

```
x<-c(1,2,3,4)
class(x)
```

```
## [1] "numeric"
```

```
gender<-c("male","female")
gender
```

```
## [1] "male" "female"
```

```
class(gender)
```

```
## [1] "character"
```

```
2:7
```

```
## [1] 2 3 4 5 6 7
```

```
class(2:7)
```

```
## [1] "integer"
```

```
seq(from=1, to = 7, by=1/3)
```

```
## [1] 1.000000 1.333333 1.666667 2.000000 2.333333 2.666667 3.000000
## [8] 3.333333 3.666667 4.000000 4.333333 4.666667 5.000000 5.333333
## [15] 5.666667 6.000000 6.333333 6.666667 7.000000
```

```
class(seq(from=1, to = 7, by=1/3))
```

```
## [1] "numeric"
```

```
rep(1,times=10)
```

```
## [1] 1 1 1 1 1 1 1 1 1 1
```

```
class(rep(1,times=10))
```

```
## [1] "numeric"
```

```
rep(1:3,times=5)
```

```
## [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
rep("martin",times=5)
```

```
## [1] "martin" "martin" "martin" "martin" "martin"
```

```
rep(seq(from=2,to=5,by=0.5),times=5)
```

```
## [1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0 2.0 2.5 3.0 3.5 4.0 4.5 5.0 2.0 2.5 3.0  
## [18] 3.5 4.0 4.5 5.0 2.0 2.5 3.0 3.5 4.0 4.5 5.0 2.0 2.5 3.0 3.5 4.0 4.5  
## [35] 5.0
```

```
rep(c("a","b"),times=5)
```

```
## [1] "a" "b" "a" "b" "a" "b" "a" "b" "a" "b"
```

```
x<-1:5  
x
```

```
## [1] 1 2 3 4 5
```

```
y<-c(1,3,5,7,9)  
y
```

```
## [1] 1 3 5 7 9
```

```
x+5
```

```
## [1] 6 7 8 9 10
```

```
x-5
```

```
## [1] -4 -3 -2 -1 0
```

```
x*10
```

```
## [1] 10 20 30 40 50
```

```
x/10
```

```
## [1] 0.1 0.2 0.3 0.4 0.5
```

```
x
```

```
## [1] 1 2 3 4 5
```

#if two vectors are of the same length, we may add/subtract/mul/div corresponding elements

```
x<-1:5  
y<-c(1,3,5,7,9)  
class(x)
```

```
## [1] "integer"
```

```
class(y)
```

```
## [1] "numeric"
```

#There are multiple classes that are grouped together as "numeric" classes, the 2 most common of which are double (for double precision floating point numbers) and integer. R will automatically convert between the numeric classes when needed, so for the most part it does not matter to the casual user whether the number 3 is currently stored as an integer or as a double. Most math is done using double precision, so that is often the default storage.

```
x*y
```

```
## [1] 1 6 15 28 45
```

```
x+y
```

```
## [1] 2 5 8 11 14
```

```
x-y
```

```
## [1] 0 -1 -2 -3 -4
```

```
x[3]
```

```
## [1] 3
```

```
y[-3]
```

```
## [1] 1 3 7 9
```

```
# y[-3] will print y except third element ie 5  
y[1:3]
```

```
## [1] 1 3 5
```

```
y[c(1,5)]#just 1 st and 5th element of y
```

```
## [1] 1 9
```

```
y[-c(1,5)]#except 1st and 5 th element of y
```

```
## [1] 3 5 7
```

```
y[y<6]
```

```
## [1] 1 3 5
```

```
mat<- matrix(c(1:9),nrow=3,byrow = T)  
mat
```

```
##      [,1] [,2] [,3]  
## [1,]    1    2    3  
## [2,]    4    5    6  
## [3,]    7    8    9
```

```
mat<- matrix(c(1:9),nrow=3,byrow = FALSE)  
mat
```

```
##      [,1] [,2] [,3]  
## [1,]    1    4    7  
## [2,]    2    5    8  
## [3,]    3    6    9
```

```
mat2<- matrix(c(1:9),nrow=3,byrow = F)  
mat2
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
mat3<- matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,byrow = T)
mat3
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
mat3[1,2]
```

```
## [1] 2
```

```
mat3[c(1,3),2]  #row 1 and 3 and column 2
```

```
## [1] 2 8
```

```
class(c(1,2))
```

```
## [1] "numeric"
```

```
mat3[2,]  # row 2 all columns
```

```
## [1] 4 5 6
```

```
mat3*10
```

```
##      [,1] [,2] [,3]
## [1,]   10   20   30
## [2,]   40   50   60
## [3,]   70   80   90
```