UNIT II

1.3. Introduction to various Data Types:

In Analytics the data is classified as Quantitative(numeric) and Qualitative(Character/Factor) on very broad level.

- Numeric Data: - It includes 0~9, "." and "- ve" sign.
- Character Data: - Everything except Numeric data type is Character. For Example, Names, Gender etc.

For Example, "1,2,3…" are Quantitative Data while "Good", "Bad" etc. are Qualitative Data. We can convert Qualitative Data into Quantitative Data using Ordinal Values. For Example, "Good" can be rated as 9 while "Average" can be rated as 5 and "Bad" can be rated as 0.

Table 1. List of Data types in R

Data Type		Verify

Logical	TRUE , FALSE	`v <- TRUE`
		`print(class(v))`
		`[1] "logical"`

Numeric	12.3, 5, 999	`v <- 23.5`
		`print(class(v))`
		`[1] "numeric"`

Integer	2L, 34L, 0L	`v <- 2L`
		`print(class(v))`
		`[1] "integer"`

Complex	3 + 2i	`v <- 2+5i`
		`print(class(v))`
		`[1] "complex"`

Character	'a' , '"good", "TRUE", '23.4'	`v <- "TRUE"`
		`print(class(v))`
		`[1] "character"`

Raw	"Hello" is stored as 48 65 6c 6c 6f	`v <- charToRaw("Hello")`
		`print(class(v))`
		`[1] "raw"`

Create new variables using already available variables:

Example:

mydata$sum <- mydata$x1 + mydata$x2
New variable is created using two already available variables.
Modifying existing variable: Rename the existing variable by using rename() function. For examples,
mydata<- rename(mydata, c(oldname="newname"))

1.3.1. Vectors:

Vector is the most common data structure in R. Vectors must be homogeneous i.e, the type of data in a given vector must all be the same. Vectors can be numeric, logical, or character. If a vector is mix data types then R forces (**coerces**, if you will) the data into one mode.
Creating a vector:
To create a vector , "concatenate" a list of numbers together to form a vector.
x < - c(1, 6, 4, 10, -2) ## c() to concatenate elements
my.vector<- 1:24 ## a numeric vector with 1 to 24 numbers **List of built-in functions to get useful summaries on vectors:**

Example1:

> sum(x) ## sums the values in the vector
> length(x) ## produces the number of values in the vector, ie its length
> mean(x) ## the average (mean)
> var(x) ## the sample variance of the values in the vector
> sd(x) ## the sample standard deviation of the values in the vector (square root of the sample variance)
> max(x) ## the largest value in the vector

Example2:

linkedin <- c(16, 9, 13, 5, 2, 17, 14)
> last <- tail(linkedin, 1)
> last
[1] 14
> **# Is last under 5 or above 10?**
> # Is last between 15 (exclusive) and 20 (inclusive)?
> **# Is last between 0 and 5 or between 10 and 15?**
> **(last > 0 | last < 5)**
[1] TRUE
> (last > 0 & last < 5)
[1] FALSE
> (last > 10 & last < 15)
[1] TRUE

Example3: Following are some other possibilities to create vectors

> x <- 1:10
> y <- seq(10) #Create a sequence
> z <- rep(1,10) #Create a repetitive pattern
> x
[1] 1 2 3 4 5 6 7 8 9 10
> y
[1] 1 2 3 4 5 6 7 8 9 10
> z
[1] 1 1 1 1 1 1 1 1 1 1

Adding elements to vector:

```
> x <- c(x, 11:15)
>x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Vector Arithmetic:
> x <- c(1:10)
> x
[1] 1 2 3 4 5 6 7 8 9 10
> y <- 10
> x + y
[1] 11 12 13 14 15 16 17 18 19 20
> 2 + 3 * x #Note the order of operations
[1] 5 8 11 14 17 20 23 26 29 32
> (2 + 3) * x #See the difference
[1] 5 10 15 20 25 30 35 40 45 50
> sqrt(x) #Square roots
[1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
[9] 3.000000 3.162278
> x %% 4 #This is the integer divide (modulo) operation
[1] 1 2 3 0 1 2 3 0 1 2
> y <- 3 + 2i #R does complex numbers
> re(y) #The real part of the complex number
[1] 3
> im(y) #The imaginary part of the complex number
[1] 2
> x * y
```