

Introduction to R Software
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 10
Logical Operators

Welcome to that next lecture on the course introduction to R software. You may kindly recall that in the last lecture we started with the logical operators; we had taken several examples and through those examples we have try to understand the basic concepts and their execution in R console. We will continue with the same topic and if we were try to see what are the different other options available for the logic operators. So, now, just to have a quick review what we did in the earlier lecture, let us try to revise the different symbols for different logical operators for example, you may recall that we have used the greater than sign to indicate the greater than logical operator and then we had used the greater than or equal to sign.

(Refer Slide Time: 01:04)

Logical Operators and Comparisons

The following table shows the operations and functions for logical comparisons (True or False).

Operator	Executions
> ✓	Greater than
>= ✓	Greater than or equal
< ✓	Less than
<= ✓	Less than or equal
== ✓	Exactly equal to
!= ✓	Not equal to
! ✓	Negation (not)
&, && ✓	and
, ✓	or

Operator	Executions
<u>xor ()</u>	either... or (exclusive)
<u>isTRUE (x)</u>	test if x is TRUE
<u>TRUE</u>	true
<u>FALSE</u>	false

2

As another logical operator is less than sign for say less than logical operator and less than equal to and exactly equal to was denoted by 2 equal to signs and not equal to was denoted by exclamatory sign followed by equal to sign, the negation was be denoted by see that here exclamatory sign, and double denoting the and logical operator under different types of conditions and similarly with this bar and double bar they were also

indicating the or logical operator. And similarly we had done some other operator like that x or an inside the bracket that was used to test to be whether the statement for either or condition, and similarly we had use another statement is true that was used to test whether a statement called as x is true or not, and similarly true and false are the 2 words which are reserved for the logical operators and we cannot use it and say as variable or at any other place by our choice right ok.

(Refer Slide Time: 02:25)

```

Examples:
> x = 1:6 ✓ # Generates x=1,2,3,4,5,6
> (x > 2) & (x < 5) # Checks whether the values are greater than 2 and less than 5

x=1 1>2 False 1<5 Yes True
x=6 6>2 True 6<5 False False
[1] FALSE FALSE TRUE TRUE FALSE FALSE
x=3, 3>2 Yes True 3<5 Yes True } True
name x > 2 and x < 5 → values ?

> x[(x > 2) & (x < 5)] # Finds which values are greater than 2 and smaller than 5
[1] 3 4
x=1 1>2 X
x=2 2>2 X
x=3 3>2 Yes and 3<5 Yes → 3
x=4 4>2 Yes and 4<5 Yes → 4
x=5 5>2 Yes and 5<5 No → X
  
```

Now, after this quick revision let us try to take some more example and we try to understand how this logical operators are going to be executed. So, let me take the first example, where I am trying to generate the data from 1 to 6; that means, now x is my here variable which is containing the data 1 2 3 4 5 6; this is another approach to generate the data, another approach that we discussed earlier that was to write down this number 1 2 3 4 5 6 inside the bracket.

If using the combined command, but because this is another command that we use the same data and we are going to discuss about such commands in more detail later on. So, now, briefly I have a data containing 1 to 6 integers, and I want to check whether the values are greater than 2 and is smaller than 5.

So, my question is with this statement I want to check whether the values contained inside the x are greater than 2 and they are also less than 5. So, now, if I try to solve this equation let me take it to here x equal to here 1. So, now, 1 is greater than 2 this is false

this one is a smaller than 5 answer is yes true. So, now, the combination of false and yes this will yield false. This outcome is based on something called truth table that we are going to discuss after a couple of slides. Similarly in case if I try to suppose take here x equal to here 6, now the first statement to tells me whether 6 is greater than 2 answer is here true and.

Similarly, if I try to say here whether 6 is smaller than 5, the answer here is false. So, now, once again the combination of true and false this will give me the answer here false and which is denoted here. Similarly if I try to take here x equal to here 3 then the first condition said does 3 is greater than 2 answer is yes it is true and the second condition x less than 5 states whether 3 is a smaller than 5 answer is yes this is true. So, the combination of true and true will be true as per the truth table. So, and this answer is denoted over here right.

So, now you can see here we have operated the logical operators over a combination of 2 vectors. The first statement is containing some values in vector format and second statement x less than 5 is also containing the values in some vector format. Now let me introduce you here another command; now in this example we have seen that we have a decision in terms of true and false whether my statement is true or whether my statement is false and that is simply indicating whether a number is greater than 2 and the number is smaller than 5 simultaneously.

But now suppose I want to know what are the values in this vector which are satisfying this condition; that means, which are the values in this vector which are here greater than 2, and which are also a is smaller than 5.

That is my question now. In order to get such an answer I can write the logical a statement like as follows. So, in order to do so, what we do? That first we try to write down our requirement in the form of this condition here, suppose in this case I wanted to know that what are the values containing in the vector x are greater than 2, and what are the values contained in the vector x are a smaller than 5 and I want to know what are those values.

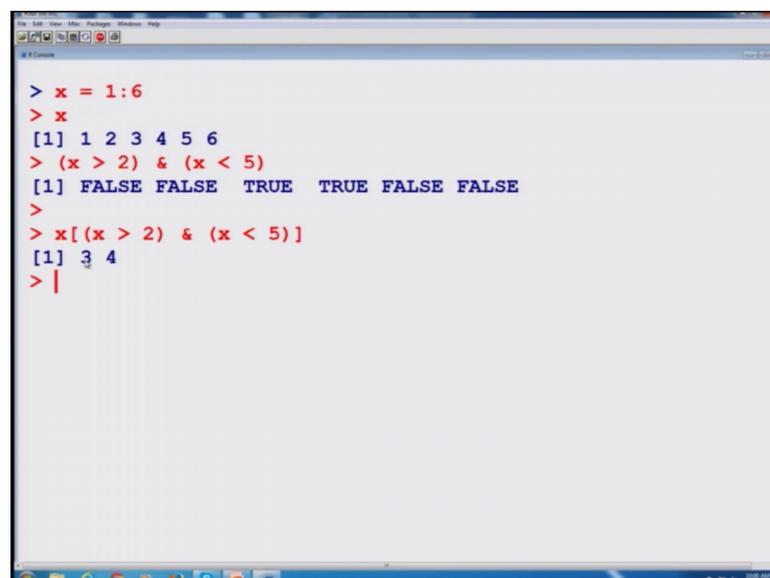
So, what I try to do? I try to write this condition inside the square bracket and here I try to write the variable name, which is here x and as soon as you try to do so, this will give me the numbers inside the vector x that are satisfying this condition.

That means the numbers which are greater than 2 and simultaneously they are a smaller than 5. Now let us try to solve this suppose the first value in x is here 1 is 1 greater than 2? Answer is no the condition is violated and this will not appear in my answer now I try to take here x equal to 2 whether 2 is greater than 2? No this condition is violated if I try to take here x equal to 3, 3 is greater than 2? and so, this yes and now we go to another condition does 3 is a smaller than 5 answer is yes.

So; that means, 3 is the number which is satisfying both the condition that 3 is greater than 2 as well as 3 is a smaller than 5 and similarly I try to take here 4, is 4 greater than 2 answer is yes and is 4 smaller than 5 answer is yes. So, 4 also satisfies this condition and similarly if you try to write for x equal to 5 and 6 you can see here is 5 greater than 2? Answer is yes and is 5 is smaller than 5? Answer is no so; that means, this also does not satisfy the condition and the same thing you can do for x equal to 6 and this results here these 2 values 3 and 4.

So, let us try to do this operation on the R console. So, I try to first generate these values over here.

(Refer Slide Time: 09:30)



```
> x = 1:6
> x
[1] 1 2 3 4 5 6
> (x > 2) & (x < 5)
[1] FALSE FALSE TRUE TRUE FALSE FALSE
>
> x[(x > 2) & (x < 5)]
[1] 3 4
> |
```

So, you can see here now this is giving me 6 integer values 1 2 3 4 5 6 and now I try to write down this condition over here, now you can see that here for 1 this is false, for 2 this is false, for 3 this is true, for 4 this is true, for 5 this is false and for 6 this is false so; that means, there are 2 value 3 and 4 for which this statement is true right. But suppose if

there are thousand values in your vector or say 5000 values in your vector, it is very difficult to count that what are the values for which the statement is true and what are the values for which the statement is false. So, in order to this command that end I would like to know.

What are those values which are satisfying my condition and this comes out to be 3 and 4 and this is the same thing which I have done in this slide and I have reported in the next slide. This is simply a screenshot of what we have done here.

(Refer Slide Time: 10:50)

```
Examples:
> x = 1:6 # Generates x=1,2,3,4,5,6
> (x > 2) | (x < 5) # Checks whether the values are greater than 2 or less than 5
[1] TRUE TRUE TRUE TRUE TRUE TRUE
x=1 1 > 2 False OR 1 < 5 True → True
x=2 2 > 2 False OR 2 < 5 True → True
> x[(x > 2) | (x < 5)] # Finds which values are greater than 2 or smaller than 5
[1] 1 2 3 4 5 6
```

Now let us try to the same example and try to see with other operators. We know that for the R operators we have used this vertical line that is available on your keyboard right.

So, now I want to know that out of this values x equal to 1 to 6, 1 2 3 4 5 and 6 what are the values contained in x which are greater than 2, but now here I am saying or what are the values in x which are a smaller than 5. So, if either of the condition is true my statement will result in the true now let us try to solve it.

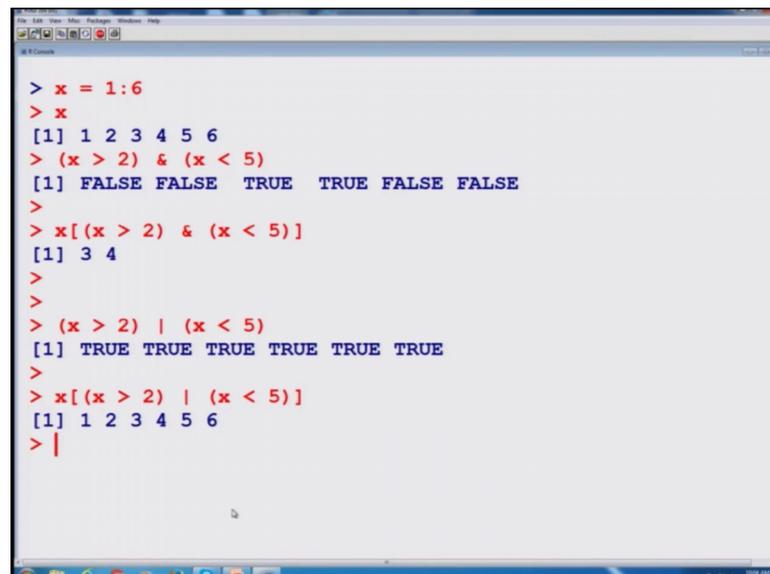
So, now suppose if I say here x equal to here 1, 1 is greater than 2? No this is false, but my logical operator say here is R. So, now, I am trying to say if one is a smaller than 5 answer is true yes this is correct. So, now, if any of the statement here or see here is true the answer will come out to be true because I am interested if any one of them is correct then my statement will be correct. In the case of and this becomes opposite similarly for

see here x equal to 2 I am trying to say here 2 is greater than 2 this is again false, but my joining operator is R and I try to see here whether 2 is smaller than 5 answer is true and again the answer comes out to be true.

So, similarly you can operate with x equal to 3 4 5 and 6 and can and you will see that here you get the same result here all values are here true because all the values are greater than 2 or they are a smaller than 5. Now similarly on the same lines as we did earlier I would like to know that here what are the values which are greater than 2 or a smaller than 5. In this earlier statement we had simply tested or checked the statement in terms of true or false, but now I want to know what are the values has satisfying this condition. So, again I try to do the same thing that I try to write down the condition here, same condition and this is enclose by the square brackets and I try to write down here the variable name and this is going to tell us that.

Which are the values that are greater than 2 or they are a smaller than 5 and you can see here, here I get this answer 1 2 3 4 5 and 6 and that we already have established analytically also that this statement is correct. Now let us try to do this thing in the R console so that we can see here what answer does this R gives to us.

(Refer Slide Time: 14:00)



```
> x = 1:6
> x
[1] 1 2 3 4 5 6
> (x > 2) & (x < 5)
[1] FALSE FALSE TRUE TRUE FALSE FALSE
>
> x[(x > 2) & (x < 5)]
[1] 3 4
>
>
> (x > 2) | (x < 5)
[1] TRUE TRUE TRUE TRUE TRUE TRUE
>
> x[(x > 2) | (x < 5)]
[1] 1 2 3 4 5 6
> |
```

So, I try to write in write down this condition and you can see here that everything is coming out to be true. Now I want to know what are the values. So, as I told you earlier

that instead of copying and pasting again and again I can also use by arrow keys downward arrow key and upward arrow key.

So, for example, here I use here upward arrow key and I get here the same statement and I can simply just modify it. So, that will save my time and you can see here that I get the same answer that 1 2 3 4 5 and 6 are the values, which are satisfying this condition right. So, let us try to come back to a slide and in the next slide I have again just clicking the screenshot of this operation and I have pasted it here. So, that you can try it yourself and you can verify whether you get the same outcome or not.

(Refer Slide Time: 15:04)

```
Examples:
> x = 1:6      # Generates x = 1,2,3,4,5,6
> (x > 2) | (x > 10) # Checks whether the
                    # values are greater than
                    # 2 or greater than 10
[1] FALSE FALSE TRUE TRUE TRUE TRUE
    x = 1    1 > 2 No    1 > 10 NO → False
        2
        ⋮
        6

> x[(x > 2) | (x > 10)] # Finds which values
                        # are greater than 2 or
                        # smaller than 10
[1] 3 4 5 6
```

Now I try to take here one more example similarly on the same lines, I am giving you here, but I would ask you that you please try to execute it with your own hand. So, here again I am trying to change my conditions from the earlier example.

That I am trying to say here that now x is the same vector containing the value 1 2 3 4 5 and 6, and I want to check whether the values contained in the vector x are greater than 2 and this operator is for R or the values contained in the vector x are greater than 10. So, earlier I had taken the condition is a smaller than 5 and, but now I am taking the condition x greater than 10. Now I would ask you please try to do this exercise over here try to take x equal to 1 and we can see here that this one can be greater than 2? No answer is no and similarly I can check whether one is greater than 10? Answer is no for

the combination of no and no that is false and false that is going to be here false and this is denoted here.

Similarly, I would request you to do the entire exercise yourself and try to check it for x equal to 2 3 4 up to 6 and similarly in the next statement I am simply trying to find out here that what are the values out of this 1 2 3 4 5 and 6 which are contained in x vector which are greater than 2 or they are smaller than 10. And we get here an answer here 3 4 5 and 6, and in the on the next screen I have taken only the screenshot of the R console where I am trying to get this outcome and I would request you that you please try to do it with your own hand so that you can gain some confidence also.

(Refer Slide Time: 17:10)

Logical Operators and Comparisons

Operator	Executions
&, &&	and
,	or

Shorter form → *longer form*

- The shorter form performs element-wise comparisons in almost the same way as arithmetic operators.
- The longer form evaluates left to right examining only the first element of each vector. Evaluation proceeds only until the result is determined.

9

Now, just coming on say another aspect, this is the same slide that we used in the earlier lecture and I had explained that when I am trying to use the operator here and there are 2 options one to use say this symbol or to use here double symbol and similarly for the or here I am using the single symbol and here I am trying to use the double symbol, and we had discussed that this is the shorter form and this we are calling as a longer form and the same thing here also right. So, we had discussed that these are shorter form. So, writing only one letter this performs the element wise comparison in almost the same way as we try to do with any arithmetic operator and when we are trying to use the longer form.

Then it evaluates the entire expression from left to right, but it examines only the first element of the each vector what does this mean. So, in order to understand the meaning

of this statement, now I am going to take an example and I would try to show you that how it is executed and what is really happening inside r.

(Refer Slide Time: 18:31)

Example of "The longer form evaluates left to right examining only the first element of each vector"

```
> x = 1:6      # Generates x = 1,2,3,4,5,6
```

```
> (x > 2) && (x < 5)
```

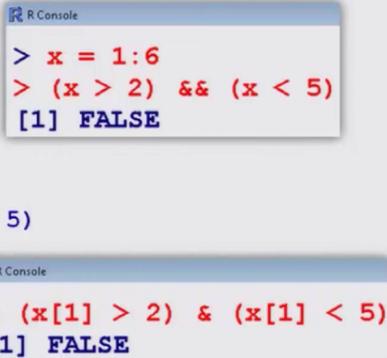
```
[1] FALSE
```

is equivalent to:

```
> (x[1] > 2) & (x[1] < 5)
```

```
[1] FALSE
```

Note that `x[1]` is only the first element in `x`



10

So, I again take the same data set that I have here the values `x` containing 1 2 3 4 5 and 6, and now I am trying to use here double operator double and sign. And my condition is the same that we used in the earlier example that `x` is greater than 2 and `x` is a smaller than 5; that means, I want to know whether the values contained in the vector `x` are greater than 2 and simultaneously they are a smaller than 5 and when I try to do so, I get here this type of answer false.

Let me try to show you here otherwise you may not believe on me.

(Refer Slide Time: 19:38)

```
> x = 1:6
> x
[1] 1 2 3 4 5 6
> (x > 2) & (x < 5)
[1] FALSE FALSE TRUE TRUE FALSE FALSE
>
> x[(x > 2) & (x < 5)]
[1] 3 4
>
>
> (x > 2) | (x < 5)
[1] TRUE TRUE TRUE TRUE TRUE TRUE
>
> x[(x > 2) | (x < 5)]
[1] 1 2 3 4 5 6
> |
```

So, I try to copy this data in the R console and let us try to find it out here right and then you can see here you are getting this thing. On this is screen I would like to have your attention over this which I am highlighting. If you try to see I have executed here the condition with a single and sign and I am getting here 6 words for true and false whereas, here I am using the same statement, but I am using here 2 and sign, but here I am getting only one word that is false why this is so, do not you think that this is very puzzling.

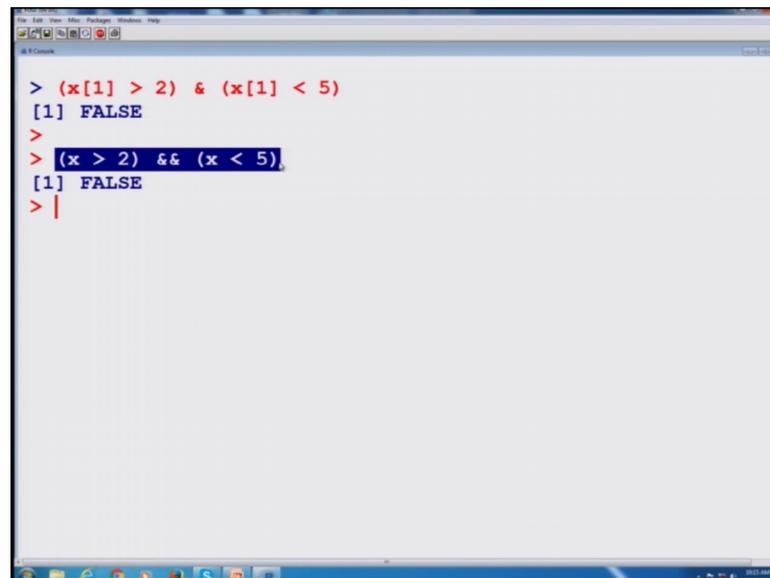
That x has 6 values and here I am getting 6 outcomes whereas, here I am getting only one outcome, and if you try to see this is the same outcome which I have copied and pasted here. So, now the question is what is really happening? Let us try to understand when we are trying to use here this double symbol and we are using here a vector ; that means, as I said this is examining only the first element of each vector; that means, the first element of x vector here is only here one. So, I can write this statement equivalently as say the first element x 1 is greater than 2, and x 1 is a smaller than 5.

So, now you can see here that this is saying one greater than 2 and this is saying one is smaller than 5; one cannot be greater than 2, but one is smaller than 5. So, this answer is false and this answer here is true and the outcome here is false. So, that when we are trying to use the double symbol, this operator is not continuing with the remaining elements in the vector, but it is operating only on the first element; and same thing if you try to see here I have done it here and I can show you here over the R also whether this

holds true or not. So, I try to copy this statement, but before that let me try to show you here that what do we get with x 1 I will just write.

So, you can see here with x 1 I am getting the first element in the vector x, now what I try to do here that I try to type out type this command and you can see here this is the same thing which we have given in the outcome and this is the same output which was obtained by the earlier a statement like this right.

(Refer Slide Time: 22:52)



```
> (x[1] > 2) & (x[1] < 5)
[1] FALSE
>
> (x > 2) && (x < 5)
[1] FALSE
> |
```

So, now this is clear to us what do we really mean by the single operator and double operator right. And now in case if I want to know that what are the values which are satisfying this thing then I can use the earlier command that first I try to write down the conditions here and this condition is being enclosed by this a square bracket, but note that here I am using here the double and sign, but it is giving me here the integer 0; that means, there are no integers which are satisfying this thing.

This means here what? Because this statement is operating over only the first element and it is something like this I want to know whether x 1 is the first element is greater than 2 and the first element in x is a smaller than 5, and whether this condition holds to simultaneously or not my first value here is one is this greater than 2 and my first value here is 1 which is smaller than 5 is it possible one cannot be greater than 2. So, this is false one is a smaller than 5 this is true, but here we having a condition add that we both

these conditions have to hold simultaneously, and that is why there is no integer which is satisfying this condition and I am getting here the value 0.

So, now after this slide going to stop here and now you have seen that we have considered different types of combination of the logically statement, and we have tried to test their truthfulness whether the statement is true or false. Now one thing is remaining that I want to give you the details about the truth table what is truth table and how it is created. So, that if we will try to do in the next lecture and in between you try to take such more example and try to execute them with your own hand on your own computer till then goodbye.