STA 141A
Fundamentals of Statistical Data Science

Fall 2016

Instructor: Debashis Paul

Lecture 2

More on Vectors

- Vectors can be used to store a sequence of **scalars**, **characters**, **logical values** (TRUE or FALSE) or **missing values** (NA). All the entries need not be of same ***type***, in which case a common type will be determined by R.

```
x = c(10,11.5,NA,-1.6)  # a numeric (floating point) vector with a missing value
x = c( “a”, “bc”, “last element”) # a vector with character strings as elements
x = c(T,F,T,T)   # or x = c(TRUE,FALSE,TRUE,TRUE)  ;  a vector of logical entries
x = c(“abc”,1.4,-3.4,F) # treated as a vector of characters
typeof(x) # returns the type of the vector x: (i) double; (ii); character; (iii) logical; (iv) character
seq(2,7,1) # creates the vector c(2,3,4,5,6,7)
seq(5,3,-0.5) # creates the vector c(5,4.5,4,3.5,3)
```

Vector operations

- Apart from combining two vectors by mathematical operations such as addition (+), subtraction (-), multiplication (*), division (/), power (^), remainder after division (%%) etc., which operate elementwise, we can also extract parts of vector and apply various R functions to it.

length(x) # dimension of the vector x

mean(x); median(x); hist(x); plot(x) # only work if x is numeric

mean(x, na.rm=T) # computes mean of x, while skipping over missing values (NA)

c(x,7:15) # or c(x,c(7:15)) append or ***concatenate*** the vector c(7:15) to x

x[2:4] # 2nd to 4th elements of x

x[-3] # extract all but the 3rd element of x in a vector

x[-c(3:5)] # extract the vector excluding 3rd, 4th and 5th element

Vector equality and inequality

- The operator (function) == is used to check equality

```
“==”(3,2) # same as 3 == 2;  returns FALSE
x = 1:3; y = c(4,2,5); x == y   # returns c(FALSE,TRUE,FALSE)
any(x==y) # returns TRUE
all(x==y) # returns FALSE
which(x==y) # returns 2 (position of the elements that are equal)
x > y  # returns c(TRUE,FALSE,FALSE)
x >= y # returns c(TRUE,TRUE,FALSE)
identical(x,y) # FALSE
x = 1:3; y = c(1,2,3); identical(x,y) # returns FALSE !! (typeof(x) is “integer”, typeof(y) is “double”)
```

Filtering a vector

- Can use a logical condition to extract parts of a vector, this is termed filtering

```
x = c(4,5,7,-10,-2,3)
x[x>0]  # returns c(4,5,7,3);  same as      u = (x > 0);  x[u]
y = 1:6; y[x>0] # returns c(1,2,3,6)
```

- Can use subset( ) function to do the same, which also handles NA effectively

```
subset(x,x>0) # returns c(4,5,7,3);  same as  x[x>0]
z = c(4,5,-3,NA,-4)
z[z > 1] # returns c(4,5,NA)
subset(z,z>1)  # returns c(4,5)
```

Working with character strings : first look

- Can use vectors to store characters or words. Can *paste* the elements of the vector together to create a sentence

```
wrd = c(“a”, “new”, “sentence”) # vector with 3 elements that are characters
sent1 = paste(wrd,sep= “ ”)  # doesn’t quite work: sent1 is same as wrd itself
sent2 = paste(wrd[1],wrd[2],wrd[3],sep= “ ”)  #  “a new sentence”
wrd2 = paste(wrd, “!”, sep=“”) #  c(“a!”, “new!”, “sentence!”)
x = 1:3; wrd3 = paste(wrd, “_”,x,sep=“”)  # c(“a_1”, “new_2”, “sentence_3”)
comps=strsplit(sent2,split=“ ”) # break sent2 into component words; object comps is a list
comps[[1]] # first element (field) of the list comps; same as wrd
strsplit(sent2,split=NULL)[[1]] # c(“a”, “ ”, “n”, “e”, “w”, “ ”, “s”, “e”, “n”, “t”, “e”, “n”, “c”, “e”)
```

Helpful functions

```
objects( )   # lists all the objects you created
history(20) # last 20 lines of codes you typed in the current R session
?strsplit  # help on the function strsplit( )
example(strsplit) # runs the examples used in the R documentation of strsplit( )
args(strsplit) # displays argument names and corresponding default values of the function strsplit( )
??string # lists R functions with string as part of its name or ones that manipulate or use strings
getwd( ) # current working directory
save.image(file=“C:/Users/DP/mywork/lecture2.Rdata”)  # saves everything in file lecture2.Rdata
remove(y) # same as rm(y) ;  remove object y from current R session
```