

```
---
title: "R Data Types and Data Structures"
output: html_notebook
---
```

**\*\*Everything in R is an object\*\***

```
```{r}
# Create an R object
my_obj = 48
\`\`
```

**\*\*Basic data types in R\*\*:**

1. Character
2. Numeric (real or decimal)
3. Integer
4. Logical
5. Complex

```
```{r}
# Create objects of simple data types
my_char = 'a'
print(my_char)
```

```
my_int = 10L
print(my_int)
```

```
my_numeric = 10.09
print(my_numeric)
```

```
my_logic = TRUE
print(my_logic)
\`\`
```

**\*\*Data structures in R\*\*:**

1. Atomic vector
2. List
3. Matrix
4. Data frame
5. Factor

We will look at atomic vectors, also referred simply to as vectors, first. A vector is a collection of objects of the same data type.

```
```{r}
# Create a vector
my_vec1 = c(1L, 2L, 3L)
my_vec2 = 1:3
my_vec3 = seq(from = 1, to = 3, by = 1)
print(my_vec1)
print(my_vec2)
print(my_vec3)
```

```
# Functions on objects
length(my_vec1)
class(my_vec1)
class(my_vec2)
class(my_vec3)
typeof(my_vec3)
str(my_vec3)
```

```
# Access elements of a vector
my_vec1[1]
```

```
# Modify element of a vector
my_vec1[1] = 10L
```

```

print(my_vec1)

# Missing data
my_vec4 = c(1, 2, NA, 4)
print(my_vec4)
is.na(my_vec4)
anyNA(my_vec4)

# Special values
1/0
0/0
```

```

A list is a special type of a vector which can contain objects of possibly different data types.

```

```{r}
# Create an empty list
my_list1 = list(5)
print(my_list1)
class(my_list1)

# Create a list with values
my_list2 = list(1, 'Name', c('a', 'b', 'c'))
print(my_list2)
class(my_list2)

# Assign names to slots of list
names(my_list2) = c('first', 'second', 'third')
str(my_list2)

# Coerce a vector into a list
```

```

Accessing and modifying elements of a list

```

```{r}
# Access elements of a list
my_list2[1]
my_list2[2]
my_list2[3]

my_list2$first
my_list2$second
my_list2$third

# Modify elements of a list
my_list2[1] = 10
print(my_list2)
```

```

A matrix is an atomic vector with one or two dimensions.

```

```{r}
# Create a matrix
my_matrix1 = matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)
print(my_matrix1)

my_matrix2 = matrix(c(1,2,3,4,5,6), nrow = 2, ncol = 3, byrow = TRUE)
print(my_matrix2)

# Assign row and column names
rownames(my_matrix2) = c('row1', 'row2')
colnames(my_matrix2) = c('col1', 'col2', 'col3')
str(my_matrix2)

# Access elements of a matrix
my_matrix2[1, 2]

```

```
```
```

A data frame is a list of lists with each sublist of same length; think of it as a rectangular list. A data frame is typically used to store data that are read from text/CSV files by retaining the underlying structure such as row names, column names etc. A data frame can also be created manually.

```
```{r}
# Create a dataframe manually
ID = c('A', 'B', 'C')
Age = c(21, 22, 20)
Height = c(150, 160, 170)
sData = data.frame(ID, Age, Height)
print(sData)

# Assign names to the rows and columns of the data frame
rownames(sData) = c('Ajith', 'John', 'Bob')
colnames(sData) = c('ID', 'Age', 'Height')

print(sData) # Prints the DataFrame sData
```
```

In-built functions on data frame

```
```{r}
# Structure of the data frame
str(sData)

# Print 1st five rows
head(sData, 2)

# Print last five rows
tail(sData, 2)

# Get the dimension of the data frame
dim(sData)

# Number of rows in the data frame
nrow(sData)

# Number of columns in the data frame
ncol(sData)
```
```

Accessing elements of a data frame

```
```{r}
# Access a particular column
sData$Age
sData[['Age']]
sData['Age']

# Access a particular row
sData['John', ]

# Access multiple columns
sData[c('ID', 'Age')]
```
```

A factor is a vector that can contain only predefined values, and is used to store categorical data.

```
```{r}
# Create a factor for storing a list of genders
gender = factor(c('Male', 'Male', 'Female', 'Female'))
print(gender)

# In-built functions on factors
```

```
levels(gender)
```

```
# Modify a gender  
gender[1] = 'Female'  
print(gender)  
\\
```