

# 다변량 분석: R의 기초

Jinseog Kim

October 8, 2007

# 1 R의 기초

## 1.1 R의 시작

- Download the R software from
  - <http://www.r-project.org/> or
  - <http://bibs.snu.ac.kr/R/>.
- Install R software (R-2.5.0-win32.exe: Setup program (about 29 megabytes)).

## 1.2 R objects

vector,            matrix,            list  
data.frame,        factor,            function  
...

## 1.3 Indexing

R has three indexing constructs:

```
object [ arg1, ... , argn ]      # for vector, matrix, array
object [[ arg1, ... , argn ]]    # for list
object $ tag                      # for data.frame or named list
```

## 1.4 R operators

```
-      :Minus, can be unary or binary
+      :Plus, can be unary or binary
*      :Multiplication, binary
/      :Division, binary
%%     :Modulus, binary
<      :Less than, binary
>      :Greater than, binary
```

== :Equal to, binary  
>= :Greater than or equal to, binary  
<= :Less than or equal to, binary  
! :Unary not  
: :Sequence, binary (in model formulae: interaction)  
^ :Exponentiation, binary  
& :And, binary, vectorized && :And, binary, not vectorized  
| :Or, binary, vectorized || :Or, binary, not vectorized  
<- :Left assignment, binary,  
<<- : global assignment  
>- :Right assignment, binary  
= :Left assignment, binary  
\$ :List subset, binary

## 1.4.1 R operators:Example

`%*%` :Matrix product, binary

```
> a
      [,1] [,2]
[1,]    3    6
[2,]    4    8
> b
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> a%*%b
      [,1] [,2]
[1,]   15   33
[2,]   20   44
```

## 1.4.2 R operators:Example

`%o%` :Outer product, binary

```
> c(1,2) %o% c(3,4)
```

```
      [,1] [,2]
[1,]    3    4
[2,]    6    8
```

`%x%` :Kronecker product, binary

```
> a
```

```
      [,1] [,2]
[1,]    3    6
[2,]    4    8
```

```
> b
```

```
      [,1] [,2]
[1,]    1    0
[2,]    0    1
```

```
> b%x%a
```

```
      [,1] [,2] [,3] [,4]
[1,]    3    6    0    0
[2,]    4    8    0    0
[3,]    0    0    3    6
[4,]    0    0    4    8
```

## 1.5 Flow control

```
if ( cond ) expr  
if ( cond ) expr1 else expr2  
while ( cond ) expr  
repeat expr  
for ( var in list ) expr
```

Within the loop constructs (while, repeat, for), one may use

- break (to terminate the loop) and
- next (to skip to the next iteration).

## 1.6 R Functions: two simple examples

```
> name <- function(arg_1, arg_2, ...) expression
```

(예제)mile을 km로 바꾸는 프로그램.

```
miles.to.km <- function(miles) miles*8/5
```

```
> miles.to.km(175) # Approximate distance to Sydney, in miles  
[1] 280
```

만일 100, 200 300 miles를 kilometer로 바꾼다면

```
> miles.to.km(c(100,200,300))  
[1] 160 320 480
```

## 1.7 Common Useful Functions

```
print()      # Prints a single R object  
cat()        # Prints multiple objects, one after the other  
length()     # Number of elements in a vector or of a list  
mean()  
median()
```



```
range()
sum()
unique()      #Gives the vector of distinct values
diff()       # Replace a vector by the vector of first differences
              # diff(x) has one less element than x
sort()       # Sort elements into order, but omitting NAs
order()      # x[order(x)] orders elements of x, with NAs last
rev()        # reverse the order of vector elements
cumsum()
cumprod()
```

## 1.8 평균, 표준편차를 구하는 프로그램

```
mean.and.sd <- function(x=1:10) {
  av <- mean(x)
  sd <- sqrt(var(x))
}
```

```
c(mean=av, SD=sd)
}
```

```
> mean.and.sd()
      mean      SD
5.500000 3.027650
```

## 1.9 Reading data from files

If there is a text file named `houses.data` as follows:

	Price	Floor	Area	Rooms	Age	Cent.heat
01	52.00	111.0	830	5	6.2	no
02	54.75	128.0	710	5	7.5	no
03	57.50	101.0	1000	5	4.2	no
04	57.50	131.0	690	6	8.8	no

```
05  59.75    93.0    900    5    1.9    yes
...
```

The data frame may then be read as

```
> HousePrice <- read.table("houses.data", header=TRUE)
```

## 1.10 다변량정규분포

```
\library(MASS)
Sigma <- matrix(c(10,3,3,2),2,2)
Sigma
x<-mvrnorm(n=1000, rep(0, 2), Sigma)
var(X)
```