
Programming

-
- Conditional executions
 - Repetitive executions

Conditional executions:

if statement

```
if (expr_1) expr_2 else expr_3
```

where:

expr_1 is a conditional statement which can assume value TRUE or FALSE:

- ❑ if *expr_1* is TRUE, *expr_2* will be evaluated
 - ❑ if *expr_1* is FALSE, *expr_3* will be evaluated
-

-
- Relation operators `&&` and `||` are often used in the if statement:

Say `c1` and `c2` are the conditions:

`if (c1 && c2) expr_2 else expr_3`

`if (c1 || c2) expr_2 else expr_3`

Note:

- `&&` and `||` apply to vectors of length one
 - `&` and `|` apply to each element of the vector
-

Example

```
> a<- seq(5,50,by=5)
> a
[1] 5 10 15 20 25 30 35 40 45 50
> b <- c(1:10)
> b
[1] 1 2 3 4 5 6 7 8 9 10
>
>
> if (length(a)==length(b)) c=a*b else cat ("Warning, the vectors have different length ")
> c
[1] 5 20 45 80 125 180 245 320 405 500
>
> b <- c(2*(0:10))
>
> b
[1] 0 2 4 6 8 10 12 14 16 18 20
> length(b)
[1] 11
> if (length(a)==length(b)) c=a*b else cat ("Warning, the vectors have different length ")
Warning, the vectors have different length
```

Repetitive executions: for loop

```
for (var in range )  
{  
    command_1  
    command_2  
    ...  
    command_n  
}
```

- “var” is the loop variable varying in “range”
 - command_1 ... command_n is repeatedly evaluated till “var” is in the range
-

Example

```
> for (i in 1:5)  
+ {cat ("Natural logarithm of ", i , "is ", log(i), "\n")}
```

Natural logarithm of 1 is 0

Natural logarithm of 2 is 0.6931472

Natural logarithm of 3 is 1.098612

Natural logarithm of 4 is 1.386294

Natural logarithm of 5 is 1.609438
