

# **Introduction to R Software**

**Basics of Calculations**

**:::**

**Functions and Matrices**

**Shalabh**

**Department of Mathematics and Statistics**

**Indian Institute of Technology Kanpur**

# Functions

- **Functions are a bunch of commands grouped together in a sensible unit**
- **Functions take input arguments, do calculations (or make some graphics, call other functions) and produce some output and return a result in a variable. The returned variable can be a complex construct, like a list.**

# Functions

## Syntax

```
Name <- function(Argument1, Argument2, ...)  
  {  
    expression  
  }
```

where **expression** is a single command or a group of commands

## **Function arguments with description and default values**

- **Function arguments can be given a meaningful name**
- **Function arguments can be set to default values**
- **Functions can have the special argument '...'**

# Functions (Single variable)

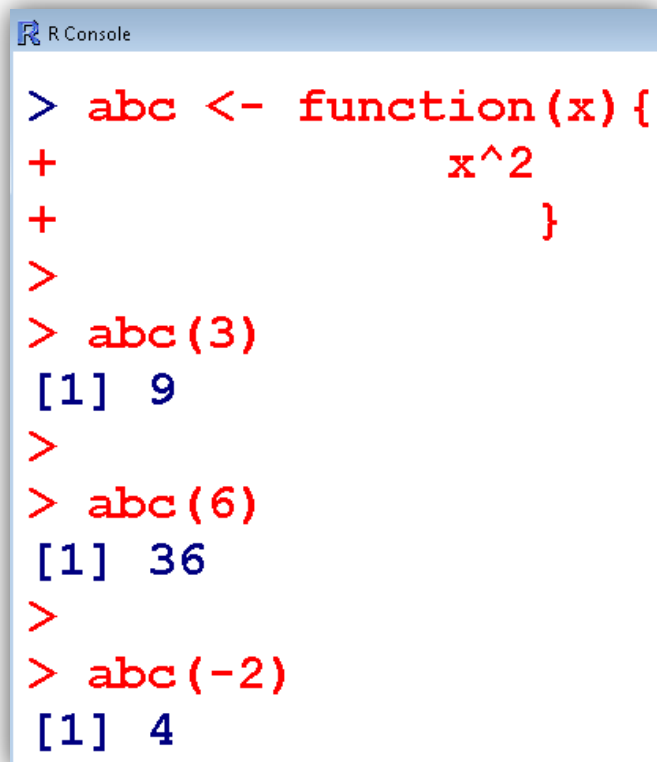
The sign `<-` is furthermore used for defining functions:

```
> abc <- function(x) {  
    x^2  
}
```

```
> abc(3)  
[1] 9
```

```
> abc(6)  
[1] 36
```

```
> abc(-2)  
[1] 4
```



The screenshot shows an R Console window with the following text:

```
> abc <- function(x) {  
+     x^2  
+ }  
>  
> abc(3)  
[1] 9  
>  
> abc(6)  
[1] 36  
>  
> abc(-2)  
[1] 4
```

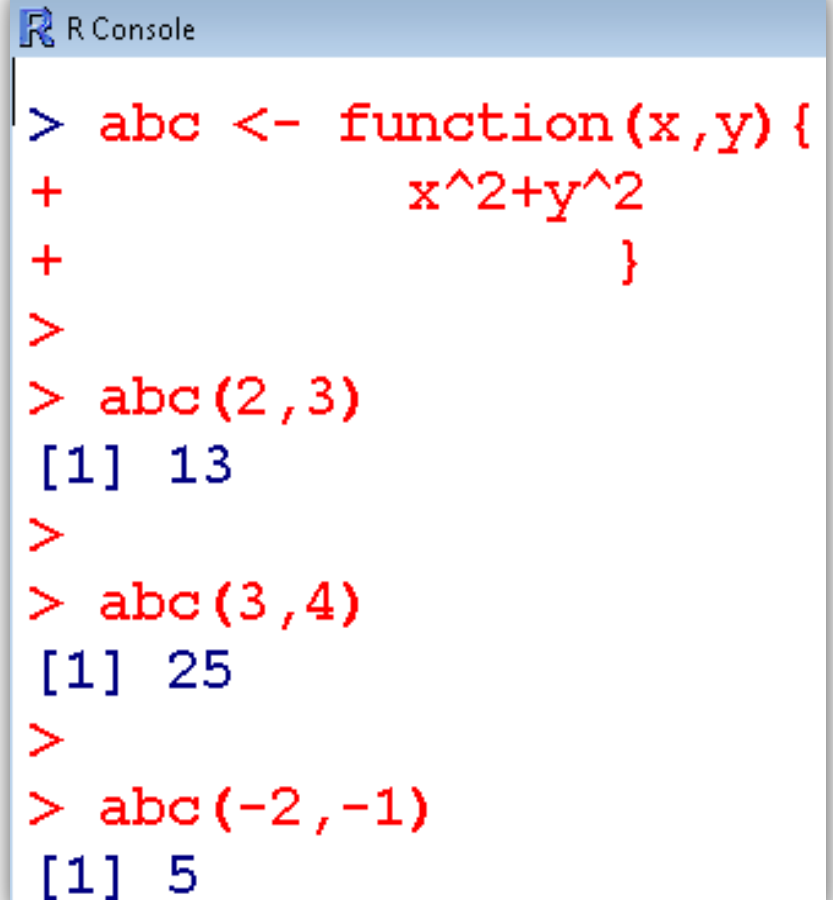
# Functions (Two variables)

```
> abc <- function(x,y) {  
  x^2+y^2  
}
```

```
> abc(2,3)  
[1] 13
```

```
> abc(3,4)  
[1] 25
```

```
> abc(-2,-1)  
[1] 5
```



R Console

```
> abc <- function(x,y) {  
+   x^2+y^2  
+ }  
>  
> abc(2,3)  
[1] 13  
>  
> abc(3,4)  
[1] 25  
>  
> abc(-2,-1)  
[1] 5
```

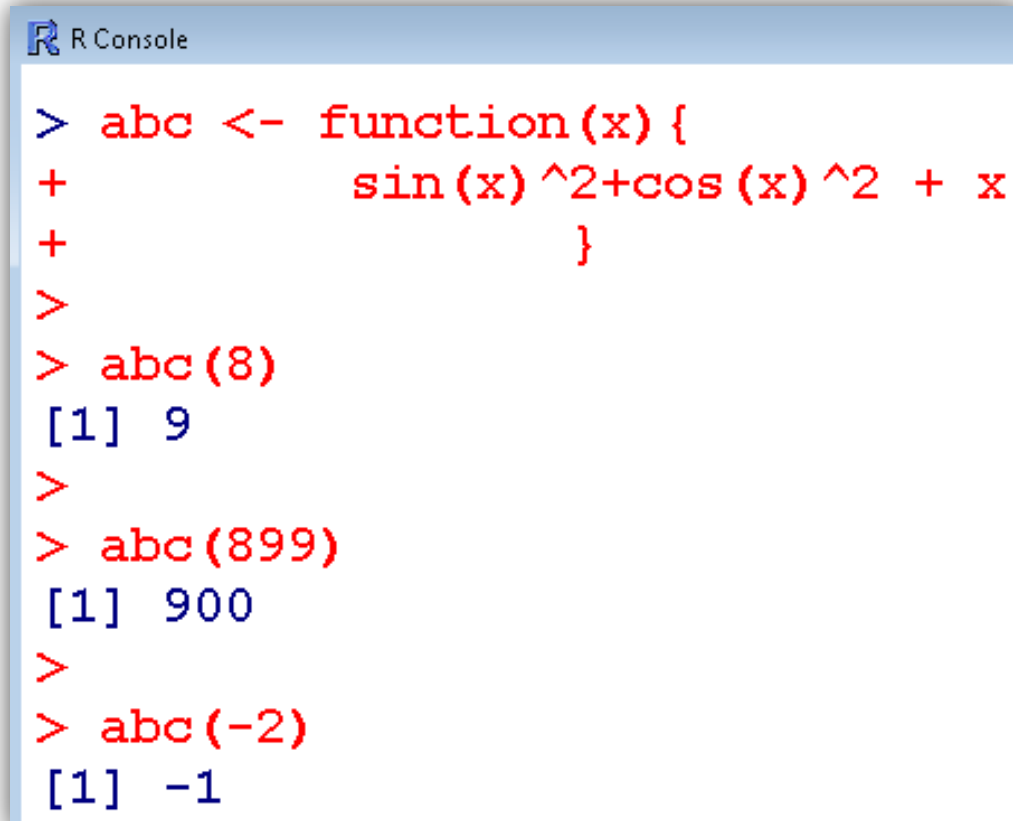
# Functions- Another example

```
> abc <- function(x) {  
  sin(x)^2+cos(x)^2 + x  
}
```

```
> abc(8)  
[1] 9
```

```
> abc(899)  
[1] 900
```

```
> abc(-2)  
[1] -1
```



```
R Console  
  
> abc <- function(x) {  
+   sin(x)^2+cos(x)^2 + x  
+ }  
  
>  
> abc(8)  
[1] 9  
  
>  
> abc(899)  
[1] 900  
  
>  
> abc(-2)  
[1] -1
```

# Matrix

Matrices are important objects in any calculation.

A matrix is a rectangular array with  $p$  rows and  $n$  columns.

An element in the  $i$ -th row and  $j$ -th column is denoted by  $X_{ij}$  (book version) or  $X[i, j]$  ("program version"),  $i = 1, 2, \dots, n, j = 1, 2, \dots, p$ .

An element of a matrix can also be an object, for example a string. However, in mathematics, we are mostly interested in numerical matrices, whose elements are generally real numbers



In R, a  $4 \times 2$ -matrix  $X$  can be created with a following command:

```
> x <- matrix( nrow=4, ncol=2,  
                data=c(1,2,3,4,5,6,7,8) )
```

```
> x
```

	[,1]	[,2]
[1,]	1	5
[2,]	2	6
[3,]	3	7
[4,]	4	8

R Console

```
> x <- matrix( nrow=4, ncol=2, data=c(1,2,3,4,5,6,7,8) )
```

```
>
```

```
> x
```

	[,1]	[,2]
[1,]	1	5
[2,]	2	6
[3,]	3	7
[4,]	4	8

We see:

The parameter **nrow** defines the row number of a matrix.

The parameter **ncol** defines the column number of a matrix.

The parameter **data** assigns specified values to the matrix elements.

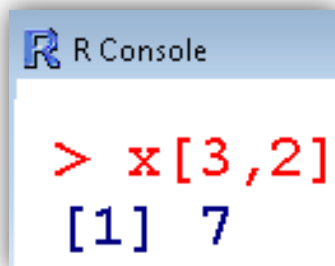
The values from the parameters are written column-wise in matrix.

```
> x
```

	[,1]	[,2]
[1,]	1	5
[2,]	2	6
[3,]	3	7
[4,]	4	8

One can access a single element of a matrix with **x[i,j]** :

```
> x[3,2]  
[1] 7
```



R Console

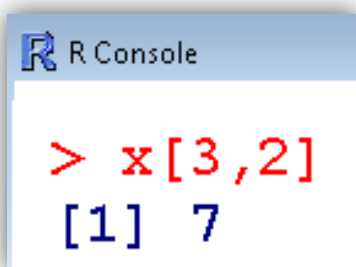
```
> x[3,2]  
[1] 7
```

```
> x
```

	[,1]	[,2]
[1,]	1	5
[2,]	2	6
[3,]	3	7
[4,]	4	8

One can access a single element of a matrix with `x[i,j]` :

```
> x[3,2]  
[1] 7
```



R Console

```
> x[3,2]  
[1] 7
```

In case, the data has to be entered row wise, then a  $4 \times 2$ -matrix  $X$  can be created with

```
> x <- matrix( nrow=4, ncol=2,  
               data=c(1,2,3,4,5,6,7,8) , byrow = TRUE)
```

```
> x
```

	[,1]	[,2]
--	------	------

[1,]	1	2
------	---	---

[2,]	3	4
------	---	---

[3,]	5	6
------	---	---

[4,]	7	8
------	---	---

```
> x <- matrix( nrow=4, ncol=2, data=c(1,2,3,4,5,6,7,8) )
```

```
> x
```

	[,1]	[,2]
[1,]	1	5
[2,]	2	6
[3,]	3	7
[4,]	4	8

```
>
```

```
> x <- matrix( nrow=4, ncol=2, data=c(1,2,3,4,5,6,7,8) , byrow = TRUE)
```

```
> x
```

	[,1]	[,2]
[1,]	1	2
[2,]	3	4
[3,]	5	6
[4,]	7	8