Notes for Week 1

Rob Hayward

September 26, 2013

There are five main objects in R

- character

- numeric (real numbers)

- integer

- complex

- logical

The most basic object is a *vector*. A vector can only contain objects of the same class. However, there is one exception. The exception is a *list*. A list is a sort of vector that can contain different class elements. There are to elements to a vector: the class of object and the length.

A number is a numeric (a double precision real number). If you want an integer, you have to speficy an L suffix. In is infinity. NaN is an undefined number (not a number).

Attributes of an object tell us something about it. Attributes() let you set or modify the attibutes of an object.

Expressions are typed at the prompt. Useful ways of creating matrices is to rbind or cbind vectors together.

Lists can be created with different class of object. The elements of a list will have [[1]], while the elements of the part of the list are []. Factors are be ordered or unordered. Factors are treated in a special way by modelling functions like lm and glm. Levels of a factor can be set an an argument of the factor function. The baseline level is the first factor. This is done by alphabetical order unless set.

NA can have a class (ie numeric NA or character NA). NaN value is NA but NA is not NaN. For example, NA is not a missing value (it has been ascribed NA).

A datafrmae is a special type of list. Each element of the list must have the same length. They do not have to be the same type.

Elements of any object can be named. For example.

```
x <- 1:3
names(x)

## NULL

names(x) <- c("foo", "bar", "norf")
x

##  foo  bar norf
##    1    2    3

names(x)

## [1] "foo"  "bar"  "norf"
```

Lists and matrices can also have names in this way.

[] can be used to extract or subset information from a vector. You will get back something of the same class. [[]] can only be used to extract ONE element and you can change the class from that of the vector. Often used on a list or a data frame. $ can used a name but its semantics are similar to [[]]. When subsetting a matrix you will get back a vector. If you take the first column of a 3*3 matrix and you want a matrix you have to

```
x <- matrix(1:6, nrow = 2, ncol = 3)
x[1, ]

## [1] 1 3 5

x[1, , drop = FALSE]

##      [,1] [,2] [,3]
## [1,]    1    3    5
```

Subsetting lists

```
x <- list(foo = 1:4, bar = 0.6)
x[1]

## $foo
## [1] 1 2 3 4

x[[1]]

## [1] 1 2 3 4

x$bar

## [1] 0.6
```

```
x[["bar"]]

## [1] 0.6

x["bar"]

## $bar
## [1] 0.6
```

The last of these will return a list.

Using the [[]] is particularly useful if you are using a computed name.

```
x <- list(foo = 1:4, bar = 0.6, baz = "hello")
name <- "foo"
x[[name]]

## [1] 1 2 3 4

x$name

## NULL

x$foo

## [1] 1 2 3 4
```

The second of these will not work because it is looking for the object "name".

The [[]] can also be used to extract an element from a list element. So [[2,3]] will extract the third element from the second item in a list.

Complete cases will identify those rows or or elements of more than one vector that are complete.

Peng Video

1 Vectorisation

```
x <- 1:4
y <- 6:9
x + y

## [1]  7  9 11 13

x > 2

## [1] FALSE FALSE  TRUE  TRUE

x <= 2
```

```
## [1]  TRUE  TRUE FALSE FALSE

y == 8

## [1] FALSE FALSE  TRUE FALSE

x * y

## [1]  6 14 24 36
```

This is element-wise. If you have two matrices and want to multiply, you must us

2   Reading and Writing Files

read and write data into or out of R with these commands. It is important to read the help files for read.table. Using the options will speed the process. Can input a few lines and then set the class of variables.

Dumping and duputing can be used to input data that is in other formats. These will maintain more of the information. These are particularly useful for textual formats. Using dput you can export the R object to a file. It contains informtaioin that will allow the R object to be reconstrcuted. Dump can be used for multiple objects. Dput can only deal with one object.

Others

- file - opens connection to a file (including web page)

- gzfile - gzip

- bzfile - bzip2 files

It is possible to use 'connection' to read files. Need to open a connection and the close it. url function can open a connection to a web page.

If you have a function, you can load it into the session using "source".

'split' function will split a dataframe into components that are specified.

3   Assignment One

```
read.csv(hw1.data.zip)

## Error:  object 'hw1.data.zip' not found
```