

An Introduction to R

Christopher G. Wilson, Ph.D.

Loma Linda University
Dept. of Pediatrics and Center for Perinatal Biology

May 13, 2015

Outline

1. Project 1 due today
2. Why use R?
3. R and supporting software installation
4. Working within R

Project 1

Please make sure that the IPython notebook for Project 1 is in the class GitHub repository.

Why should you use R

- ▶ Complete statistical environment and programming language
- ▶ Efficient functions and data structures for data analysis
- ▶ Powerful graphics
- ▶ Access to fast growing number of analysis packages
- ▶ Most widely used language in bioinformatics
- ▶ Is standard for data mining and biostatistical analysis
- ▶ Technical advantages include free, open-source, multi-platform

Books and Documentation

- ▶ simpleR - Using R for Introductory Statistics (John Verzani, 2004)
- ▶ Bioinformatics and Computational Biology Solutions Using R and Bioconductor (Gentleman et al., 2005)
- ▶ Some free tutorials/books are available at:
<http://www.statmethods.net/about/books.html>
- ▶ Extensive documentation available at
<http://www.r-project.org/>

Package Depositories

There are many contributed packages that make R extremely versatile

- ▶ CRAN (> 3500 packages) general data analysis
- ▶ Bioconductor (> 700 packages) bioscience data analysis
- ▶ Omegahat (> 30 packages) programming interfaces

R Installation

- ▶ Install *R* for your operating system from:
`http://cran.r-project.org/mirrors.html`
- ▶ Install *R-Studio* from:
`http://www.rstudio.com/ide/download`

Installation of CRAN Packages

From the command prompt, install packages:

```
> install.packages(c("pkg1", "pkg2"))  
> install.packages("pkg.zip", repos=NULL)
```


Installation of BioConductor Packages

From the command prompt, install packages:

```
> source("http://www.bioconductor.org/biocLite.R")  
> library(BiocInstaller)  
> BiocVersion()  
> bioclite()  
> bioclite(c("pkg1", "pkg2"))
```

Startup and Closing R

- ▶ **Starting R:** The R GUI versions, including RStudio, under Windows and Mac OS X can be opened by double-clicking their icons. Alternatively, one can start it by typing 'R' in a terminal (default under Linux)
- ▶ **Closing R:** The R environment is controlled by hidden files in the startup directory: .RData, .Rhistory and .Rprofile (optional).
- ▶ **Close R by typing:** `q()`

Getting Around

Create an object with the assignment operator `<` *—(or =)*

```
> object < —...
```

List objects in current R session

```
> ls()
```

Return content of current working directory

```
> dir()
```

Return path of current working directory

```
> getwd()
```

Change current working directory

```
> setwd("/home/user")
```

Basic R Syntax

General R command syntax

`> object <- function_name(arguments)`

`> object <- object[arguments]`

Finding help

`> ?function_name`

Load a library

`> library("my_library")`

Lists all functions defined by a library

`> library(help = "my_library")`

Load library manual (PDF file)

`> vignette("my_library")`

Executing R Scripts

Execute an R script from within R:

```
> source(" my_script.R" )
```

Execute an R script from command-line:

```
$ Rscript my_script.R
```

```
$ R CMD BATCH my_script.R
```

```
$ R - -slave < my_script.R
```

Reading and Writing External Data

Import data from tabular files into R:

```
> myDF <- read.delim("myData.xls", sep = "\t")
```

Export data from R to tabular files:

```
> write.table(myDF, file = "myfile.xls", sep = "\t", quote = FALSE, col.names = NA)
```

Copy and paste (e.g. from Excel) into R:

On Windows/Linux systems:

```
> read.delim("clipboard")
```

On Mac OS X systems:

```
> read.delim(pipe("pbpaste"))
```

For the remainder of class: Install *R* and *R-Studio*
and work through Holly Harlin's R tutorial