

OPERATORS

	Description	Example	Result
General			
<-	Assignment operator	x <- 1	1
=	Assignment operator (suggested)	y = 1+1 z=3	2 3
#	Comment	#This is a comment	R will not read/execute
?	Help	?topic	Help on term "topic"
Mathematical			
+	Addition	2.5+z	5.5
-	Subtraction	z-2.5	0.5
*	Scalar multiplication	2*3	6
/	Division operator	6/2	3
%%	Remainder	5%%2	1
^	Exponentiation	2^3	8
%*%	Matrix multiplication	mat1%*%mat1	
Logical			
==	Equals	z==3	TRUE
!=	Not Equal	z!=3	FALSE
>	Greater Than	z>3	FALSE
>=	Greater Than or Equal To	z>=3	TRUE
<	Less Than	z<3	FALSE
<=	Less Than or Equal To	z<=3	TRUE
	Or	x==2 x>3	TRUE FALSE TRUE
&	And	x==2 & x<3	TRUE FALSE FALSE

VECTORS

	Command	Example	Result
Creating numeric vector			
Combine elements	<code>c()</code>	<code>x=c(2,4,6)</code> <code>y=c(1,3,5)</code> <code>z=c(x, y)</code>	2,4,6 1,3,5 2,4,6,1,3,5
Colon operator	from : to	3:7	3,4,5,6,7
Sequence generation	<code>seq(from, to, by)</code> <code>seq(from, to, length)</code>	<code>seq(from=2,to=8, by=3)</code> <code>seq(from=2,to=4, length=6)</code>	2,5,8 2.0, 2.4, 2.8, 3.2, 3.6, 4.0
Replicate elements	<code>rep(vector, times)</code> <code>rep(vector, each)</code>	<code>rep(c(1,2),times=3)</code> <code>rep(c(1,2),each=3)</code>	1,2,1,2,1,2 1,1,1,2,2,2
Random sample	<code>sample(integer,size)</code> <code>sample(vector,size)</code> <code>sample(vector,size,replace='TRUE')</code>	<code>sample(5,4)</code> <code>sample(c(5,7,9,4),3)</code> <code>sample(c(5,7,9,4),5,replace='TRUE')</code>	3,5,4,1 #randomly 7,4,5 9,7,9,4,5
Random normally distributed numbers	<code>rnorm(n)</code> <code>rnorm(n,mean,sd)</code>	<code>rnorm(4)</code> <code>rnorm(4,mean=2,sd=2)</code>	1.09, 0.76, 0.36, 0.40
Creating character vector			
Combine elements	<code>c(value1, value2,..)</code>	<code>c("a", "b", "c")</code>	a, b, c
Letters	<code>letters</code>	<code>letters</code> <code>letters[1:4]</code>	a to z a, b, c, d
Concatenate	<code>paste(vectors, sep)</code>	<code>paste("C",1:3,sep=" ")</code>	C1,C2,C3
Fetching elements from a vector			
Single element	<code>object[index]</code>	<code>x=c(2,4,6,7,9)</code> <code>x[2]</code>	2,4,6,7,9 4
Multiple elements	<code>object[c(indices)]</code>	<code>x[c(1,3)]</code>	2, 6
Range elements	<code>object[index : index]</code>	<code>x[2:4]</code>	4,6,7
Exclude single element	<code>object[-index]</code>	<code>x[-2]</code>	2,6,7,9
Exclude multiple	<code>object[-c(indices)]</code>	<code>x[-c(3,5)]</code>	2,4,7
		<code>x[10]</code>	NA
		<code>x[2,4,5]</code>	Error
Functions for a numeric vector			
Function	Description	Function	Description
<code>length(vector)</code>	Size of the vector	<code>sort(vector)</code>	Sort in ascending order
<code>order(vector)</code>	Order the indices	<code>sort(vector, decreasing=TRUE)</code>	Sort in descending order
<code>max(vector)</code>	Returns maximum	<code>min(vector)</code>	Returns minimum
<code>range(vector)</code>	Returns range of vector	<code>mean(vector)</code>	Arithmetic mean
<code>median(vector)</code>	Median of a vector	<code>quantile(vector, prob)</code>	Quantiles with probability

summary(vector)	Statistical summary of vector	var(vector)	Variance of a vector
sd(vector)	Standard deviation	log(,base)	Logarithm with specified base value
sin(vector)	Sine of a vector	cosine(vector)	Cosine of vector
unique(vector)	Returns unique elements of vector		
Functions for a pair of numeric vectors	Syntax Function_name(vector1, vector2)		
Function	Description	Function	Description
cor(vector1, vector2)	Correlation between vectors	union(vector1, vector2)	Union of sets of vectors
intersect(vector1, vector2)	Intersection of vectors	setdiff(vector1, vector2)	Returns unique elements of first vector
Functions for character vector	Syntax: Function_name(vector)		
Function	Description	Function	Description
nchar(vector)	Count the number of characters in each element of vector	substr(vector, start, stop)	Substrings of character elements from start to stop
Functions for data type conversion of vector			
Function	Description	Function	Description
str(vector)	Returns type of data	as.numeric(vector)	Convert to numeric
as.character(vector)	Convert to character		

MATRIX

	Command	Example	Result
Creating a matrix		x=1:6; y=7:12; z=13:18	
From a vector	matrix(vector, nrow)	matrix(x,nrow=3)	1 4 2 5 3 6
	matrix(vector, ncol)	matrix(x,ncol=3)	1 3 5 2 4 6
	matrix(vector, ncol, byrow=TRUE)	matrix(x,ncol=2,byrow=TRUE)	1 2 3 4 5 6
Binding vectors row wise	rbind(vector1, vector2,..)	rbind(x,y,z)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Binding vectors column wise	cbind(vector1, vector2,..)	cbind(x,z)	1 13 2 14 3 15 4 16 5 17 6 18
Fetch elements of a matrix	Object: Matrix object	mat= rbind(x,y,z)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Single element	object[row index, column index]	mat[1,1] mat[3,5]	1 17
Multiple elements of single column	object [c(row indices), column index]	mat[c(2,3),4]	10 16
Multiple elements of single row	object [row index, c(column indices)]	mat[3,c(2,3)]	14 15
Single row	object [row index,]	mat[2,]	7 8 9 10 11 12
Multiple rows	object [c(row indices),]	mat[c(2,3),] mat[1:2,]	7 8 9 10 11 12 13 14 15 16 17 18 1 2 3 4 5 6 7 8 9 10 11 12
Multiple columns	object [, c(column indices)]	mat[,c(2,3)]	2 3 8 9 14 15
Multiple rows and columns	object [c(row indices), c(column indices)]	mat[c(2,3),c(3,4)]	9 10 15 16

Functions for matrix			
Function	Description	Function	Description
str(matrix)	Structure of a matrix	dim(matrix)	Dimensions of a matrix
nrow(matrix)	Number of rows in a matrix	ncol(matrix)	Number of columns in a matrix
colMeans(matrix)	Column means of matrix	rowMeans(matrix)	Row means of matrix
colnames(matrix)	Column names of matrix	rownames(matrix)	Row names of matrix
t(matrix)	Transpose matrix		
Matrix manipulation		x=1:6; mx=matrix(x,ncol=3)	1 3 5 2 4 6
Add column(s) to matrix	cbind(matrix, vector)	y=c(7,8); mx=cbind(mx, y)	1 3 5 7 2 4 6 8
Add row(s) to matrix	rbind(matrix,vector)	z=9:12; mx=rbind(mx, z)	1 3 5 7 2 4 6 8 9 10 11 12
Delete row from a matrix	matrix=matrix[-row index,]	mx=mx[-3,]	1 3 5 7 2 4 6 8
Delete column from a matrix	matrix=matrix[, -column index]	mx=mx[, -4]	1 3 5 2 4 6
Replace value of element	mx[row ,column]= value	mx[2,3]=7	1 3 5 2 4 7
Similarly we can replace rows and columns			

DATA FRAME

	Command	Example	Result
Creating a data frame		x=1:6; y=letters[1:6]; z=paste(y, x, sep= "")	1 2 3 4 5 6 a b c d e f a1 b2 c3 d4 e5 f6
From vectors	data.frame()	D=data.frame(x,y,z)	x y z 1 a a1 2 b b2 3 c c3 4 d d4 5 e e5 6 f f6
Fetch values from a data frame			
Complete column	Data_frame_name\$column_name	D\$x	1 2 3 4 5 6
Single value from a column	DataFrameName\$ColName[index]	D\$x[2]	2
Multiple values from a column	DataFrameName\$ColName[range]	D\$x[3:6]	3 4 5 6
Data frame manipulation		p=c(2,5,7,8,4,9)	
Add column	Data_frame_name\$new_column_name=values	D\$p=p	x y z p 1 a a1 2 2 b b2 5 3 c c3 7 4 d d4 8 5 e e5 4 6 f f6 9
Delete column	Data_frame_name\$column_name=NULL	D\$p=NULL	x y z 1 a a1 2 b b2 3 c c3 4 d d4 5 e e5 6 f f6
Replace single value of column	Data_frame_name\$column_name[index]=new_value	D\$x[2]=100	x y z 1 a a1 100 b b2 3 c c3 4 d d4 5 e e5 6 f f6

Functions to check the data structure of R object

is.vector()	is.matrix()	is.data.frame()	is.factor()
--------------------	--------------------	------------------------	--------------------

CONTROL STRUCTURE

	Syntax	Example
For Loops	<code>for(i in <vector>){ do stuff }</code>	<code>x=1 for(i in 1:5){ x = x*i } x</code>

FUNCTIONS

Syntax	Example
<code>func.name <- function(arg1, arg2, ...){ do stuff; return(ans) }</code>	#Define function to calculate factorial <code>My_factorial <- function(x){ if(!is.integer(x)){ stop("x must be an integer") } ans=1 for(i in 1:x){ ans= ans*i } return(ans) }</code> #Call a function My_factorial to calculate factorial of 5 <code>My_factorial(5)</code>

INPUT AND OUTPUT

Command	Description
<code>read.table(file)</code> <code>read.table(file.choose())</code>	Reads a file in table format and creates a data frame from it Default separator <code>sep=""</code> is any whitespace; Use <code>header=TRUE</code> to read the first line as a header of column names <code>file.choose()</code> allows to interactively select a file.
<code>read.csv(file,header=TRUE)</code>	Same as <code>read.table()</code> but with defaults set for reading comma-delimited files.
<code>library(gdata)</code> <code>read.xls(excel_file, sheet=sheet_number)</code>	Read excel file into data frame
<code>readLines(file)</code>	Read file line by line as character vector. Use option <code>n</code> to specify number of lines to be read.

<code>read.table("clipboard")</code>	Read text copied from a file into data frame
<code>write.table(x,file="",row.names=TRUE,col.names=TRUE,sep=" ")</code>	<p>Prints x after converting to a data frame</p> <p>If quote is TRUE, character or factor columns are surrounded by quotes ("");</p> <p>sep is the field separator;</p> <p>col is the end-of-line separator;</p> <p>na is the string for missing values;</p> <p>use col.names=NA to add a blank column header to get the column headers aligned correctly for spreadsheet input</p> <p>The file argument should be a quoted string specifying the file name or replace it with <code>file.choose(new=FALSE)</code> to interactively select a file.</p>