R Programming Cousera Notes

Andrew Andrade (andrew *at* andrewandrade.ca)

17/9/2014

Professor: Roger D. Peng, PhD, Jeff Leek, PhD, Brian Caffo, PhD  
Johns Hopkins University

# Data Types and Basic Operations

## Objects

### Atomic classes of objects

* Character
* Numeric
* Integer
* Complex
* Logical

### Vector

* Vectors can only contain objects of same class
* List is a vector that can contain objects of different classes
* Empty vector can be created using vector() function

### Number

* Generally treated as a double real number

class(1)

## [1] "numeric"

* use L suffix for integers

class (1L)

## [1] "integer"

# Goals: To write functions  
# To write functions that send back multiple objects.  
  
# FIRST LEARN ABOUT LISTS --  
X = list(height=5.4, weight=54)  
print("Use default printing --")

## [1] "Use default printing --"

print(X)

## $height  
## [1] 5.4  
##   
## $weight  
## [1] 54

print("Accessing individual elements --")

## [1] "Accessing individual elements --"

cat("Your height is ", X$height, " and your weight is ", X$weight, "\n")

## Your height is 5.4 and your weight is 54

# FUNCTIONS --  
square <- function(x) {  
 return(x\*x)  
}  
cat("The square of 3 is ", square(3), "\n")

## The square of 3 is 9

# default value of the arg is set to 5.  
cube <- function(x=5) {  
 return(x\*x\*x);  
}  
cat("Calling cube with 2 : ", cube(2), "\n") # will give 2^3

## Calling cube with 2 : 8

cat("Calling cube : ", cube(), "\n") # will default to 5^3.

## Calling cube : 125

# LEARN ABOUT FUNCTIONS THAT RETURN MULTIPLE OBJECTS --  
powers <- function(x) {  
 parcel = list(x2=x\*x, x3=x\*x\*x, x4=x\*x\*x\*x);  
 return(parcel);  
}  
  
X = powers(3);  
print("Showing powers of 3 --"); print(X);

## [1] "Showing powers of 3 --"

## $x2  
## [1] 9  
##   
## $x3  
## [1] 27  
##   
## $x4  
## [1] 81

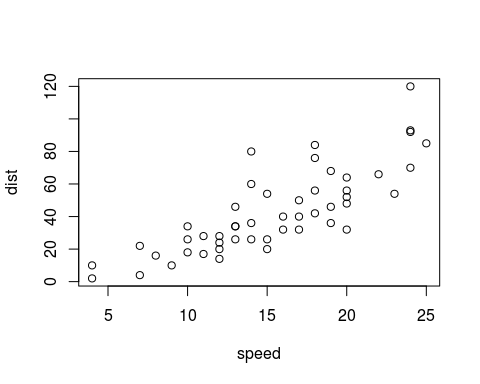
# WRITING THIS COMPACTLY (4 lines instead of 7)  
  
powerful <- function(x) {  
 return(list(x2=x\*x, x3=x\*x\*x, x4=x\*x\*x\*x));  
}  
print("Showing powers of 3 --"); print(powerful(3));

## [1] "Showing powers of 3 --"

## $x2  
## [1] 9  
##   
## $x3  
## [1] 27  
##   
## $x4  
## [1] 81

# In R, the last expression in a function is, by default, what is  
# returned. So you could equally just say:  
powerful <- function(x) {list(x2=x\*x, x3=x\*x\*x, x4=x\*x\*x\*x)}

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.