and tutorial material can be found at a variety of websites with links here http://www.r-project.org/other-docs.html. An organized collaborative documentation/tutorial is provided at the RWiki (http://rwiki.sciviews.org/doku.php). One final source of information about R is the R Journal (http://journal.r-project.org/) which used to be called R News. This is a refereed online journal containing articles of interest to R users and developers.
8) Are you a blog-holic? Personally I am not, but I do subscribe to http://www.r-bloggers.com/ which sends me a single email once a day. At the top of the email is a list of the topics. I scan those and if I see something that interests me, I click on it and read it. Once I'm done I delete the email. It is a great way to be notified of new advances or example tutorials that may be useful for your particular needs. It is not invasive or over-burdensome and I've learned a lot from the postings.

Now, do NOT get over-whelmed by all of this material. Learn now what you need to accomplish your task and explore as you go. Set aside a little time each week to learn something new about R. Get proficient with the help files. When something isn't working or doesn't seem correct the help files should be the first place you should look.