Building R Packages

Derek Young

Department of Statistics
The Pennsylvania State University

S.O.S. Talk
April 13th, 2007

Outline of Topics

Introduction

Getting Started

Building Your R Package

Submission to CRAN

Introduction

- R is a (free) software environment used for statistical computing and graphics.

Introduction

- R is a (free) software environment used for statistical computing and graphics.
- If no prior experience before entering this program, usually you will be introduced to R in STAT 597C (Introduction to Computing Environments).

Introduction

- R is a (free) software environment used for statistical computing and graphics.
- If no prior experience before entering this program, usually you will be introduced to R in STAT 597C (Introduction to Computing Environments).
- This may be the primary computing language used in your research.

Introduction

- R is a (free) software environment used for statistical computing and graphics.
- If no prior experience before entering this program, usually you will be introduced to R in STAT 597C (Introduction to Computing Environments).
- This may be the primary computing language used in your research.
- If your functions have a general theme, then consider combining them into a package.

Introduction

- R is a (free) software environment used for statistical computing and graphics.
- If no prior experience before entering this program, usually you will be introduced to R in STAT 597C (Introduction to Computing Environments).
- This may be the primary computing language used in your research.
- If your functions have a general theme, then consider combining them into a package.
  - You may upload the package to the Comprehensive R Archive Network (CRAN).

Introduction

- R is a (free) software environment used for statistical computing and graphics.
- If no prior experience before entering this program, usually you will be introduced to R in STAT 597C (Introduction to Computing Environments).
- This may be the primary computing language used in your research.
- If your functions have a general theme, then consider combining them into a package.
  - You may upload the package to the Comprehensive R Archive Network (CRAN).
  - Since the summer of 2005, I developed a package called `mixtools`.

Operating Systems

- You can build R packages under the major operating systems (Windows, Linux, Mac OS).

Operating Systems

- You can build R packages under the major operating systems (Windows, Linux, Mac OS).
- Dan Bailey and Matt Nunes at the University of Bristol have provided an idiot's guide with helpful instructions pertaining to the different operating systems.

Operating Systems

- You can build R packages under the major operating systems (Windows, Linux, Mac OS).
- Dan Bailey and Matt Nunes at the University of Bristol have provided an idiot's guide with helpful instructions pertaining to the different operating systems.
- It does not matter under which OS you build your package as long as all of the "checks" that need to be performed are satisfied (to be discussed later).

Operating Systems

- You can build R packages under the major operating systems (Windows, Linux, Mac OS).
- Dan Bailey and Matt Nunes at the University of Bristol have provided an idiot's guide with helpful instructions pertaining to the different operating systems.
- It does not matter under which OS you build your package as long as all of the "checks" that need to be performed are satisfied (to be discussed later).
  - I built `mixtools` under Windows, so I will focus on building packages using this OS.

What You Need

- The biggest obstacle is getting the files and programs you need in the correct place on your system.

What You Need

- The biggest obstacle is getting the files and programs you need in the correct place on your system.
- Brian Ripley and Duncan Murdoch (R Project Contributors) have a building R for Windows site.

What You Need

- The biggest obstacle is getting the files and programs you need in the correct place on your system.
- Brian Ripley and Duncan Murdoch (R Project Contributors) have a building R for Windows site.
  - Here you will find links and downloads of everything you need.

What You Need

- The biggest obstacle is getting the files and programs you need in the correct place on your system.
- Brian Ripley and Duncan Murdoch (R Project Contributors) have a building R for Windows site.
  - Here you will find links and downloads of everything you need.
  - You will want to download only the bare essentials to avoid any conflicts with other programs you already have!

What You Need

- The biggest obstacle is getting the files and programs you need in the correct place on your system.
- Brian Ripley and Duncan Murdoch (R Project Contributors) have a building R for Windows site.
  - Here you will find links and downloads of everything you need.
  - You will want to download only the bare essentials to avoid any conflicts with other programs you already have!
- Since I developed `mixtools`, new versions of R as well as the release of Microsoft Vista have created new bugs for building the packages.

The Essentials

- Rtools: A set of Unix tools so that the command prompt in Windows acts like a Unix terminal.

The Essentials

- Rtools: A set of Unix tools so that the command prompt in Windows acts like a Unix terminal.
  - This includes commands such as `tar`, `gzip`, and `mkdir`.

The Essentials

- Rtools: A set of Unix tools so that the command prompt in Windows acts like a Unix terminal.
  - This includes commands such as `tar`, `gzip`, and `mkdir`.
  - The `.exe` files for other commands can be found online and placed in this directory as well.

The Essentials

- Rtools: A set of Unix tools so that the command prompt in Windows acts like a Unix terminal.
  - This includes commands such as `tar`, `gzip`, and `mkdir`.
  - The `.exe` files for other commands can be found online and placed in this directory as well.
- Perl 5.8.4 (or later): Necessary for reading the Perl scripts when building the HTML help files.

The Essentials

- Rtools: A set of Unix tools so that the command prompt in Windows acts like a Unix terminal.
  - This includes commands such as `tar`, `gzip`, and `mkdir`.
  - The `.exe` files for other commands can be found online and placed in this directory as well.
- Perl 5.8.4 (or later): Necessary for reading the Perl scripts when building the HTML help files.
- Microsoft HTML Help Workshop: Required for building the Windows help files.

The Essentials

- Rtools: A set of Unix tools so that the command prompt in Windows acts like a Unix terminal.
  - This includes commands such as `tar`, `gzip`, and `mkdir`.
  - The `.exe` files for other commands can be found online and placed in this directory as well.
- Perl 5.8.4 (or later): Necessary for reading the Perl scripts when building the HTML help files.
- Microsoft HTML Help Workshop: Required for building the Windows help files.
- MinGW: A compiler for any C and/or FORTRAN code in your package.

Some Caveats

- Make sure you have all the correct patches for the versions of the programs you are running!

Some Caveats

- Make sure you have all the correct patches for the versions of the programs you are running!
- All of the programs discussed earlier should be placed in directories that DO NOT have spaces in the names!

Some Caveats

- Make sure you have all the correct patches for the versions of the programs you are running!
- All of the programs discussed earlier should be placed in directories that DO NOT have spaces in the names!
- The directories for each of the programs need to be placed in your Path system variable (this can be edited under your system's Environment Variables).

Some Caveats

- Make sure you have all the correct patches for the versions of the programs you are running!
- All of the programs discussed earlier should be placed in directories that DO NOT have spaces in the names!
- The directories for each of the programs need to be placed in your Path system variable (this can be edited under your system's Environment Variables).
  - My Path has the programs in the following order: MinGW, Rtools, Miktex, Perl, R, HTML Workshop. These are placed BEFORE all of the other stuff already in my Path.

The General Steps

- Set your working directory to C:/.../R-2.x.x/src/library and use the command `package.skeleton` to create a template for your package.

The General Steps

- Set your working directory to C:/.../R-2.x.x/src/library and use the command `package.skeleton` to create a template for your package.
- Modify the "Title" and "Description" files to give all the information about your package (such as version number, required packages, authors, etc.).

The General Steps

- Set your working directory to C:/.../R-2.x.x/src/library and use the command `package.skeleton` to create a template for your package.
- Modify the "Title" and "Description" files to give all the information about your package (such as version number, required packages, authors, etc.).
- Place all R functions in the "R" folder.

The General Steps

- Set your working directory to C:/.../R-2.x.x/src/library and use the command `package.skeleton` to create a template for your package.
- Modify the "Title" and "Description" files to give all the information about your package (such as version number, required packages, authors, etc.).
- Place all R functions in the "R" folder.
- Place all C/FORTRAN code in the "src" folder.

The General Steps

- Set your working directory to C:/.../R-2.x.x/src/library and use the command `package.skeleton` to create a template for your package.
- Modify the "Title" and "Description" files to give all the information about your package (such as version number, required packages, authors, etc.).
- Place all R functions in the "R" folder.
- Place all C/FORTRAN code in the "src" folder.
- Place all data sets (saved in an R environment) in the "data" folder.

The General Steps

- Set your working directory to C:/.../R-2.x.x/src/library and use the command `package.skeleton` to create a template for your package.
- Modify the "Title" and "Description" files to give all the information about your package (such as version number, required packages, authors, etc.).
- Place all R functions in the "R" folder.
- Place all C/FORTRAN code in the "src" folder.
- Place all data sets (saved in an R environment) in the "data" folder.
- Place all documentation (written in LaTeX) in the "man" folder.

Specifics

- When writing `.Rd` files (the R documentation files) there are formatting guidelines CRAN expects you to follow.

Specifics

- When writing `.Rd` files (the `R` documentation files) there are formatting guidelines CRAN expects you to follow.
- In your `R` code, make sure that all logical values (`TRUE` and `FALSE`) are actually written out. DO NOT use `T` and `F`.

Specifics

- When writing `.Rd` files (the `R` documentation files) there are formatting guidelines CRAN expects you to follow.
- In your `R` code, make sure that all logical values (`TRUE` and `FALSE`) are actually written out. DO NOT use `T` and `F`.
- All arguments in your functions must match the arguments verbatim in the documentation.

The Final Product

- Run `RCMD CHECK mypkg` to see if there are any errors.

The Final Product

- Run `RCMD CHECK mypkg` to see if there are any errors.
  - All of these errors must be corrected before submission to CRAN.

The Final Product

- Run `RCMD CHECK mypkg` to see if there are any errors.
  - All of these errors must be corrected before submission to CRAN.
  - During this check, a `.pdf` help file of the package is also produced. The LaTeXcode for this file is provided so that you may easily insert this code into other documentation.

The Final Product

- Run `RCMD CHECK mypkg` to see if there are any errors.
  - All of these errors must be corrected before submission to CRAN.
  - During this check, a `.pdf` help file of the package is also produced. The LaTeXcode for this file is provided so that you may easily insert this code into other documentation.
- Upon correcting any errors, run `RCMD INSTALL mypkg` and then it is ready to use!

The Final Product

- Run `RCMD CHECK mypkg` to see if there are any errors.
  - All of these errors must be corrected before submission to CRAN.
  - During this check, a `.pdf` help file of the package is also produced. The LaTeXcode for this file is provided so that you may easily insert this code into other documentation.
- Upon correcting any errors, run `RCMD INSTALL mypkg` and then it is ready to use!
- All arguments in your functions must match the arguments verbatim in the documentation.

Submitting a Package to CRAN

- Run `RCMD BUILD mypkg` to get the release `.tar.gz` file.

Submitting a Package to CRAN

- Run `RCMD BUILD mypkg` to get the release `.tar.gz` file.
  - This is the file you upload to CRAN.

Submitting a Package to CRAN

- Run `RCMD BUILD mypkg` to get the release `.tar.gz` file.
  - This is the file you upload to CRAN.
- Regardless of which OS you built your package under, CRAN will eventually post the package source, Mac OS binary, Windows binary, and reference manual for your package.

Submitting a Package to CRAN

- Run `RCMD BUILD mypkg` to get the release `.tar.gz` file.
  - This is the file you upload to CRAN.
- Regardless of which OS you built your package under, CRAN will eventually post the package source, Mac OS binary, Windows binary, and reference manual for your package.
- All of the details can be found in the Writing R Extensions manual.