

Essentials of Data Science With R Software - 1

Probability and Statistical Inference

Introduction to R Software

:::

Lecture 6

Built-in Commands and Bivariate Plots

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Built in commands

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

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

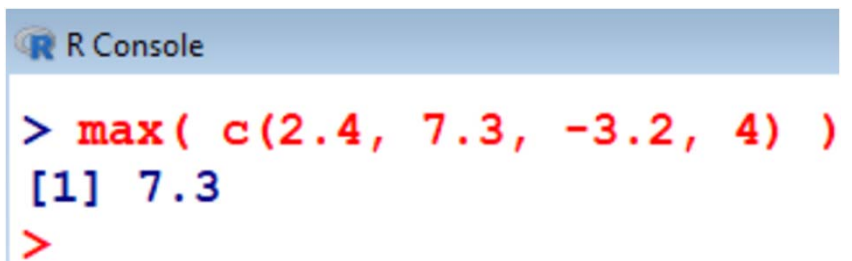
Maximum

```
> max(2.4, 7.3, -3.2, 4)
[1] 7.3
```

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(2.4, 7.3, -3.2, 4)` in red text, followed by the output `[1] 7.3` in blue text, and a red prompt character `>` on the next line.

```
> max(2.4, 7.3, -3.2, 4)
[1] 7.3
>
```

```
> max( c(2.4, 7.3, -3.2, 4) )
[1] 7.3
```

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(2.4, 7.3, -3.2, 4))` in red text, followed by the output `[1] 7.3` in blue text, and a red prompt character `>` on the next line.

```
> max( c(2.4, 7.3, -3.2, 4) )
[1] 7.3
>
```

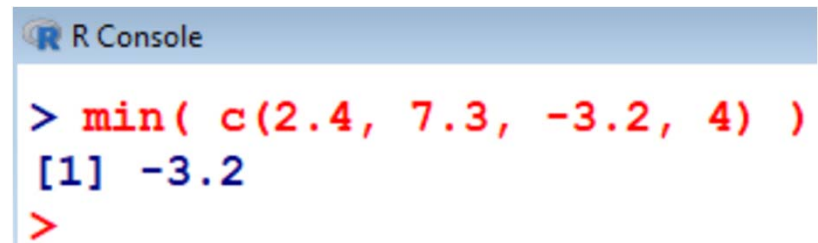
Minimum

```
> min(2.4, 7.3, -3.2, 4)
[1] -3.2
```

A screenshot of the R Console window. The title bar says "R Console". The prompt ">" is followed by the command "min(2.4, 7.3, -3.2, 4)" in red text. The output "[1] -3.2" is shown in blue text. The prompt ">" is shown again in red text at the bottom.

```
> min(2.4, 7.3, -3.2, 4)
[1] -3.2
>
```

```
> min( c(2.4, 7.3, -3.2, 4) )
[1] -3.2
```

A screenshot of the R Console window. The title bar says "R Console". The prompt ">" is followed by the command "min(c(2.4, 7.3, -3.2, 4))" in red text. The output "[1] -3.2" is shown in blue text. The prompt ">" is shown again in red text at the bottom.

```
> min( c(2.4, 7.3, -3.2, 4) )
[1] -3.2
>
```

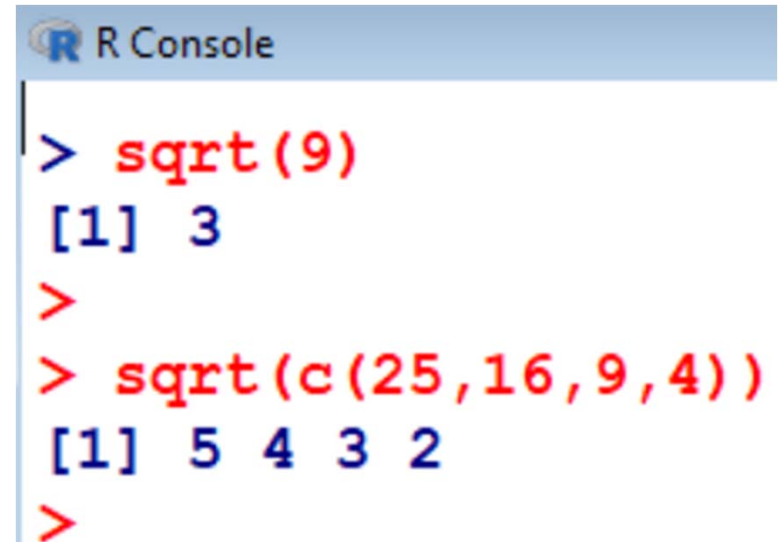
Overview Over Other Functions

<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(9)
[1] 3
```

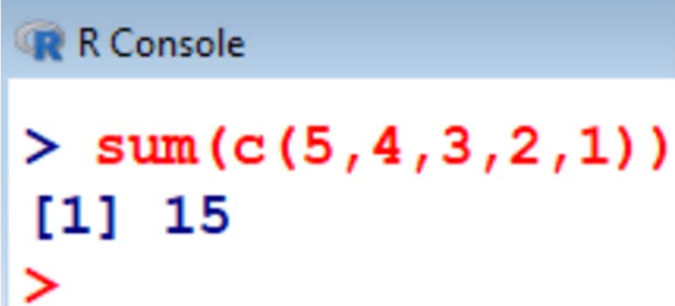
```
> sqrt(c(25,16,9,4))
[1] 5 4 3 2
```

A screenshot of an R console window. The title bar is blue and contains the R logo and the text "R Console". The console area is white and shows the following text: a red prompt character ">" followed by the red command "sqrt(9)", a blue output "[1] 3", a red prompt character ">", a red command "sqrt(c(25,16,9,4))", a blue output "[1] 5 4 3 2", and a final red prompt character ">".

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

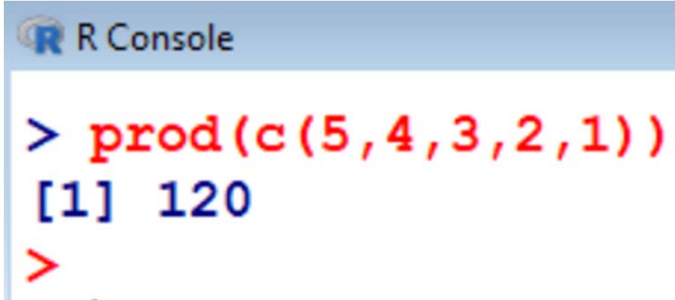
Examples

```
> sum(c(5,4,3,2,1))  
[1] 15
```



A screenshot of the 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 command `> sum(c(5,4,3,2,1))` in red text, followed by the output `[1] 15` in blue text. A red prompt character `>` is on the next line.

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



A screenshot of the 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 command `> prod(c(5,4,3,2,1))` in red text, followed by the output `[1] 120` in blue text. A red prompt character `>` is on the next line.

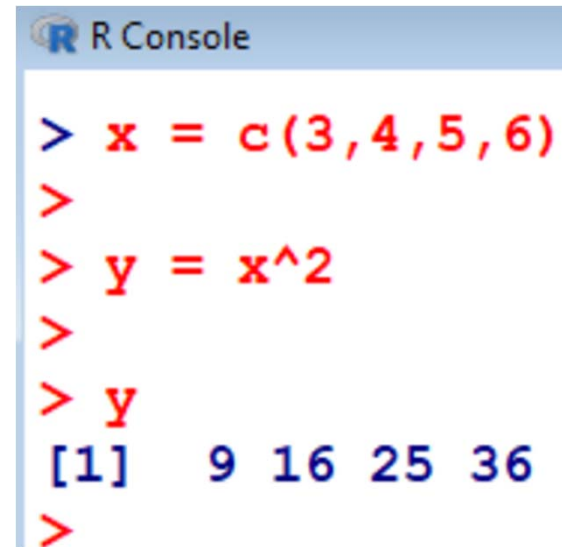
Assignments

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

```
> x = c(3,4,5,6)
```

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

```
> y  
[1] 9 16 25 36
```

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

```
> x = c(3,4,5,6)  
>  
> y = x^2  
>  
> y  
[1] 9 16 25 36  
>
```

The commands and their corresponding output are displayed in red text. The prompt character is a red greater-than sign (>).

Bivariate plots:

Provide first hand visual information about the nature and degree of relationship between two variables.

Relationship can be linear or nonlinear.

We discuss several types of plots through examples.

Bivariate plots: Scatter plot

Plot command:

x, y: Two data vectors

`plot(x, y)`

`plot(x, y, type)`

type	
"p" for <u>p</u> oints	"l" for <u>l</u> ines
"b" for <u>b</u> oth	"c" for the lines part alone of "b"
"o" for both ' <u>o</u> verplotted'	"s" for stair <u>s</u> teps.
"h" for ' <u>h</u> istogram' like (or 'high-density') vertical lines	

Bivariate plots: Scatter plot

Plot command:

x, y: Two data vectors

```
plot(x, y)
```

```
plot(x, y, type)
```

Get more details from help: `help("type")`

Other options:

main an overall title for the plot.

suba sub title for the plot.

xlaba title for the x axis.

ylaba title for the y axis.

aspthe y/x aspect ratio.

Bivariate plots: Example

Number of marks obtained by students depend upon the number of hours of study.

Data on marks out of 500 maximum marks and number of hours per week for 20 students are collected as follows:

Marks out of 500 maximum marks

```
marks <- c(337,316,334,327,340,360, 374,330,352,  
353,370,380,384,398,413,428,430,438,439,450)
```

Number of hours per week

```
hours <- c(23,25,25,26,27,28,30,26,29,32,33,34,  
35,38,39,42,43,44, 45,45.5)
```

Bivariate plots: Scatter plot

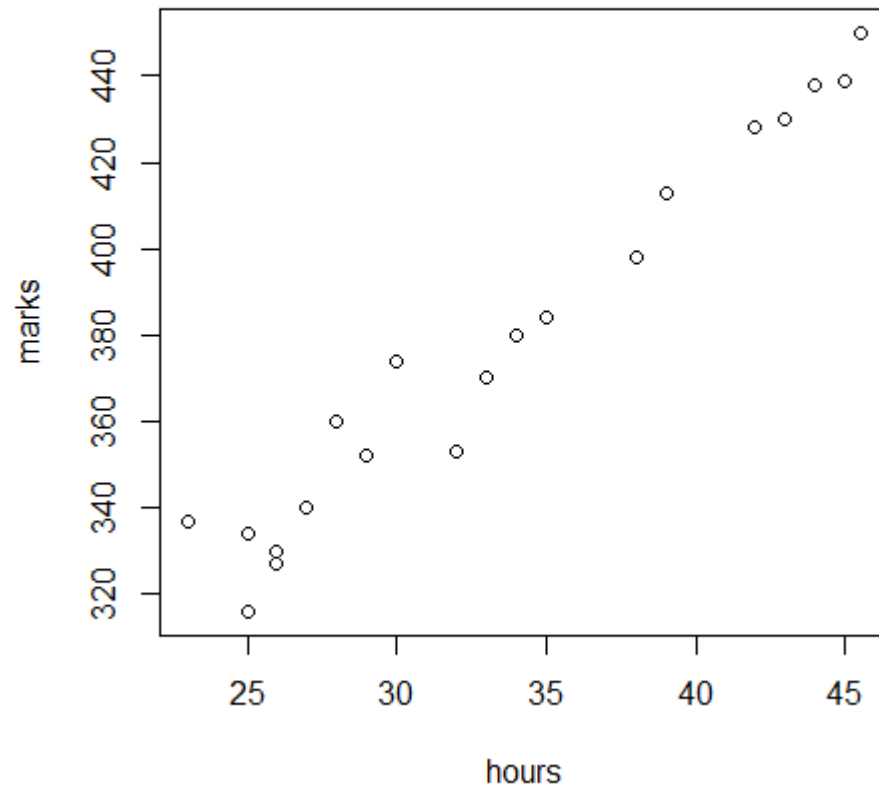
`plot` command:

`x, y`: Two data vectors

Various type of plots are possible to draw.

`plot(x, y)`

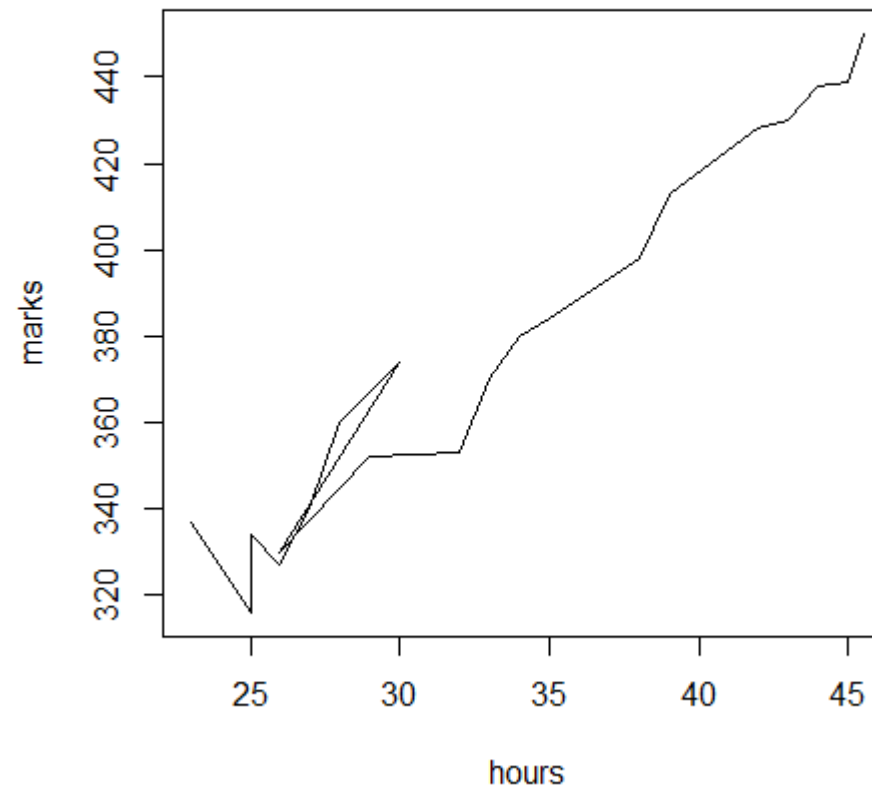
`plot(hours, marks)`



Bivariate plots: Scatter plot

```
plot(hours, marks, "l")
```

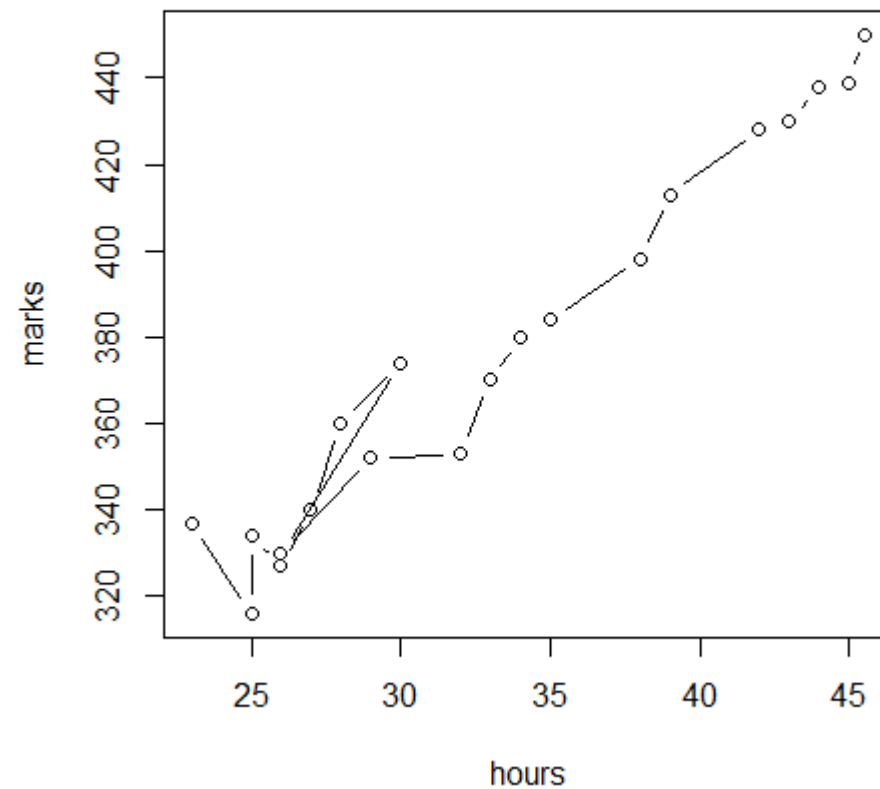
“l” for lines,



Bivariate plots: Scatter plot

```
plot(hours, marks, "b")
```

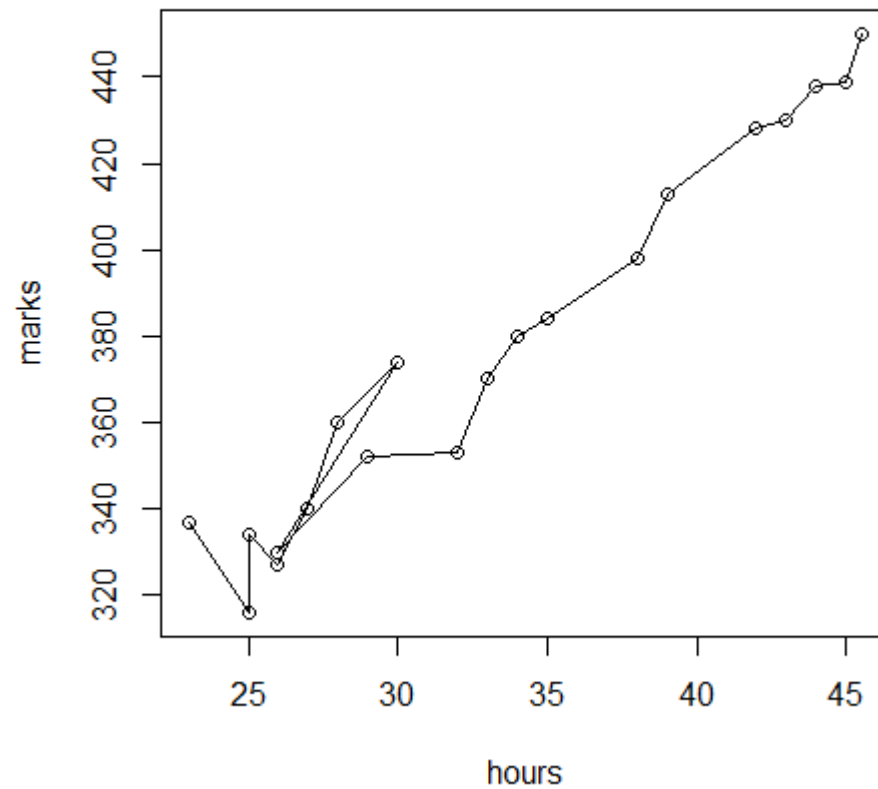
"b" for both – line and point



Bivariate plots: Scatter plot

```
plot(hours, marks, "o")
```

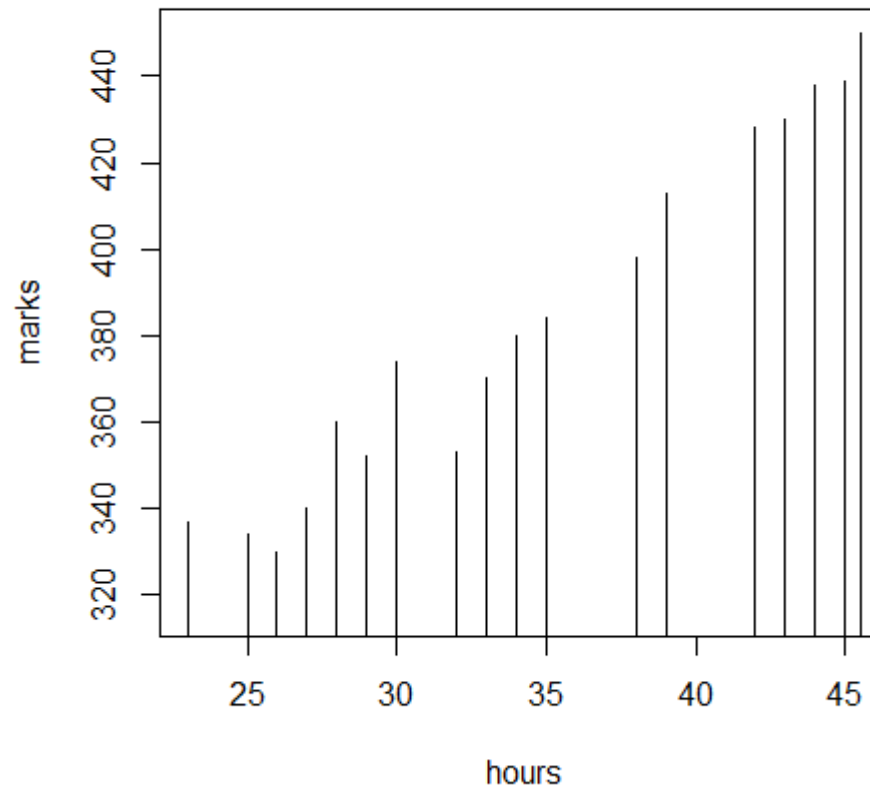
“o” for both ‘overplotted’



Bivariate plots: Scatter plot

```
plot(hours, marks, "h")
```

"h" for 'histogram' like (or 'high-density') vertical lines



Bivariate plots: Scatter plot

```
plot(hours, marks, "s")
```

“s” for stair steps.

