

# Section 1: Data Structures

Sandeep Kumar

March 16, 2018

## Data Types

1. Logical
2. Integer
3. Numeric
4. Complex
5. Character

### 1. Logical (TRUE and FALSE)

```
4 == 4  
## [1] TRUE
```

### 2. Numeric

```
a <- 4  
a  
## [1] 4  
class (a)  
## [1] "numeric"  
c <- 3.5  
c  
## [1] 3.5  
class (c)  
## [1] "numeric"
```

### 3. Integer

```
b <- 4L  
b  
## [1] 4  
class (b)  
## [1] "integer"
```

## 4. Complex (Not covered)

## 5. Character

```
e <- 'Sandeep Kumar'
e

## [1] "Sandeep Kumar"

class(e)

## [1] "character"
```

### Summary: Five Common Data Types

1. Logical (e.g., TRUE, FALSE)
2. integer (e.g., 2L, as.integer(3))
3. numeric (real or decimal) (e.g, 2, 2.0, pi)
4. complex (e.g, 1 + 0i, 1 + 4i)
5. character (e.g. "Sandeep Kumar", 'Paul')

## Data Structures

### Vector, Factor, List, Matrix and Data Frame

#### 1. Vectors

Create a vector using `c()` command.

```
num.a <- c(1,2,4,6,7)
num.a

## [1] 1 2 4 6 7

class(num.a)

## [1] "numeric"
```

Check the class of vector num.a. It is numeric so `is.numeric` will return TRUE, and `is.logical` will return FALSE

```
is.numeric(num.a)

## [1] TRUE

is.logical(num.a)

## [1] FALSE

num.b <- c("Paul", "John", "Sandeep", "Ali")
num.b

## [1] "Paul"      "John"      "Sandeep"   "Ali"
```

```

class(num.b)
## [1] "character"

is.character(num.b)
## [1] TRUE

num.c <- c(TRUE, F, FALSE, T)
num.c

## [1] TRUE FALSE FALSE TRUE

class(num.c)
## [1] "logical"

is.logical(num.c)
## [1] TRUE

```

## 2. Factors

Factor changes the characters to categories

```

airports <- c('DEL', 'CAL', 'NYC', 'CAL', 'DEL', 'DEL')
summary(airports)

##      Length      Class      Mode 
##         6 character character

# barplot(summary(airports))

```

**barplot() commands an error. In the above case.**

Let's change items in the vector to factors. We will be able to plot a barplot by doing so.

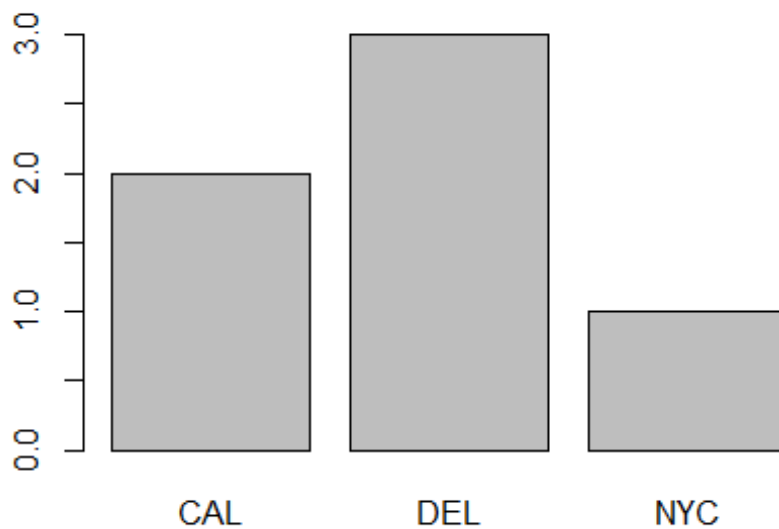
```

airports <- c('DEL', 'CAL', 'NYC', 'CAL', 'DEL', 'DEL')
airports.fact <- factor(airports)
summary(airports.fact)

## CAL DEL NYC
##   2   3   1

barplot(summary(airports.fact))

```



### 3. List

Lists are used to place number of items in a bundle.

```
a <- c(1,4,6)
b <- c('Red', 'Green')
c <- "Welcome"
```

```
my.list <- list(a,b,c)
```

### 4. Matrix

Matrix has two dimensional data of similar type

```
stu.hours <- c(20,24,46,62,22,37,45,27,65,23)
stu.marks <- c(40,55,69,83,27,44,61,33,71,37)
```

*# Matrix function using just stu.hours*

```
stu.matrix <- matrix(stu.hours)
stu.matrix
```

```
##      [,1]
## [1,]  20
## [2,]  24
## [3,]  46
## [4,]  62
## [5,]  22
## [6,]  37
```

```
## [7,] 45
## [8,] 27
## [9,] 65
## [10,] 23
```

**Join two columns (stu.hours,stu.marks) as stu.data**

```
stu.data <- c(stu.hours,stu.marks)
stu.matrix <- matrix(stu.data)
stu.matrix
```

```
##      [,1]
## [1,] 20
## [2,] 24
## [3,] 46
## [4,] 62
## [5,] 22
## [6,] 37
## [7,] 45
## [8,] 27
## [9,] 65
## [10,] 23
## [11,] 40
## [12,] 55
## [13,] 69
## [14,] 83
## [15,] 27
## [16,] 44
## [17,] 61
## [18,] 33
## [19,] 71
## [20,] 37
```

**Above provide 20 values in a single column instead of two separate columns. Add the argument nrow to create 10 rows**

```
stu.matrix <- matrix(stu.data, nrow = 10)
stu.matrix
```

```
##      [,1] [,2]
## [1,] 20  40
## [2,] 24  55
## [3,] 46  69
## [4,] 62  83
## [5,] 22  27
## [6,] 37  44
## [7,] 45  61
## [8,] 27  33
## [9,] 65  71
## [10,] 23  37
```

## 5. Data Frame

Here you can have mix of data types, unlike matrix.

Let's create 4 vectors which will form 4 columns of the data frame.

```
stu.name <- c('Mr1', 'Ms2', 'Ms3', 'Mr4', 'Ms5', 'Ms6', 'Mr7', 'Ms8', 'Mr9',  
'Mr10')  
stu.hours <- c(20,24,46,62,22,37,45,27,65,23)  
stu.marks <- c(40,55,69,83,27,44,61,33,71,37)  
stu.male <- c(T,F,F,T,F,F,T,F,T,T)
```

Combine these four vectors as a data frame.

```
stu.data <- data.frame(stu.name, stu.hours, stu.marks, stu.male)  
stu.data
```

##	stu.name	stu.hours	stu.marks	stu.male
## 1	Mr1	20	40	TRUE
## 2	Ms2	24	55	FALSE
## 3	Ms3	46	69	FALSE
## 4	Mr4	62	83	TRUE
## 5	Ms5	22	27	FALSE
## 6	Ms6	37	44	FALSE
## 7	Mr7	45	61	TRUE
## 8	Ms8	27	33	FALSE
## 9	Mr9	65	71	TRUE
## 10	Mr10	23	37	TRUE