

Foundations of R Software

Lecture 21

Basics of Calculations

::::

Functions for Conditional Executions – switch and which commands

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Control structures in R :

Control statements,

Functions,

Loops.

Some functions useful in conditional execution: **switch()**

switch is a substitute for long **if** statements that compare a variable to several integral values.

switch is a multiway branch statement.

switch tests a variable for equality against a list of values.

switch map and search over a list of values.

If there are more than one matches for any given value, then

switch returns the first matched value.

Some functions useful in conditional execution: **switch()**

switch evaluates **expr** and accordingly chooses one of the further arguments (in ...).

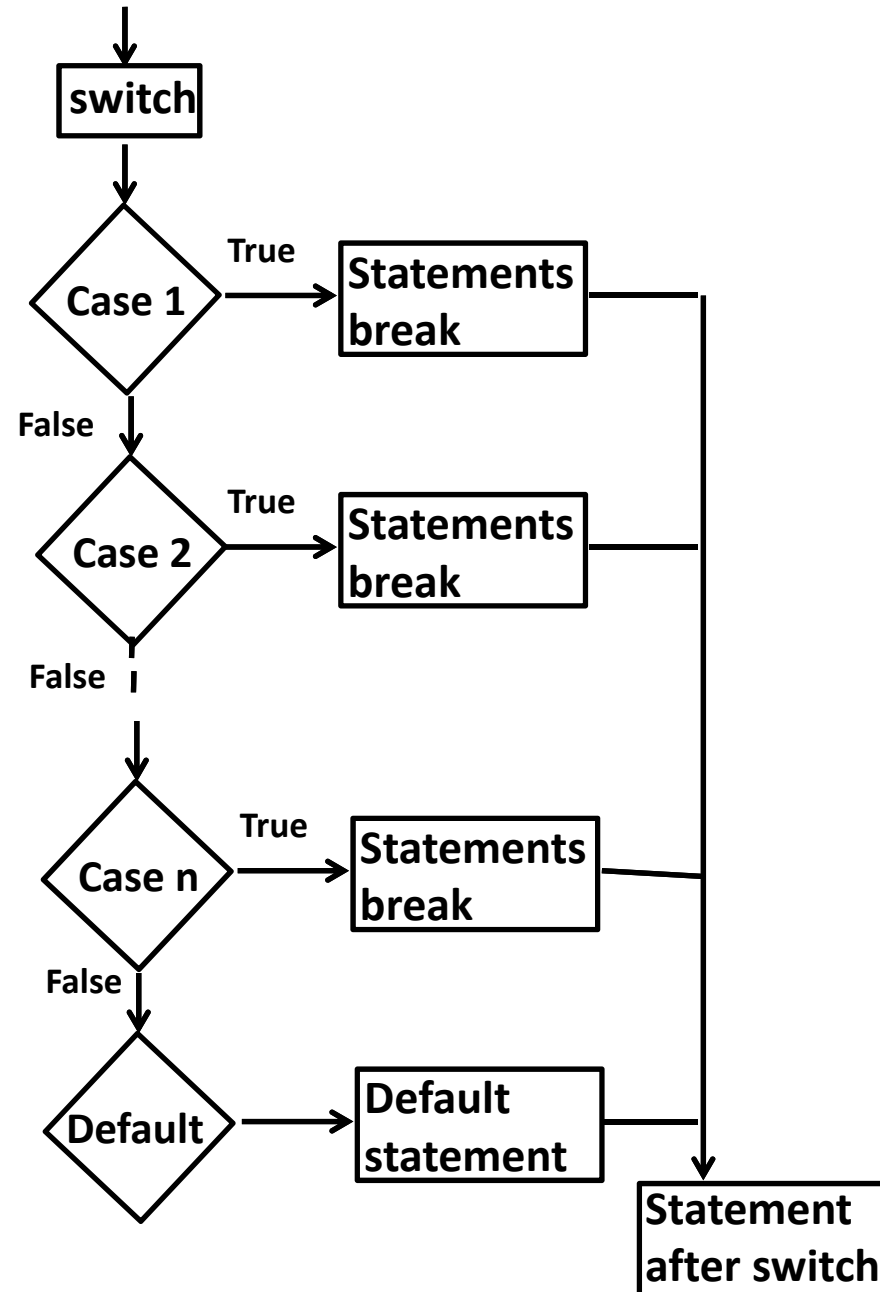
switch(expr, ...)

expr : an expression evaluating to a number or a character string.

switch command allows a variable to be checked for equality against a list of values or cases.

switch(expr, case1, case2,)

Some functions useful in conditional execution: **switch()**



Some functions useful in conditional execution: switch()

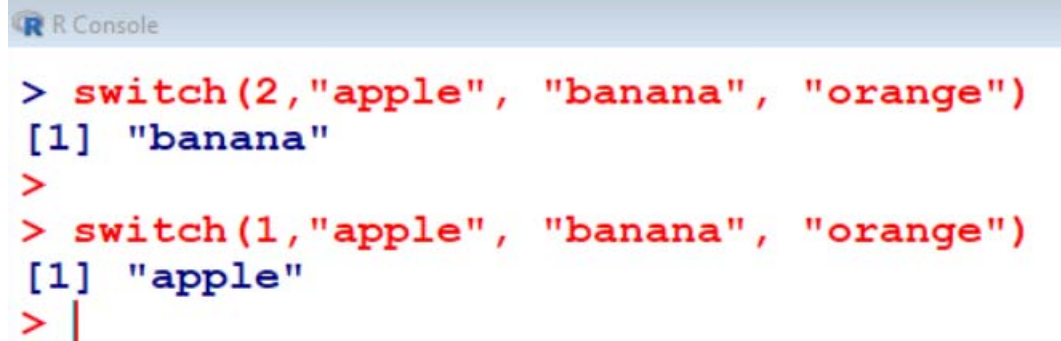
- **A character string expression always matched to the listed cases.**
- **A non character string expression is coerced to integer.**
- **For multiple matches, the first match element will be used.**

Some functions useful in conditional execution: `switch()`

Example 1: `switch ()` function used as an integer

```
> switch(2, "apple", "banana", "orange")  
[1] "banana"
```

```
> switch(1, "apple", "banana", "orange")  
[1] "apple"
```

A screenshot of an R console window showing the execution of the switch function. The console has a light blue header with the R logo and the text "R Console". Below the header, the following commands and outputs are displayed:

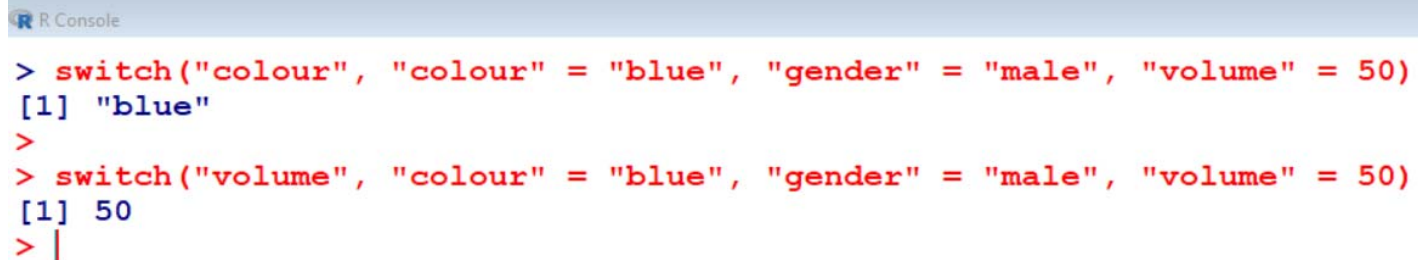
```
> switch(2, "apple", "banana", "orange")  
[1] "banana"  
>  
> switch(1, "apple", "banana", "orange")  
[1] "apple"  
> |
```

Some functions useful in conditional execution: switch()

Example 2: `switch ()` function used as a string as well. The matching named item's value is returned.

```
> switch("colour", "colour" = "blue", "gender" =  
"male", "volume" = 50)  
[1] "blue"
```

```
> switch("volume", "colour" = "blue", "gender" =  
"male", "volume" = 50)  
[1] 50
```

A screenshot of an R console window showing the execution of the switch function. The console has a light blue header bar with the 'R' logo and the text 'R Console'. Below the header, the following commands and their outputs are shown in red text:

```
> switch("colour", "colour" = "blue", "gender" = "male", "volume" = 50)  
[1] "blue"  
>  
> switch("volume", "colour" = "blue", "gender" = "male", "volume" = 50)  
[1] 50  
> |
```

Some functions useful in conditional execution: switch()

Example 3: In the case of no match, if there is a unnamed element of ... its value is returned.

```
> switch(4, "apple", "banana", "orange")
```

No outcome

```
> switch("size", "colour" = "blue", "gender" =  
"male", "volume" = 50)
```

No outcome

```
R Console  
> switch(4, "apple", "banana", "orange")  
> switch("size", "colour" = "blue", "gender" = "male", "volume" = 50)  
> |
```

Some functions useful in conditional execution: `which()`

The `which()` function returns the position of the elements in a logical vector which are **TRUE**.

Give the **TRUE** indices of a logical object, allowing for array indices.

`which(x, arr.ind, useNames)`

x: Specified input logical vector

arr.ind: logical, returns the array indices if **x** is an array.

useNames: logical, says the dimension names of an array.

Some functions useful in conditional execution: which()

Example 4:

```
> x = c(10,15,8,14,6,12)
```

```
> x
```

```
[1] 10 15 8 14 6 12
```

```
> which(x == 14)
```

```
[1] 4
```

```
> which(x != 12)
```

```
[1] 1 2 3 4 5
```

```
> which(x > 10)
```

```
[1] 2 4 6
```

```
R Console  
  
> x = c(10,15,8,14,6,12)  
> x  
[1] 10 15 8 14 6 12  
> which(x == 14)  
[1] 4  
> which(x != 12)  
[1] 1 2 3 4 5  
> which(x > 10)  
[1] 2 4 6  
> |
```

Some functions useful in conditional execution: which()

Example 5:

```
> x = matrix(nrow=3, ncol=3, data=1:9)
```

```
> x
```

	[,1]	[,2]	[,3]
[1,]	1	4	7
[2,]	2	5	8
[3,]	3	6	9

```
> which.min(x) #find which is the minimum value
```

```
[1] 1
```

```
> which.max(x) #find which is the maximum value
```

```
[1] 9
```

Some functions useful in conditional execution: which()

Example 5:

```
> x = matrix(nrow=3, ncol=3, data=1:9)
```

```
> x
```

	[,1]	[,2]	[,3]
[1,]	1	4	7
[2,]	2	5	8
[3,]	3	6	9

```
> which(x %% 2 == 1)
```

```
[1] 1 3 5 7 9
```

```
> which(x %% 2 == 1, arr.ind = TRUE)
```

	row	col
[1,]	1	1
[2,]	3	1
[3,]	2	2
[4,]	1	3
[5,]	3	3

Gives the positions (row, columns) of the values

Some functions useful in conditional execution: which()

```
R Console

> x = matrix(nrow=3,ncol=3, data=1:9)
> x
      [,1] [,2] [,3]
[1,]     1     4     7
[2,]     2     5     8
[3,]     3     6     9
> which.min(x)
[1] 1
> which.max(x)
[1] 9
> which(x %% 2 == 1)
[1] 1 3 5 7 9
> which(x %% 2 == 1, arr.ind=TRUE)
      row col
[1,]    1  1
[2,]    3  1
[3,]    2  2
[4,]    1  3
[5,]    3  3
```