R Packages II:
Function helps and vignette

Review

- We have covered how to find and install R packages.

- We also covered how to create a simple R package, using "`package.skeleton`" function.

- In this lecture, we will cover:

  - Writing R function helps.

  - Create a package "vignette" using Sweave.

Writing R function helps

- In package, we have defined some functions (contained in R files under the /R directory).

- We want to provide helps for these functions that can be accessed by typing "`? function`" in R.

- The function helps are written in simple text files with extension **Rd**. These files are saved under **/man** directory.

- You don't have to have help files for making a package, e.g., `man` directory can be empty. But it's better to have function helps.

- If you were to submit package to CRAN/Bioconductor, function helps are required.

Creating an Rd file

- The simplest way is to use "**prompt**" function. It takes the name of a function and generates a template of the Rd file. You can then modify the Rd file to put in more information.

- For example, I have following function in my R workspace:

```
myRowSums=function(x) {
   n0 = nrow(x)
   result = rep(0, n0)
   for(i in 1:n0) {
     result[i]=sum(x[i,])
   }
   return(result)
}
```

- I can use prompt to create a help file (saved under current directory) for this function:

```
> prompt(myRowSums)
Created file named 'myRowSums.Rd'.
Edit the file and move it to the appropriate directory.
```

myRowSums.Rd

```
\name{myRowSums}
\alias{myRowSums}
%- Also NEED an '\alias' for EACH other topic documented here.
\title{
%%  ~~function to do ... ~~
}
\description{
%%  ~~ A concise (1-5 lines) description of what the function does. ~~
}
\usage{
   myRowSums(x)
}
%- maybe also 'usage' for other objects documented here.
\arguments{
  \item{x}{
%%      ~~Describe \code{x} here~~
  }
}
```

```
\details{
%%  ~~ If necessary, more details than the description  above ~~
}
\value{
%%  ~Describe  the value returned
%%  If it is a LIST, use
%%  \item{comp1 }{Description  of 'comp1'}
%%  \item{comp2 }{Description  of 'comp2'}
%% ...
}
\references{
%% ~put references  to the literature/web  site here ~
}
\author{
%%  ~~who you are~~
}
\note{
%%  ~~further  notes~~
}
```

\seealso{

%% ~~objects to See Also as \code{\link{help}}, ~~~

}

\examples{

##---- Should be DIRECTLY executable !! ----

##-- ==>   Define data, use random,

##--   or do   help(data=index)   for the standard data sets.

}

% Add one or more standard keywords, see file 'KEYWORDS' in the

% R documentation directory.

\keyword{ ~kwd1 }

\keyword{ ~kwd2 }% __ONLY ONE__ keyword per line

Rd file format

- Similar to a LaTex file format.

- Multiple sections. Section title starts with \. Section is within curly braces {}. Pay attention to the pairing of curly braces.

- Required fields: `\name, \title, \description, \usage, \arguments, \value`

- Others are optional: `\alias, \details, \references, \author, \note, \seealso, \examples, \keyword`.

- The `arguments` field, use \item to create bulleted list.

Edited myRowSums.Rd

```
\name{myRowSums}
\alias{myRowSums}
\title{
  My own function to compute means of rows of a matrix.
}
\description{
  Computing the row sums of a matrix. I do it using a loop.
}
\usage{
myRowSums(x)
}
\arguments{
  \item{x}{A numeric matrix.}
}
\details{
  This is pretty simple. Don't really need any more explanation.
}
```

```
\value{

  Returns a vector with length equals to the number of rows of the input
matrix.
}
\author{

  Hao Wu <hao.wu@emory.edu>

}
\note{

  None.

}


\seealso{

  rowMeans, apply

}
\examples{
x=matrix(rnorm(100,nrow=20))

myRowSums(x)

}


\keyword{test}
```

Build/install package with function help

- We can use the same commands to build (`R CMD build`) and install (`R CMD INSTALL`) the package.

- This time the function helps will be available, and can be accessed by "?function".

What we get

```
myRowSums                  package:coolpkg                  R Documentation

My own function to compute means of rows of a matrix.

Description:

    Computing the row sums of a matrix. I do it using a loop.

Usage:

    myRowSums(x)

Arguments:

     x: A numeric matrix.

Details:

    This is pretty simple. Don't really need any more explanation.

Value:

    Returns a vector with length equals to the number of rows of the
    input matrix.

Note:

    None.

Author(s):

    Hao Wu <hao.wu@emory.edu>

See Also:

    rowMeans, apply

Examples:

    x=matrix(rnorm(100,nrow=20))
    myRowSums(x)
```

Data in R packages

- If one wants to distribute data with an R package, the saved data files (with extension .RData or .rda) need to be put under the `data` directory.

- The data then can be loaded using "data" function. For example, if there's **mydata.RData** under **\data** directory for my package, doing "**data(mydata)**" will load that data into R.

- "prompt" function can be used to create Rd file for the help page for the data.

Other (more advanced) package contents

- In addition to the Rd files, one can provide additional software manual or user guide (often under `/inst/doc` or `/vignette`). These can be written using Sweave (discuss a little later).

- A `CITATION` file can be used to include references to literature connected to the package, the contents of the file can be accessed from within R using:

  `citation("coolpkg")`.

- If there are C/Fortran functions, the complied binary need to be loaded when loading the library. In that case a "processing function" (traditionally called `zzz.R`) is needed.

Sweave and R vignette

What is Sweave

- Sweave is a **system** that enables integration of R code into LaTeX documents. The purpose is "*to create dynamic reports, which can be updated automatically if data or analysis change*".
- Sweave is an also an **R function** (?Sweave) that implements the system.
- The name of Sweave (S-weave) is from Cweave, which is a way to generate documentations for C.

The basic ideas of Sweave

- When writing a document such as data analysis report, one can:
    - perform all analyses and save the figures/tables, then create a document with these; or
    - write an Sweave file, which is a mix of LaTex and R codes. At compilation, the R codes will run and result figures/tables will be included in the document automatically.

The steps of Sweave

The steps of using Sweave to create a document are:

1. Create a text file with extension .Rnw. The text file is a mixture of LaTex and R codes. Assume the file is called coolpkg.Rnw.

2. Process the file using Sweave function:

   ```
   > Sweave("coolpkg.Rnw")
   Writing to file coolpkg.tex
   Processing code chunks with options …
   You can now run (pdf)latex on 'coolpkg.tex'
   ```

   This generates a .tex file under current directory. What it did is to run the R codes embedded in Rnw, and replace those with proper tex commands.

3. Compile the tex file using your LaTex compiler (latex, pdflatex, etc.)

A (tiny) bit of LaTex

- LaTex is a high quality typesetting system (like MS Office on PC or iWork on Mac).

- Supports multiple types documents: article, slides.

- Widely used in academia, especially when there are a lot of equations. So this is important for statisticians.

- The idea is to write the source in a "markup language" as a text file (with extension .tex), then compile and generate the document in postscript or pdf.

Compared to other WYSIWYG software

- Compared to other "what you see is what you get" typesetting software (MS Office, Google docs, iWorks, etc.), LaTex has following advantages:
  - Free.
  - Better, more professional looking.
  - Better control of formatting, if you know it.
  - Faster and smaller, because it's a simple text file.
  - Most importantly, faster to type equations once you know it (note that you can type LaTex in Word to generate equations now).
- The disadvantages are:
  - A little more difficult to learn compared to, say, MS Word.
  - All formatting needs to be set manually. This could be troublesome sometimes.
  - Lack of some useful features (e.g., no track changes.).

A simple LaTex example

```
\documentclass{article}
\title{Manual of coolpkg}

\begin{document}
\maketitle

\section{Introduction}
Some introduction to my coolpkg.

\section{Results}
Some data analysis results.

\section{Conclusion}
Conclusion texts are here.

\end{document}
```

Compiling LaTex

- You will need some software to compile LaTex:
  - On Linux, the "latex" and "pdflatex" command are distributed with the OS, so no additional installation needed.
  - On Mac, one needs to install MacTex, and use TexShop as the GUI.
  - On PC, use MikTex and WinEdt.
  - There are other options. Google them.

What I get

The tex file is saved as coolpkg.tex. After compilation (running "`pdflatex coolpkg.tex`" at command window), I get the following cookpkg.pdf.

<div align="center">

Manual of coolpkg

February 20, 2013

</div>

1 Introduction

Some introduction to my coolpkg.

2 Results

Some data analysis results.

3 Conclusion

Conclusion texts are here.

Now back to Sweave

- A Sweave file (.Rnw) is a mix of LaTex and R codes. The R codes are within special tags: start by **<<>>=** , and end by **@**.

- Control commands can be specified within <<>>.

- For example, to include some R codes and results, one can add following in the tex:

```
\section{Results}
Some data analysis results. Below is example of table function.
<<echo=TRUE, eval=TRUE>>=
table(mtcars$cyl)
@
```

After Sweave

After running Sweave("coolpkg.Rnw"), I got coolpkg.tex like this:

```
\documentclass{article}
\usepackage{Sweave}
......

\section{Results}
Some data analysis results. Below is an example histogram.
\begin{Schunk}
\begin{Sinput}
> table(mtcars$cyl)
\end{Sinput}
\begin{Soutput}
 4  6  8
11  7 14
\end{Soutput}
\end{Schunk}
```

The document

After compiling coolpkg.tex (Sweavy.sty is needed!), the pdf looks like:

<div align="center">

Manual of coolpkg

February 21, 2013

</div>

1 Introduction

Some introduction to my coolpkg.

2 Results

Some data analysis results. Below is an example histogram.

```
> table(mtcars$cyl)

 4  6  8
11  7 14
```

3 Conclusion

Conclusion texts are here.

Include figures

- To include figure, one can do:

```
\section{Results}
Some data analysis results. Below is example of table function.
<<echo=TRUE, eval=TRUE>>=
table(mtcars$cyl)
@

Below is a histogram:
<<fig=TRUE>>=
  hist(rnorm(100))
@
```

After Sweave

After running Sweave("coolpkg.Rnw"), now the coolpkg.tex is:

```
\documentclass{article}
\usepackage{Sweave}
......

\section{Results}
......
Below is a histogram:
\begin{Schunk}
\begin{Sinput}
>    hist(rnorm(100))
\end{Sinput}
\end{Schunk}
\includegraphics{coolpkg-002}
```
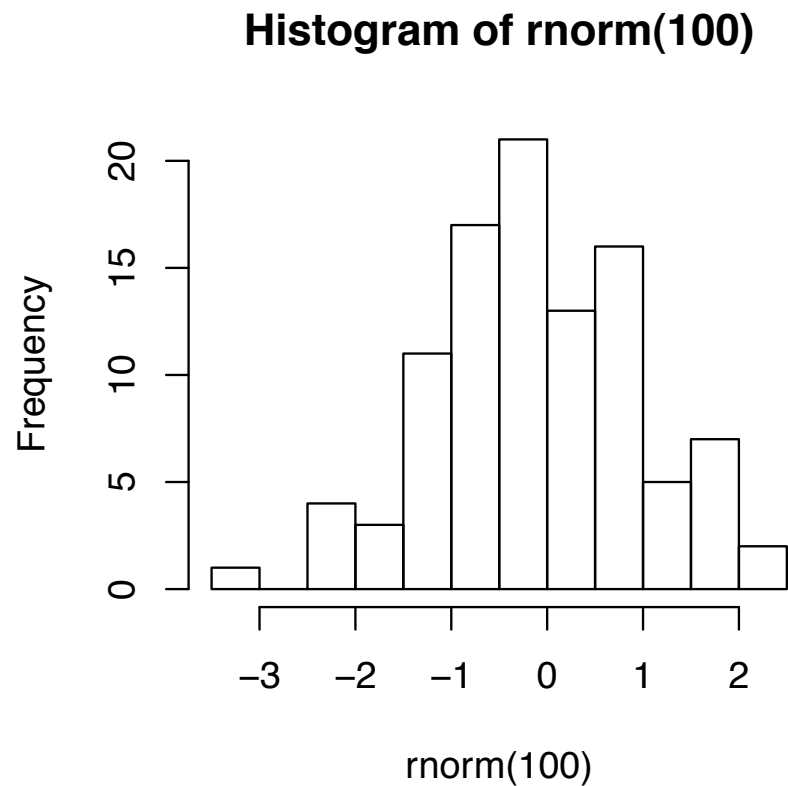
The document with the figure

After compiling the tex, the figure is there now:

Below is a histogram:

```
>   hist(rnorm(100))
```

Histogram of rnorm(100)

The Sweave commands

Now you've learned it! You can start to write your homework using Sweave.

The only other things you need to know are the commands (within <<>>), and there aren't many. Here are the most common ones:

- **echo**: logical (default is TRUE). Indicating whether to Include R code in the output file.
- **eval**: logical (default is TRUE). Indicating whether to evaluate (run) the code and include the results in the output.
- **fig**: logical (default is FALSE), indicating whether to include figures produced from the code.

For others, read manual.

What is an R vignette

- A vignette is an optional, supplemental documentation for an R package. Note that the R reference manual is merely a list of function helps.

- Not all packages have vignette. Bioconductor packages are required to have vignette.
  - `vignette(all = TRUE)` lists of functions with vignette.

- For those packages with vignette, the vignette can be accessed by vignette function:
  - `vignette("Sweave")` opens the vignette (a pdf) in a separate window for package "Sweave".

Build an R package with a vignette

- Vignette is usually created using Sweave.
- The Sweave file (.Rnw) is distributed with the package under `/vignettes`.
- When building R package (e.g., running `R CMD build`), the Rnw file will be processed, and a pdf vignette will be created.
- The vignette can be accessed in R using "vignette" function.

Review

What we've covered today:

- How to write R function helps and embed them into a package.

- A little bit of LaTex.

- How to use Sweave to write documents with mixed R and latex codes.

- How to write R vignette with Sweave.

In lab:

- Create function helps and a vignette for your package.