Introduction to Control Structures

Aniel Nieves-González

Institute of Statistics

Spring 2014

Logical and relational operators

- Let x be a boolean variable, i.e., it can only be $\{0, 1\}$ ($\{\text{TRUE}, \text{FALSE}\}$). For example in R set:

$$x = T \text{ or } y = \text{c}(T, F)$$

- There are high level languages, e.g., C in which in the original language specification there was not a logical data type, but the "TRUE" or "FALSE" was simple a nonzero integer and zero integer respectively.

Logical and relational operators I

Let x and y be logical variables. The logical operators:

1. **NOT**: In R is $!x$ (in MATLAB is $\sim x$ and in C is !). Recall that the truth table for not is

x	NOT x
0	1
1	0

2. **AND**: In R we have the elementwise AND and the "regular" AND, which are denoted as $x\&y$ and $x\&\&y$ respectively. Recall that the truth table for AND is:

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

Logical and relational operators II

The "regular" AND is evaluated from left to right and the evaluation proceeds until a result is obtained (short-circuit). This form is more appropiate for control flow.

3. **OR**: In R we have the elementwise OR and the "regular" OR, which are denoted as $x|y$ and $x||y$ respectively. Recall that the truth table for AND is:

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

Logical and relational operators III

The "regular" OR is evaluated from left to right and the evaluation proceeds until a result is obtained (short-circuit). This form is more appropiate for control flow.

Those were the basic operations. Note that with those operations and with logical variables we can define an algebra over $\{0, 1\}$ were the AND and OR would be a product and a sum respectively.

Two additional operations are in R are:

- **XOR**: The exclusive OR. In R the function is XOR(x, y). The truth table for the XOR is:

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	0

Logical and relational operators IV

Observe that the XOR can be constructed with the basic operations.

- istrue(x).

The relational operations are operations of comparison between numbers. The return a logical value. Let x and y be numeric variables (not logical). The basic relational operations are:

1. $x < y$: In R is the same.
2. $x > y$: In R is the same.
3. $x \leq y$: In R is $x <= y$.
4. $x \geq y$: In R is $x >= y$.
5. $x \neq y$: In R is $x != y$.

Logical and relational operators V

6. $x = y$: In R is $x == y$. Observe that in R (as well as in MATLAB, C, and many high-level languages) the = sign is assigment rather than mathematical equality whereas the == refers to mathematical equality. For example in R $x = x + 1$ makes sense whereas mathematically ($x \in \mathbb{R}$) it doesn't.

A remark about relational operators and floating point representation:

- In a computer, real numbers are represented by means of floating point representation (this representation is akin to the scientific notation). Thus, the machine representation of real numbers is finite, and extra care must be taken when comparing noninteger variables. As a homework look

Logical and relational operators VI

for information about the floating point representation, in particular, look for the term machine epsilon ϵ_M.

Control Structures I

An algorithm has a sequence of operations that consist of:

- Input.
- Output.
- Computation.

The word sequence implies that there is an order of execution. That order can be alter by means of the control structures. As an example consider the following sequential algorithm:

Control Structures II

Algorithm 1 Gas milage.

Input: S_m, E_m, and g_u are the start milage, end milage and galons used respectively.
$Distance \leftarrow E_m - S_m$
$A_{mpg} \leftarrow Distance/g_u$
Output: A_{mpg} average milage per galon.

Two types of instructions control the flow of an algorithm:

- Conditional statement.
- Iterative statement.

Conditional statement I

In this case we ask a question whose reponse is a boolean value. depending on the answer the next statemment (operation) is selected.

---

if Boolean expression is TRUE **then**
  First sequence of instructions
else
  Second sequence of instructions
end if

Note that conditional statements can be nested.

Conditional statement I

Examples and sintax:

Iterative statement I

In this case a block of statements (or one statement) is repeated while certain condition (boolean statement) is TRUE.

---

while Boolean expression is TRUE **do**
  Sequence of statements
end while

Note that iterative statements can be nested.

Iterative statemens

Examples: