R DATA TYPES

Basic data types in R are

1. Numeric
2. Vectors
3. Lists
4. Matrices
5. Arrays
6. Factors and
7. Data Frames

There is no declaration of variables in R.

1. **Numeric variables**
   Decimal values are numeric in R. Integers are also saved as integers and not numeric.

```
a=10.6
a
b=10
b
class(a)
class(b)
is.integer(a)
is.integer(b)
```

2. Integers

To create an integer we use ***as.integer*** function. To check whether a variable is integer, we can use ***is.integer(variable_name)***.

```
p=as.integer(3)
p
class (p)
is.integer (p)
```

To coerce/convert a numeric value to integer we can use the same ***as.integer*** function.

Boolean values can also be coerced to 1 and 0 using ***as.integer*** function.

```
x=as.integer (3)
x
class (x)
is.integer(x)
y=as.integer(3.23)
y
z=as.integer("7.81")
Z
as.integer(True)
as.integer (TRUE)
```

as.integer (FALSE)

3. Complex

The complex number will be of the form a+bi, where a is the real part and b is the imaginary part.

```
x=4+6i
x
class (x)
is.complex(x)
y=as.complex (3)
y
```

4. Logical

Logical values are usually created via c**omparison between variables.**

Logical values are TRUE and FALSE. True can be coerced to 1 and False can be coerced to 0.

```
x=1; y=2;
z=x>y;
z
class (z)
as.logical (1)
as.logical (0)
```

5. Character

i.String values are stored as Character objects in R.

ii.Conversion to character happens with the as.character() function.

iii.Single and double quotes can be used to represent strings.

```
x=”abc”
y=as.character(7.8)
x
y
class (y)
```

VECTORS

Vector is a basic data structure in R. It consists of elements of the same type.

The data types can be numeric, integers, logical, double, complex or raw. Members of a vector are called ***components.***

Type of vector is found out using **typeof()** function and length by **length()**

1.Create Vectors using :

To create a vector of continuous numbers use a colon operator.

x <-7
x
y=2.5:10.5
y
x = 2.5:4.5

2. Create vectors using c() function

i). A vector must have elements of the same type.

ii). This function will coerce elements to the same type, if they are different.

iii). Coercion is from lower type to higher type. From logical to double to character.

x <- c(1,2,5,TRUE)
x
typeof(x)
x<-c(1,TRUE,"hello")
typeof(x)
length(x)

3. Create vectors using sequence (seq) operator

x <- seq(2,3, by=0.2)
x
y <- seg(1,5, length=3)
y

4. Combining Vectors

Vectors can be combined by c function.
When they are of different datatype, the lower data type is converted to higher data type to maintain the same datatype for all the vector's components.

a = c (1,2,3,4)
b= c (a,b,c,d)
c(a,b)
c

4. Accessing Vector Elements

Elements of a vector can be accessed using vector indexing. The vector used for indexing can be integer, logical or character.

a. Using integer vector as index

In R, index starts from 1 rather than 0 like other programming languages. When we access a vector, we get a sliced vector. We can

use negative indexes. This returns all the members except the specified member. If real numbers are used for indexing, it is truncated to integers. If the index is out of range, missing values will be reported.

s = c (“a”, “b”, “c”, “d”, “e”)
s[1]
s[-2]
s[20]
s[c(2,3)]
s[c(2,3,1,1)]
s[c(3,2,10)]
S[2:4]

b. Logical Vector as Index

When we use logical indexing , the position where the logical vector is True is returned.

s = c(1,2,3,-10,-20,-30)
s[c(TRUE,FALSE,FALSE,FALSE,FALSE,FALSE)]
s[s<0]
s[s>0]

c. Character Vector as Index

This indexing is useful for named vectors. We can name each member of a vector.

v= c(1,2,3,4)
v
names(v) = c("First", "Second", "Third", "Fourth")
names(v)
v["second"]
v[c("First","Second","Third")]

5. Modifying Vector elements

Modifying elements happens through assignment operators.

x=c(10,20,30,40,50)
x
x[2] <-15
x[x<30]<-5
x<-x[1:4]
x

6. **Deleting Vectors**

To delete a vector assign Null to it.

x=c(10,20,30,40,50)
x

```
x <- NULL
x
```

6. Vector Arithmetic

Two vectors can be added, subtracted, multiplied, divided, perform modulo and exponentiation.

```
x = c(10,20,30,40)
y=c(10,20,30,40)
z<-x+y
z
```

7. Vector Recycling

When vectors are of different size, the short one is repeated or recycled. Until lengths are equal.

```
x=c(10,20,30)
y=c(1)
z<-x+y
y=c(1,2)
z=x+y
Z
```

8. Vector Element Sorting

Sorting is performed using the sort function. By default it sorts in the ascending order. For descending order, we should pass decreasing=TRUE. Vectors are immutable. Original vector remains unchanged.

x = c(3,2,5,1,7,8,9,2,2)
sort (x)
sort(x,decreasing=TRUE)
X
Animals = c("lion","tiger","cat")
sort(animals, decreasing=TRUE)

9. Reading a Vector

To read an input from a user we can use the **readline() function.** This will return **a single character vector.**

If we want integers, we should use as.integer() and other similar functions like as.double(), as.logical(), as.complex()

my.name <- readline(prompt="Enter your name")
my.age <- readline(prompt="Enter age:")
my.age<-as.integer(my.age)
my.response<-readline(prompt"Have you registered for the event already?")
my.response<-as.logical(my.response)

Each readline should be executed line by line. Multiple read lines will not work.

R OPERATORS

The list of operators in R are

1. Arithmetic
2. Relational
3. Logical
4. Assignment
5. Miscellaneous Operators

Arithmetic Operators

These operators are used to perform mathematical operations.

+

-

*

/

%% Gives remainder of the first vector divided with second vector

%/% Result of division of first vector by secon vector. Only quotient will be taken

^

```
a=c(1,2,3,4)
b=c(4,5,6,7)
cat("Sum=",(a+b),"\n")
cat("Quotient=",(a/b),"\n")
cat("Remainder=",(a%%b),"\n")
cat("Integer Division=",(a%/%),"\n")
cat("Exponent=",(a^b),"\n")
```

RELATIONAL OPERATORS

Used to compare between vales. Each element of first vector is compared to second vector. Result of comparison is a boolean value.

```
a<-c(10,20,30,40)
b<-c(24,2,7,5)
cat(a,"Less than ",b,(a<b),"\n")
cat(a,"Equals ",b(a==b),"\n")
```

LOGICAL OPERATORS

Logical operators are applicable to type logical, numeric or complex. All numbers greater than 1 are considered as Logical values TRUE. Each element of the first vector is compared to the second vector. Result is a boolean vector.

Logical operators are

! Logical Not
& Element-wise Logical AND
&& Logical AND
| Element–wise Logical OR
|| Logical OR

ASSIGNMENT OPERATORS
‘<-, <<-, =’ Leftward Assignment
->,->> Rightward assignment

MISCELLANEOUS OPERATORS
: creates a range of numbers
%in% used to check whether element belongs to a vector
%*% multiply matrix to a transpose