

# INTRODUCTION TO R PROGRAMMING

# R objects and functions

Object    Assign    Value  
Name    Operator

```
>    x    <-    144
```

A diagram illustrating the components of an R assignment statement. The labels 'Object Name', 'Assign Operator', and 'Value' are positioned above the code snippet. Lines connect 'Object Name' to the variable 'x', 'Assign Operator' to the assignment operator '<-', and 'Value' to the value '144'.

```
>    y <- sqrt(x) # taking the square root
```

Function    Argument    Comment

A diagram illustrating the components of an R function call statement. The labels 'Function', 'Argument', and 'Comment' are positioned below the code snippet. Lines connect 'Function' to the function name 'sqrt', 'Argument' to the argument 'x', and 'Comment' to the comment text '# taking the square root'.

# R objects and functions

Function	Arg1	Arg2	Default value
	x	digits	0

`round(x, digits = 0)`

> round(3.14159)

> round(3.14159, 2)

> round(x = 3.14159, digits = 2)

> round(digits = 2, x = 3.14159)

> round(d = 2, x = 3.14159)

> round(2, 3.14159) ?

# Vectors

## CREATING VECTORS

```
> c(1,4,8,10)      [1] 1 4 8 10
> c("a", "b", "c" ) [1] "a" "b" "c"
> 1:6               [1] 1 2 3 4 5 6
> seq(1,6)          [1] 1 2 3 4 5 6
> seq(1,6, by = 2)  [1] 1 3 5
```

## INFO ON VECTORS

length()

str()

class()

typeof()

# Vectors

## CREATING VECTORS

```
> c(1,4,8,10)      [1] 1 4 8 10
> c("a", "b", "c" ) [1] "a" "b" "c"
> 1:6               [1] 1 2 3 4 5 6
> seq(1,6)          [1] 1 2 3 4 5 6
> seq(1,6, by = 2)  [1] 1 3 5
```

## CONVERTING BETWEEN DATA TYPES

```
as.logical()
as.numeric()
as.character()
as.factor()
```

## OTHER DATA STRUCTURES

```
factor
data.frame
matrix
list
array
```

# Vectors

## ACCESSING VECTOR ELEMENTS

`my_vect[4]` *4<sup>th</sup> element of my\_vect*

`my_vect[c(1,3)]` *1<sup>st</sup> and 3<sup>rd</sup> element*

`my_vect[1:5]` *elements 1 to 5*

`my_vect[ my_vect %in% c(1,2,5) ]` *elements in set c(1,2,5)*

`my_vect[ my_vect < 10 ]` *only the elements less than 10*

## CONDITIONAL STATEMENTS

AND	&
OR	
GREATER (OR EQUAL) THAN	> (>=)
LESS (OR EQUAL) THAN	< (<=)
EQUAL TO	==
NOT EQUAL TO	!=
SEARCH IN VECTOR	%in%

## DEALING WITH MISSING DATA

`mean(my_vect, na.rm = TRUE)`

`na.omit(my_vect)`

`complete.cases(my_vect)`

# Data.frames

Download data      `download.file( url="http://...", destfile = "data/table.csv" )`

Read a file      `read.csv( file="data/table.csv" )`

Check data frame      `head()`  
                         `View()`  
                         `str()`

Size      `dim()`  
         `nrow()`  
         `ncol()`

Content      `head()`  
             `tail()`

Names      `names(), colnames()`  
            `rownames()`

Summary      `str()`  
             `summary()`

# Factors

```
year_fct <- factor(c((1990, 1983, 1977, 1998, 1990)))
```

```
Levels: 1977,1983,1990,1998
```

```
Values: c(3,2,1,4,3)
```

number of levels	nlevels()
accessing levels	levels()