

Descriptive Statistics With R Software

Calculations with R Software

:::

Built-in Commands and Missing Data Handling

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

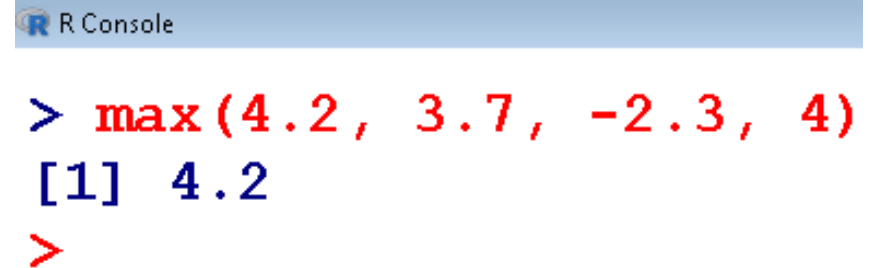
Built in commands

Some commands are readily available in R to compute mathematical functions.

How to use them and utilize them in computing various quantities.

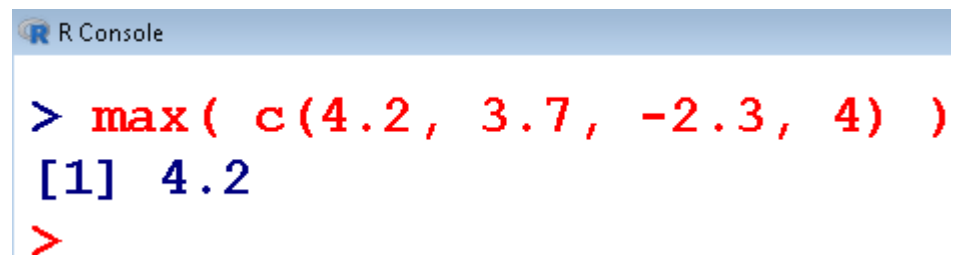
Maximum

```
> max(4.2, 3.7, -2.3, 4)
[1] 4.2
```

A screenshot of an R console window with a blue header bar containing the R logo and the text "R Console". The console shows the command `> max(4.2, 3.7, -2.3, 4)` in red text, followed by the output `[1] 4.2` in blue text, and a red prompt character `>` on the next line.

```
> max(4.2, 3.7, -2.3, 4)
[1] 4.2
>
```

```
> max( c(4.2, 3.7, -2.3, 4) )
[1] 4.2
```

A screenshot of an R console window with a blue header bar containing the R logo and the text "R Console". The console shows the command `> max(c(4.2, 3.7, -2.3, 4))` in red text, followed by the output `[1] 4.2` in blue text, and a red prompt character `>` on the next line.

```
> max( c(4.2, 3.7, -2.3, 4) )
[1] 4.2
>
```

Minimum

```
> min(4.2, 3.7, -2.3, 4)
[1] -2.3
```

```
R Console
> min(4.2, 3.7, -2.3, 4)
[1] -2.3
>
```

```
> min( c(4.2, 3.7, -2.3, 4) )
[1] -2.3
```

```
R Console
> min( c(4.2, 3.7, -2.3, 4) )
[1] -2.3
>
```

Overview Over Other Functions

<code>abs()</code>	Absolute value
<code>sqrt()</code>	Square root
<code>round()</code> , <code>floor()</code> , <code>ceiling()</code>	Rounding, up and down
<code>sum()</code> , <code>prod()</code>	Sum and product
<code>log()</code> , <code>log10()</code> , <code>log2()</code>	Logarithms
<code>exp()</code>	Exponential function
<code>sin()</code> , <code>cos()</code> , <code>tan()</code> , <code>asin()</code> , <code>acos()</code> , <code>atan()</code>	Trigonometric functions
<code>sinh()</code> , <code>cosh()</code> , <code>tanh()</code> , <code>asinh()</code> , <code>acosh()</code> , <code>atanh()</code>	Hyperbolic functions

Examples

```
> sqrt(4)
[1] 2
```

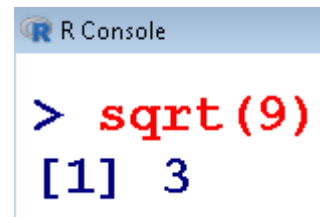
```
> sqrt(c(4,9,16,25))
[1] 2 3 4 5
```

R Console

```
> sqrt(4)
[1] 2
>
> sqrt(c(4,9,16,25))
[1] 2 3 4 5
>
```

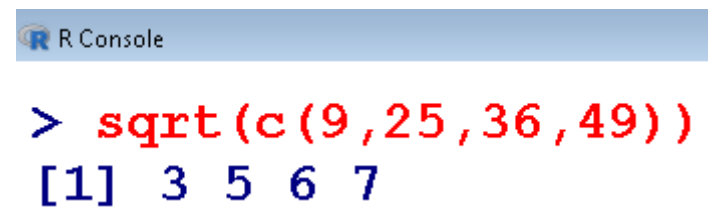
Examples

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



```
R Console
> sqrt(9)
[1] 3
```

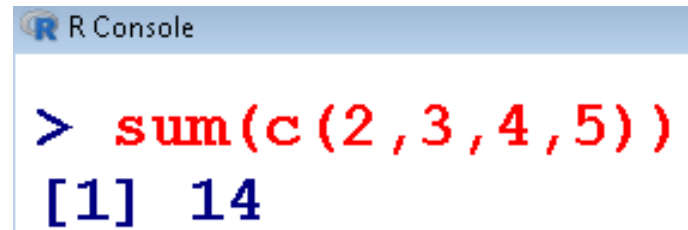
```
> sqrt(c(9,25,36,49))
[1] 3 5 6 7
```



```
R Console
> sqrt(c(9,25,36,49))
[1] 3 5 6 7
```

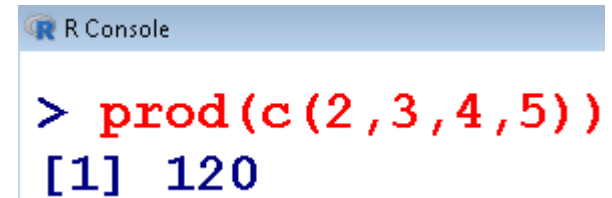
Examples

```
> sum(c(2,3,4,5))  
[1] 14
```

A screenshot of an R console window. The title bar is light blue and contains the R logo and the text "R Console". The console area has a white background and displays the command "> sum(c(2,3,4,5))" in red text, followed by the output "[1] 14" in blue text.

```
> sum(c(2,3,4,5))  
[1] 14
```

```
> prod(c(2,3,4,5))  
[1] 120
```

A screenshot of an R console window. The title bar is light blue and contains the R logo and the text "R Console". The console area has a white background and displays the command "> prod(c(2,3,4,5))" in red text, followed by the output "[1] 120" in blue text.

```
> prod(c(2,3,4,5))  
[1] 120
```

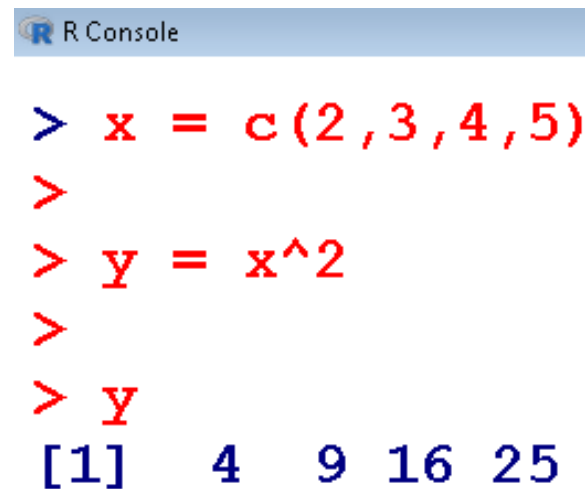

Assignments

An assignment can also be used to save values in variables:

```
> x = c(2,3,4,5)
```

```
> y = x^2
```

```
> y  
[1] 4 9 16 25
```



```
R Console  
> x = c(2,3,4,5)  
>  
> y = x^2  
>  
> y  
[1] 4 9 16 25
```

Examples

To find sum of squares:

```
> x = c(2,3,4,5)
```

$$z = \sum_{i=1}^n x_i^2$$

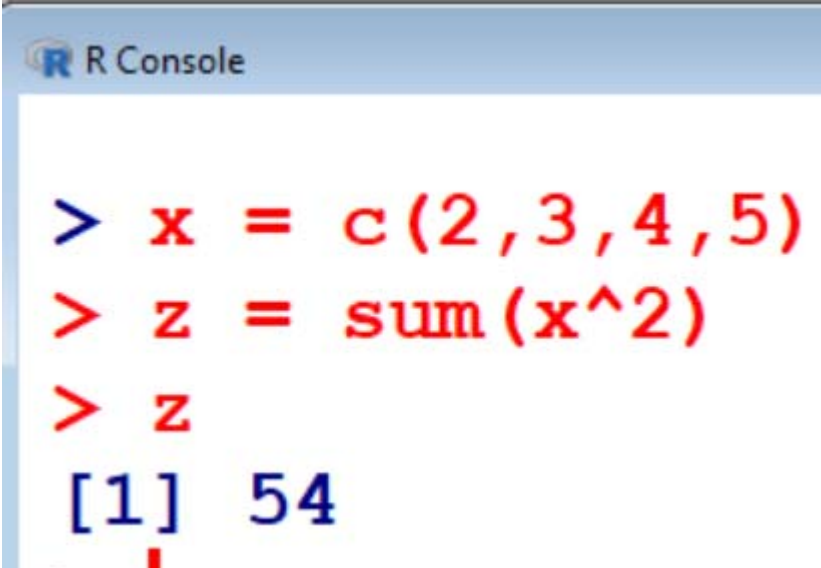
Examples

To find sum of squares:

```
> x = c(2,3,4,5)
```

```
> z = sum(x^2)
```

```
> z  
[1] 54
```



```
R Console  
  
> x = c(2,3,4,5)  
> z = sum(x^2)  
> z  
[1] 54
```

Examples

To find sum of squares of deviation from mean

> **x = c(2,3,4,5)**

$$z = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2$$

Examples

To find sum of squares of deviation from mean

```
> x = c(2,3,4,5)
```

$$z = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2$$

```
> z = sum(x^2)-length(x)*mean(x)^2
```

```
> length(x)
```

```
[1] 4
```

```
> z
```

```
[1] 5
```

```
R Console  
  
> x = c(2,3,4,5)  
> length(x)  
[1] 4  
> z = sum(x^2)-length(x)*mean(x)^2  
> z  
[1] 5  
> |
```

Examples

To find sum of cross product:

```
> x1 = c(2,3,4,5)
```

```
> x2 = c(6,7,8,9)
```

$$2 \times 6 + 3 \times 7 + 4 \times 8 + 5 \times 9$$

```
> z = sum(x1*x2)
```

```
> z
```

```
[1] 110
```

Missing data

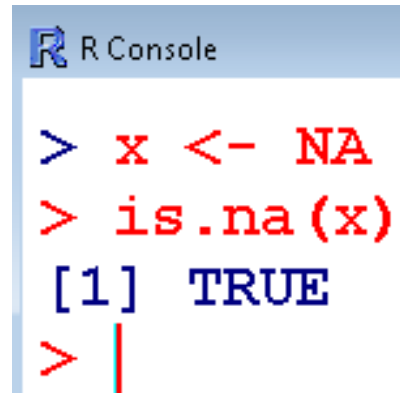
R represents missing observations through the data value **NA**

NA : Reserved word

Missing values can be detected in a data vector by using **is.na**

Create a data with NA.

```
> x <- NA      # assign NA to variable x
> is.na(x)     # is it missing?
[1] TRUE
```

A screenshot of the R Console window. The title bar says "R Console". The console shows the following commands and output: "> x <- NA", "> is.na(x)", "[1] TRUE", and "> |" (indicating the prompt is ready for the next command).

```
R Console
> x <- NA
> is.na(x)
[1] TRUE
> |
```

Missing data

TRUE and **FALSE** are logical operators that are used to compare expressions.

TRUE and **FALSE** are reserved words.

T can also be used in place of **TRUE** .

F can also be used in place of **FALSE**.

TRUE and **FALSE** are not the same as **true** and **false** respectively.

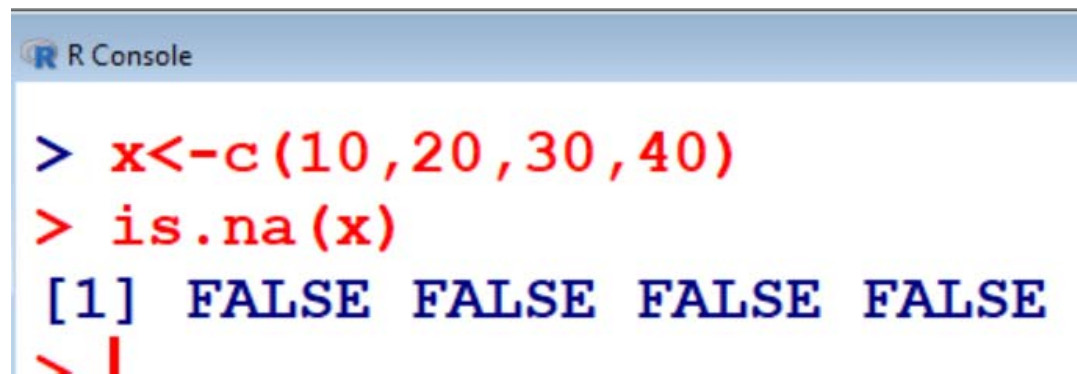
Missing data

How to know if any value is missing in a data vector?

```
> x <- c(10,20,30,40)
```

```
> is.na(x)
```

```
[1] FALSE FALSE FALSE FALSE
```

A screenshot of an R console window. The title bar is light blue and contains the R logo and the text "R Console". The console area has a white background and displays the following text: the command "> x<-c(10,20,30,40)" in red, the command "> is.na(x)" in red, the output "[1] FALSE FALSE FALSE FALSE" in blue, and a red prompt character ">" on the next line.

```
> x<-c(10,20,30,40)
> is.na(x)
[1] FALSE FALSE FALSE FALSE
> |
```

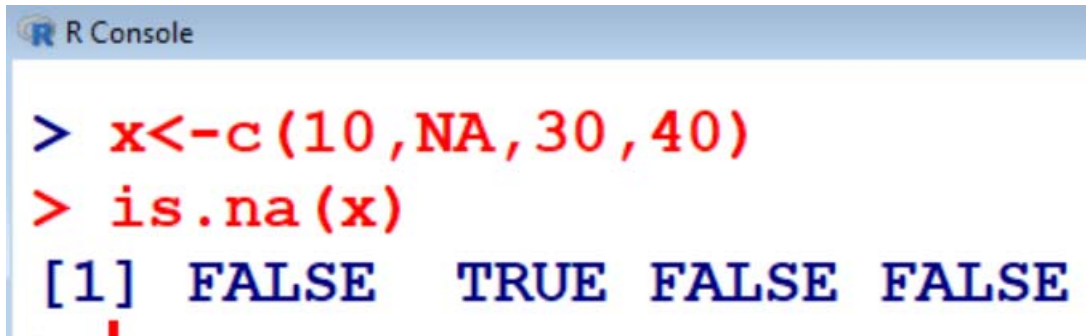
Missing data

How to know if any value is missing in a data vector?

```
> x <- c(10,NA,30,40)
```

```
> is.na(x)
```

```
[1] FALSE TRUE FALSE FALSE
```

A screenshot of an R console window. The title bar says "R Console". The console shows the following R code and output:

```
> x<-c(10,NA,30,40)
> is.na(x)
[1] FALSE TRUE FALSE FALSE
```

The screenshot shows a standard R console interface with a light blue header bar containing the R logo and the text "R Console". Below the header, the R code is entered in red text, and the output is shown in dark blue text. The code defines a vector 'x' with values 10, NA, 30, and 40, and then checks for missing values using 'is.na(x)'. The output shows a logical vector of length 4, where the second element is TRUE and the others are FALSE.

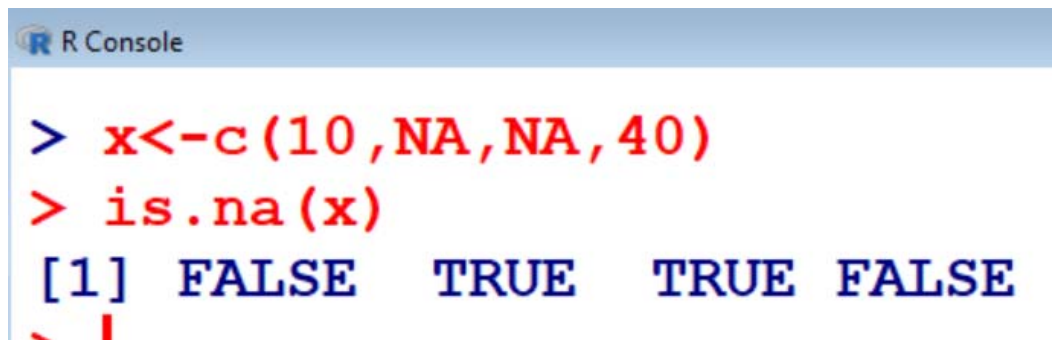
Missing data

How to know if any value is missing in a data vector?

```
> x <- c(10,NA,NA,40)
```

```
> is.na(x)
```

```
[1] FALSE TRUE TRUE FALSE
```

A screenshot of an R console window. The title bar is light blue and contains the R logo and the text "R Console". The console area has a white background. It shows the same R code and output as the previous block: the command to create a vector with missing values, the command to check for missing values, and the resulting logical vector output.

```
> x<-c(10,NA,NA,40)
> is.na(x)
[1] FALSE TRUE TRUE FALSE
~ |
```

Example : How to work with missing data

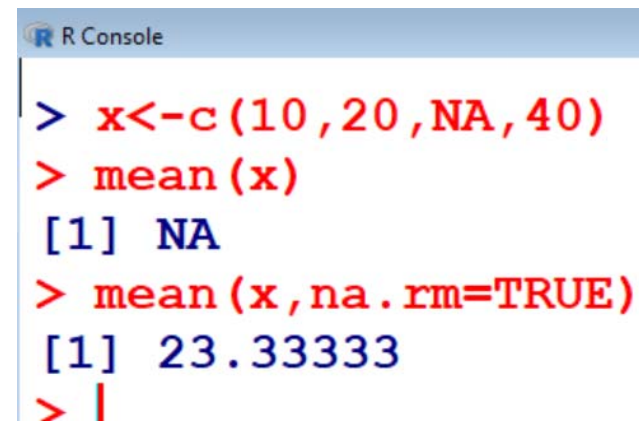
```
> x <- c(10,20,NA,40) # data vector
```

```
> mean(x)  $\frac{10+20+NA+40}{4}$   
[1] NA
```

```
> mean(x, na.rm = TRUE) # NAs can be removed
```

```
[1] 23.33333
```

$$\frac{10+20+40}{3} = 23.33$$



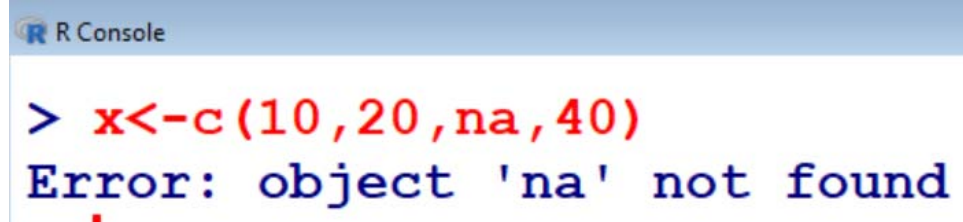
```
R Console  
> x<-c(10,20,NA,40)  
> mean(x)  
[1] NA  
> mean(x,na.rm=TRUE)  
[1] 23.33333  
> |
```

Example : How to work with missing data

The null object, called **NULL**, is returned by some functions and expressions.

Note that **NA** and **NULL** are not the same.

Note that **NA** and **na** are not the same.

A screenshot of an R console window. The title bar says "R Console". The prompt is ">". The command entered is "x<-c(10,20,na,40)". The error message displayed is "Error: object 'na' not found".

```
> x<-c(10,20,na,40)
Error: object 'na' not found
```

NA is a placeholder for something that exists but is missing.

NULL stands for something that never existed at all.