응용통계학: R 자료 타입

인하대학교 통계학과

이 경 재

03 March, 2019

R: types of data

- Scalar
- Vector
- Factor
- Matrix
- Array
- Data frame
- List

Scalar and vector

```
x = 2 # or x <- 2
x
```

```
## [1] 2
```

```
y = c(1,2,3,4) # or x = 1:4
y
```

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

```
y + x; y/x
```

```
## [1] 3 4 5 6
```

```
## [1] 0.5 1.0 1.5 2.0
```

Scalar and vector

```
y*y # Hadamard product (element-by-element)
```

```
## [1]  1  4  9 16
```

```
y%*%y # vector multiplication
```

```
##      [,1]
## [1,]   30
```

Scalar and vector

```r
x1 = seq(5)
x2 = seq(1,5)
x3 = seq(from=1, to=5, by=1)
x4 = seq(1,5,1)
x5 = seq(1,5,length=5)

x1;x2;x3;x4;x5
```

```
## [1] 1 2 3 4 5
```

```
## [1] 1 2 3 4 5
```

```
## [1] 1 2 3 4 5
```

```
## [1] 1 2 3 4 5
```

```
## [1] 1 2 3 4 5
```

Scalar and vector

```r
rep(1,5)
```

```
## [1] 1 1 1 1 1
```

```r
rep(c(1,2), 3)
```

```
## [1] 1 2 1 2 1 2
```

```r
AB = c("A", "B")
rep(AB, 3)
```

```
## [1] "A" "B" "A" "B" "A" "B"
```

```r
rep(AB, times=c(4,2))
```

```
## [1] "A" "A" "A" "A" "B" "B"
```

Scalar and vector

```
x = 1:4
x[1]
```

```
## [1] 1
```

```
x[1:3]
```

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

```
x[-4]
```

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

```
x[c(T,T,F,F)]
```

```
## [1] 1 2
```

Factor

```r
x = c("low", "med", "high", "med", "high")
xf = factor(x); xf
```

```
## [1] low  med  high med  high
## Levels: high low med
```

```r
as.numeric(xf)
```

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

```r
xf2 = factor(x, levels = c("low", "med", "high"))
xf2; as.numeric(xf2)
```

```
## [1] low  med  high med  high
## Levels: low med high
```

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

Factor

```r
# you can't assign a different levels of the factors!
xf[3]="Extreme"; xf
```

```
## Warning in `[<-.factor`(`*tmp*`, 3, value = "Extreme"): in
## level, NA generated
```

```
## [1] low  med  <NA> med  high
## Levels: high low med
```

```r
# but you can assign a different factor in the list
xf[3]="low"; xf
```

```
## [1] low  med  low  med  high
## Levels: high low med
```

Factor

```r
# It's also easy to change levels. Notice that
# I don't change the values themselves, just the levels.
levels(xf)
```

```
## [1] "high" "low"  "med"
```

```r
levels(xf)[1] <- "Extreme"
xf
```

```
## [1] low     med     low     med     Extreme
## Levels: Extreme low med
```

Matrix

```
A1 = matrix(1:9, nrow = 3, ncol = 3)
# or A1 = matrix(1:9, 3,3)
A1
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
A2 = matrix(1:9, nrow = 3, ncol = 3, byrow=T)
A2
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

Matrix

```
t(A1)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
A1 + A2
```

```
##      [,1] [,2] [,3]
## [1,]    2    6   10
## [2,]    6   10   14
## [3,]   10   14   18
```

Matrix

```
eng = c(60, 72, 57, 90, 95, 72)
math = c(75, 80, 92, 91, 87, 50)
score1 = cbind(eng, math); score1
```

```
##      eng math
## [1,]  60   75
## [2,]  72   80
## [3,]  57   92
## [4,]  90   91
## [5,]  95   87
## [6,]  72   50
```

```
score2 = rbind(eng, math); score2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## eng    60   72   57   90   95   72
## math   75   80   92   91   87   50
```

Matrix

```r
A = matrix(1, 3, 3); A
```

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

```r
I = diag(3)
B = 0.3*A + 0.7*I; B
```

```
##      [,1] [,2] [,3]
## [1,]  1.0  0.3  0.3
## [2,]  0.3  1.0  0.3
## [3,]  0.3  0.3  1.0
```

Matrix

```r
solve(B)
```

```
##            [,1]       [,2]       [,3]
## [1,]  1.1607143 -0.2678571 -0.2678571
## [2,] -0.2678571  1.1607143 -0.2678571
## [3,] -0.2678571 -0.2678571  1.1607143
```

```r
solve(B) %*% B
```

```
##               [,1]         [,2] [,3]
## [1,]  1.000000e+00 1.387779e-17    0
## [2,] -4.163336e-17 1.000000e+00    0
## [3,] -5.551115e-17 0.000000e+00    1
```

Array

```r
C = array(1:24, dim=c(3,4,2))
C
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]   13   16   19   22
## [2,]   14   17   20   23
## [3,]   15   18   21   24
```

Array

```
C[2,3,1]
```

```
## [1] 8
```

```
C[,2,1]
```

```
## [1] 4 5 6
```

Data frame

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Specie
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Data frame

```r
dim(iris)
```

```
## [1] 150   5
```

```r
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Wi
## [5] "Species"
```

```r
iris[1,]
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Specie
## 1          5.1         3.5          1.4         0.2  setosa
```

List

```r
mylist = list(Math=math, Eng=eng, Name="final")
mylist
```

```
## $Math
## [1] 75 80 92 91 87 50
##
## $Eng
## [1] 60 72 57 90 95 72
##
## $Name
## [1] "final"
```

List

```
mylist$Math; mylist[[1]]
```

```
## [1] 75 80 92 91 87 50
```

```
## [1] 75 80 92 91 87 50
```

```
mylist$Math[2]; mylist[[1]][2]
```

```
## [1] 80
```

```
## [1] 80
```

List

```
mylist2 = list(a=1:3, b=matrix(1:4, 2,2), c="example")
mylist2
```

```
## $a
## [1] 1 2 3
##
## $b
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
##
## $c
## [1] "example"
```