MDA102-Statistical methods

Lecture 4 – 29 June 2020
6.30AM – 8.30AM
Dr Sharon Varghese A

MISSION
CHRIST is a nurturing ground for an individual's holistic development to make effective contribution to the society in a dynamic environment

VISION
Excellence and Service

R objects - Data structure in R

An R object can be in any one of the following structure

- vector (can be defined by C() or vector() function in R)

```
x<-c(1,2,3) # Defines a vector of type nume
class(x)
[1] "numeric"
length(x)#length of a vector
[1] 3
x<-c('cat','dog') # Defines a vector of typ
class(x)
[1] "character"
```

R objects - Data structure in R - Contd…

```
x<-c(1,2,3)
x[1]
[1] 1
x[1:2]
[1] 1 2
```

R objects - Data structure in R - Contd…

Vector operations

```
x<-c(1,2,3)
y<-c(8,9,7)
x+y
[1]  9 11 10
x*y#dot product
[1]  8 18 21
```

R objects - Data structure in R - Contd…

Generating vector of sequences

```
x<-seq(1:10)#produces a sequence
10 with increment 1
x
 [1]  1  2  3  4  5  6  7  8  9
x<-seq(from=2,to=20,by=3)#produc
numbers from 2 to 20 with increm
x
[1]  2  5  8 11 14 17 20
```

R objects - Data structure in R - Contd…

Repeating vector with constants

```
x<-rep(8,4)#produces a vector of
x
[1] 8 8 8 8
x<-rep(c(4,5),6)#produces a vect
repeating 6 time
x
 [1] 4 5 4 5 4 5 4 5 4 5 4 5
x<-rep(c(2,3),each=2)#produces a
value in (2,3) repeating twice
```

R objects - Data structure in R - Contd…

Using all() and any()

```
x<-1:10
x
 [1]  1  2  3  4  5  6  7  8  9
any(x>8)
[1] TRUE
all(x>8)
[1] FALSE
```

- list (is a generic vector containing all types of objects and is defined u
- list is similar to python dictionary or struct in C program

```
x<-list(name='Alan', salary=75000)#name and
x
$name
[1] "Alan"

$salary
[1] 75000
```

```
x$salary#to call the value in ta
[1] 75000
x<-list("Alan",75000)# list with
x<-list(2,3)
x
[[1]]
[1] 2

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

```
a<-c(2,3)
b<-c('cat','dog')
c<-c(TRUE,FALSE)
x<-list(a,b,c)#combine three obj
x
[[1]]
[1] 2 3

[[2]]
[1] "cat" "dog"

[[3]]
[1]  TRUE FALSE
length(x)
[1] 3
```

List indexing

```
x<-list(name='Alan', salary=7500
x['salary']
$salary
[1] 75000
x$place<-'NY'
x
$name
[1] "Alan"

$salary
[1] 75000

$place
[1] "NY"
```

Matrix - extension of vector

```
m<-matrix(nrow = 2,ncol = 2)
m
     [,1] [,2]
[1,]   NA   NA
[2,]   NA   NA
dim(m)
[1] 2 2
m1<-matrix(c(1,2,3,4),2,2)
m1
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
m1<-matrix(c(1,2,3,4),2,2, byrow
elements of a vector by row
m1
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Subsetting matrix

```
m1[1,]#To get first row
[1] 1 2
m1[,1]#To get first col
[1] 1 3
```

```
m1[1:2,]#to get first to second
     [,1] [,2]
[1,]    1    2
[2,]    3    4
m1[1:2,1]#To get 1st column entr
row and
[1] 1 3
```

Replacing some elements

```
y<-matrix(c(1,2,3,4,5,6,7,8,9),3
dim(y)
[1] 3 3
y
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
y[c(1,3),]<-matrix(c(10,11,12,21
y
     [,1] [,2] [,3]
[1,]   10   12   22
[2,]    2    5    8
[3,]   11   21   23
```