TMC 204
Statistical Data Analysis with R
Unit 4
Objects and Data Types

Presented By : Aditya Joshi
Asst. Professor
Department of Computer Application
Graphic Era Deemed to be University
31-03-2020

Basic Data Types and Data Structures in R

Everything in R is an object.

you'll need a strong understanding of the basic data types and data structures and how to operate on them.

Data structures are very important to understand because these are the objects you will manipulate on a day-to-day basis in R.

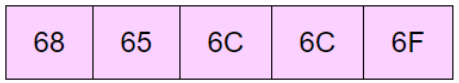
R has 6 basic data types.

- **character**
- **numeric (real or decimal)**
- **integer**
- **logical**
- **complex**

In addition to this there is a **raw** data type

Elements of these data types may be combined to form data structures.

such as atomic **vectors**. When we call a vector *atomic*, we mean that the vector only holds data of a single data type.

Data Type	Example	Description
character	‘m’, “hello”	Character and strings
numeric	2, 45.9, 3782	Number of all kinds
integer	9L, 779L	Eplicitly integers the L tells R to store this as an integer
logical	TRUE, FALSE	boolean values
Complex	8+9i	Real Value + Complex Value
Raw	 Is the value for **hello**	Any data is stored as raw bytes

These are **Atomic Vectors – formed using R Atomic Data Types**

When data type is **Raw**, user has to know the format or protocol of the data.

R provides many functions to examine features of vectors and other objects, for example

class() - what kind of object is it (high-level)?

typeof() - what is the object's data type (low-level)?

length() - how long is it? What about two dimensional objects?

attributes() - does it have any metadata?

Below are examples of atomic character vectors, numeric vectors, integer vectors, etc.

```
> x<-TRUE
> class(x)
[1] "logical“

> x<-67.54
> class(x)
[1] "numeric“

> x<-63L
> class(x)
[1] "integer“

> x <- 6 + 4i
> class(x)
[1] "complex"
```

```
> x <- "hello"
> class(x)
[1] "character“

> x <- charToRaw("hello")
> class(x)
[1] "raw"
```

Example:

```
> x<-"dataset"
> typeof(x)
[1] "character“

> attributes(x)
NULL

> y<-1:10
> y
 [1]  1  2  3  4  5  6  7  8  9 10
```

```
> typeof(y)
[1] "integer“
> length(y)
[1] 10

> z<-as.numeric(y)
> z
 [1]  1  2  3  4  5  6  7  8  9 10
> typeof(z)
[1] "double"
```

R has many data structures. These include

- atomic vector
- list
- matrix
- data frame
- Factors

Creating, Accessing, Sorting data frames, Extracting, Combining, Merging, reshaping data frames

Source: github, Beginning R Dr. Mark Gardner