R Fundamentals

Data Types and Basic Manipulation

http://bit.ly/2lcttua

What is R?

R is a open-source programming language and environment for **statistical** computing and graphics.

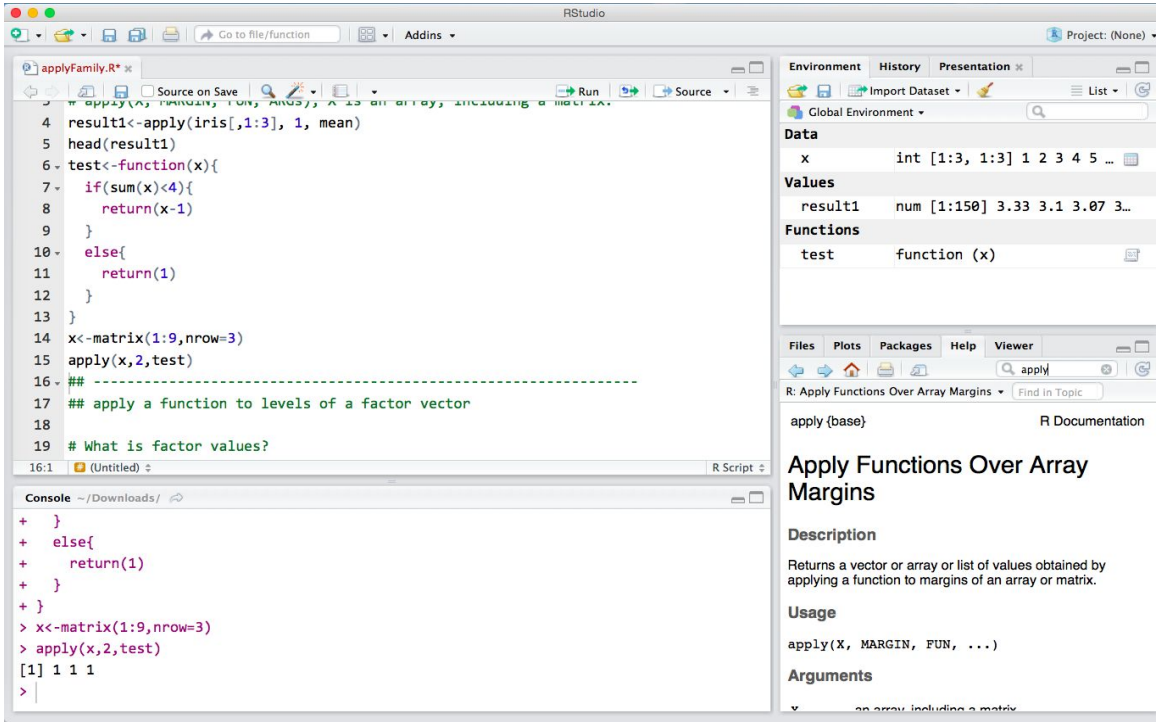
You can use R as it is, but most people prefer to use R in combination with the RStudio (GUI-based IDE for R), which has an organized layout and several extra options (RMarkdown, Shiny, etc).

To use R:

1) Install R https://www.r-project.org/
2) Install RStudio https://www.rstudio.com/

RStudio Interface

- Console
  - Where commands are actually executed
  - “>” prompt
- Script
  - Where commands are edited and saved
  - Click Run or press CTRL+ENTER(Windows)/command+return(MacOSX) to send it to the command window



- Workspace/history window
  - Workspace shows data and values R has in its memory
  - The history window shows what has been typed before
- Files / plots / packages
  - Where you can open files, view plots (also previous plots), install and load packages or use the help function

Basic R Concepts

- Working Directory
  - Your working directory is the folder on your computer in which you are currently working
  - Open, save
  - getwd()
  - setwd(“"C:/Users/yj2360”) - forward slash
- Fundamental Data Objects
  - vector(): 1-D data; all elements of the same type
  - matrix(): 2-D data, all values of the same type
  - array(): multi-D data, all values of the same type
  - list(): one-D data, elements of different modes
  - data.frame(): 2-D data, elements of different types

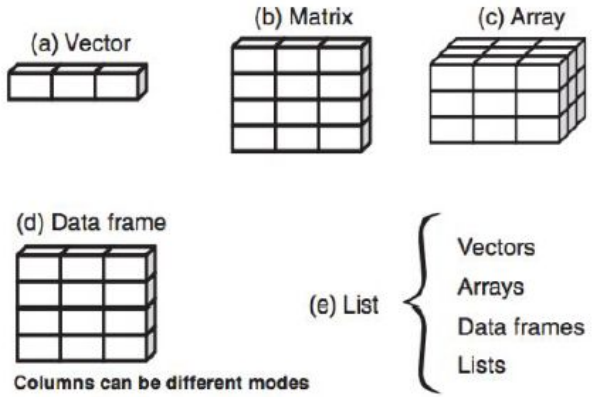


Figure 2.1 R data structures

- Factor values
  - A factor is a character vector with information about the possible categories, called the levels of the factor

Matrix and Data Frame

```
m<-matrix(1:4,nrow=2,ncol=2)
m
dim(m)
x<-1:3
y<-10:12
cbind(x,y)
rbind(x,y)
```

```
dat<-data.frame(id=letter[1:5],
x=1:10, y=11:20)
t = data.frame(x = c(11,12,14), 2 y
= c(19,20,21), z = c(10,9,7))
t
mean(t$z)
```

commonly used data frame functions:

head(); dim(); nrow(); ncol(); str(); names()

Basic R Operations

- Creation and assignment
  - Use <- to create an object and assign values to it
- Commenting
  - Anything after # will not be evaluated/executed
  - Used to leave notes for others reading your code
- Importing data
  - Read in different forms of data
- Filtering data
  - Logical and relational operators: ! & | == != >= <=
- Plotting Data
  - Use plot() function to plot data

Looping

Looping is equivalent to iterating or just replicating instructions. Let the computer take over the repetitive work and you will find your life becomes much easier.

For Loop

```
for(variable in sequence) {
   statements
}
```

```
> data <- rep(2,10)
> for(i in 1:10)
      {
        data[i] <- i^2
        if(data[i]==3) {next}
        data[i]
      }
```

While Loop

```
while(condition) {statements}
```

```
> a<-NULL
> i=1
> while (length(a)<10) {
  a[i]<-i
  i=i+1
  }
```

Writing Functions

A function is a machine which turns input objects (arguments) into an output object (return value), according to some algorithm.

```
myfct <- function(arg1, arg2, ...) {
   function_body
}
```

```
> fun1 <- function(arg1, arg2 )
       {
       w = arg1 ^ 2
       return(arg2 + w)
       }
> fun1(arg1 = 3, arg2 = 5)
  14
```

Using Packages

Applications of R normally use a package; i.e., a library of special functions designed for a specific problem.

install.packages("<the package's name>")

library("<the package's name>")

Most commonly used packages are

- Data processing: dplyr, tidyr, reshape2, XLConnect
- Plotting: gglot2, htmlwidgets, rgl
- Data modeling: car, forecast, glmnet, zoo, randomForest, caret
- Other: knitr, RColorBrewer, ggmap, shiny

More resources

- Help function
  - Use ?<function name> if you know the exact function name
  - Use ??<function name> to do a fuzzy search
- Swirl package
  - An interactive learning package for first-time R users
- RStudio's Cheatsheets
- Books
  - R for Data Science
  - R Cookbook
  - R for Statistical Analysis
- Online courses