Introduction to R

Hao Helen Zhang

Department of Mathematics
University of Arizona
hzhang@math.aricona.edu

Fall 2019

What is R

R is the most powerful and most widely used statistical software

- A language and environment for statistical computing and graphics.
- an integrated suite of software facilities for data manipulation, calculation and graphical display.
- provides a wide variety of classical and modern statistical techniques

Video:

$$http://www.youtube.com/watch?v=TR2bHSJ\_eck$$

More about R

- R is free
- R is open to all the users.
- **You can contribute to R!**
- R is strong for visualization. It can produce well-designed publication-quality plots, mathematical symbols and formulae.
- R can run on Unix, Linux, Windows and Mac

Download R at

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

Important Features of R

All *R* functions and datasets are stored in *packages*.

- Eight *basic packages*
- Can be extended via *contributed packages*, allows users to add new functions.

  cran.r−project.org/web/packages/available_packages_by_date.html

For computationally-intensive tasks,

- Advanced users can write C code to manipulate R objects directly.
- C, C++ and Fortran code can be linked and called at run time.

Examples of Packages

- base: R basic package
- MASS: main library of Venables and Ripley's MASS
- tree: classification and regression trees
- svmpath: fit the svm path
- mgcv: multiple smoothing parameter estimation and GAMs by GCV
- splines: regression spline functions and classes

Load and Check Packages

Only when a package is loaded are its contents available.

- To install a package, using the function install.packages() or through a menu item in Windows.
- To load a package, use library(package_name).
- To see which packages are installed at your computer, use library().
- To keep track of currently loaded packages, use search(). Note: this also lists the currently attached data frames and lists.

R Library

```
### list all available packages
> library()

### load packages
> library(MASS)
> library(tree)
> library(splines)
> library(svmpath)

### documentation on package
> library(help = splines)

### download and install packages from CRAN or local files
> install.packages (...)
```

Data Structures in R

- Three types of data in R: numeric, character, and logical.
- R supports *vectors, matrices, lists*, and *data frames*.
  - *Lists* provide a very general way to hold a collection of arbitrary R objects.
  - A *data frame* is a cross between a matrix and a list, - columns of a data frame can be different types, but they all must be the same length.
- Objects can be assigned values using "=" or $\leftarrow$ operator.
- **R is highly vectorized** - almost all operations work equally well on scalars and arrays.
- **Work with vectors whenever you can!** It is much more efficient than working with scalars.

Basic Commands

Get familiar with

- help(function_name)
- Input and output data and results
- Generate random data
- Basic statistical analysis
- Fit linear models
- Graphics (scatter plot, histogram)
- Write functions

How to Get Help

```
### load the package MASS (functions and datasets to
### support ‘Modern Applied Statistics with S’).
> library(MASS)

### get help
> help.start()
> help(mvrnorm)
> ? lm

### load the data from the local directory
> read.table(‘‘file_name’’)
> write.table()

### quit R
> q()
```

Example 1: Data Generation

```
### generate a sequence
> a = seq(0,10,0.5)

### generate samples from a specified distribution.
> set.seed(2014)
> x = rnorm(10, mean=0, sd=1)    # samples from N(0,1)
> x = runif(10, min=0, max=1)      # sample from Unif(0,1)
> x = runif(20, min=-2, max=2)     # sample from Unif(-2,2)

### generate data from multivariate normal distribution
> mean1 = c(2,1)
> cov1 = matrix(c(1,0,0,1),nrow=2)
> data1 = mvrnorm(5,mean1,cov1)
```

Example 2: Plotting

```
### low-level plotting
> x = seq(0,10,0.5)
> y = seq(2,12,0.5)
> plot(x,y)
> lines(x,y)
> abline(a=0,b=1)

### plot histograms
> hist(x)

### plot bar plots
> barplot(x)

### plot density
> plot(density(x))
```

Example 3: Data Import and Summary

```
### load data
> data(cars)

### show data
> cars

### make the data columns available by name
> attach(cars)

### compute mean and variance
> mean(dist)
> var(dist)
```

Example 4: Fit Linear Regression

```
### fit a simple linear  model
> mymodel =  lm(dist ~ speed)
> summary(mymodel)

### draw the scatter plot
> plot(cars, main="Stopping Distance versus Speed")

### draw the fitted regression line (red)
> lines(speed, fitted(mymodel), type="l", lty=1, col=2)

### draw a smooth line through a scatter plot (green)
> lines(stats::lowess(cars), type="l", lty=2, col=3)
> detach(cars)
```

Example 5: Write Your Own Functions

```
### fit a simple linear  model
> my.fun <- function(x, y){
    a <- mean(x)-mean(y)
  return(a)
  }

> x <- runif(50,0,1)
> y <- runif(50,0,3)
> output <- my.fun(x,y)
> print(output)
```

Useful Tutorial

- *Introduction to R*: long version (109 pages)
- *R for Beginners*: medium version (31 pages)
- *R Reference Card*: quick reference (1 page)

Download from our course page

http://www.math.arizona.edu/∼hzhang/math574m.html