Arithmetic, Logical and Matrix operations in R

R supports all the basic arithmetic operation, the first one is assignment operator. You can use either = or the back arrow <- to assign a value to be variable and standard addition, subtraction, multiplication, division, integer division and remainder operations are also available in R. In R back arrow <- is only the valid assignment operator whereas,as an R studio both = and back arrow R proper assignment operators.

Let us look at the hierarchy of operations while performing the arithmetic operations in R. So, it is similar to our normal BODMAS rule with bracket has the first importance exponent has the second priority and followed by division, multiplication, addition and subtraction.

we have standard logical operations such as <, < =, >, > =, = and so on.

MATRICES:

creating matrices:

```
> A=matrix()
> A
     [,1]
[1,]   NA
> A=matrix(c(1,2,3,4,5,6,7,8,9))
> A
      [,1]
 [1,]    1
 [2,]    2
 [3,]    3
 [4,]    4
 [5,]    5
 [6,]    6
 [7,]    7
 [8,]    8
 [9,]    9

>
> A=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3,byrow=TRUE)
> A
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9


> matrix(3,3,4)
     [,1] [,2] [,3] [,4]
[1,]    3    3    3    3
[2,]    3    3    3    3
[3,]    3    3    3    3
```

you want to fill all the rows and columns with the
element 3 which is a matrix which contains 3 rows and 4 columns. So, you have specified 3, 3 and 4 when you do that you will get the matrix printed like this.

```
> diag(1,3,3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1

>

> dim(A)
[1] 3 3
> nrow(A)
[1] 3
> ncol(A)
[1] 3
> length(A)
[1] 9

>
```

```
> colnames<-c("a","b","c")
> row.names(A)<-c("d","e","f")
> A
  [,1] [,2] [,3]
d    1    2    3
e    4    5    6
f    7    8    9
> colnames(A)<-c("a","b","c")
> A
  a b c
d 1 2 3
e 4 5 6
f 7 8 9

>

> A[1,2]
[1] 2
> A[,1]
d e f
1 4 7

>

> A[2,]
a b c
4 5 6

>

> A[,-2]
  a c
d 1 3
e 4 6
f 7 9

>

> A[-2,]
  a b c
```

```
d 1 2 3
f 7 8 9

>
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> 10:1
 [1] 10  9  8  7  6  5  4  3  2  1

>
> A[1:3,1:2]
  a b
d 1 2
e 4 5
f 7 8
> A[1:3,-2]
  a c
d 1 3
e 4 6
f 7 9
> A[,1:2]
  a b
d 1 2
e 4 5
f 7 8

>
> A[c(1:3),1:2]
  a b
d 1 2
e 4 5
f 7 8
> A[c(1:3),c(1:2)]
  a b
d 1 2
e 4 5
f 7 8
```

Matrix concatenation refers to merging of rows or columns to an existing matrix. If you want to add a row to the existing matrix you can do so by using R bind command. If you want to add a column to a matrix you can do so by using c bind command. So, one thing you have to keep in mind is you have to make sure the consistency of dimensions before you do this matrix concatenation.

```
> B<-matrix(c(10,11,12),nrow=1)
> B
     [,1] [,2] [,3]
[1,]   10   11   12
> C=rbind(A,B)
> C
   a  b  c
d  1  2  3
e  4  5  6
f  7  8  9
  10 11 12

> B<-matrix(c(10,11,12))
> B
     [,1]
[1,]   10
[2,]   11
```

```
[3,]   12
> C=cbind(A,B)
> C
  a b c
d 1 2 3 10
e 4 5 6 11
f 7 8 9 12

>
```

let us see how to do algebraic operations on matrices such as addition, subtraction, multiplication and matrix division in R.

```
> B<-matrix(c(3,1,3,5,4,6,7,2,5),nrow=3,ncol=3,byrow =TRUE)
> B
     [,1] [,2] [,3]
[1,]    3    1    3
[2,]    5    4    6
[3,]    7    2    5
> A+B
   a  b  c
d  4  3  6
e  9  9 12
f 14 10 14
> A-B
   a b c
d -2 1 0
e -1 1 0
f  0 6 4
> A/B
          a    b   c
d 0.3333333 2.00 1.0
e 0.8000000 1.25 1.0
f 1.0000000 4.00 1.8
```