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

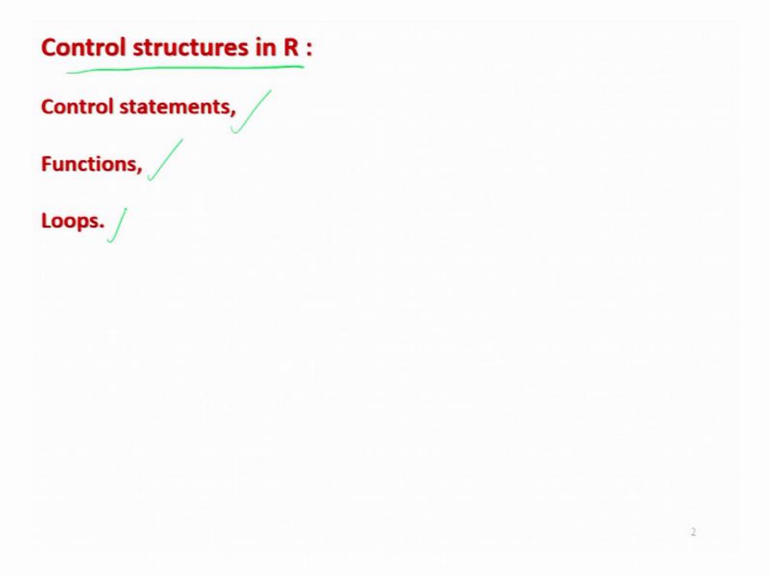
Basics of Calculations
Lecture - 23
Loops - while and repeat

Hello friends, welcome to the course Foundations of R Software. You can recall that in the last lecture we started a discussion on the topics of loops and we had discussed the for loop. We had understood that how for loops works and how it gives the outcome.

So, I believe that now you have understood functioning of the loop and in this lecture we are going to consider two more loops, they are while loop and another is repeat loop. So, as we had seen that in the case of for loop the for loop can be used in those places where the number of times the program has to be repeated is known in advance.

Similarly, this while and repeat loops they also have their own utility under different types of conditions. So, let us try to begin this lecture and we try to understand the basic functioning of while loop and repeat loop and I will try to take some examples. So, that I can explain you that how do they work and how do they give the outcome. So, let us begin our lecture.

(Refer Slide Time: 01:29)



Control structures in R :

Control statements, ✓

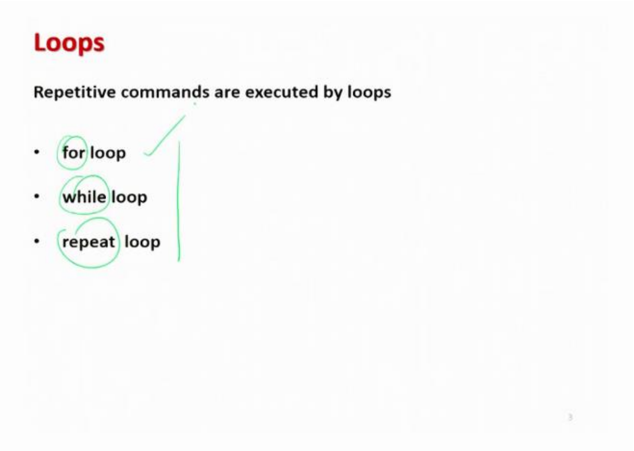
Functions, ✓

Loops. ✓

So, as you can recall that we are discussing about the control structures in R and we already have talked about the control statements; this conditional executions like if statement, if else statement, if else, if statements, etc. Similarly we have considered the functions also like a switch and which and now we are on the loops.

So, in the loops we already have covered the for loop.

(Refer Slide Time: 01:54)



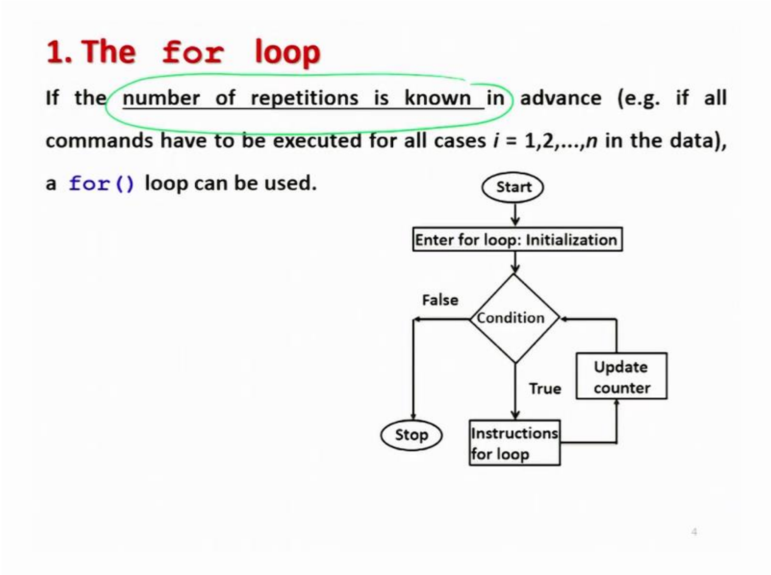
Loops

Repetitive commands are executed by loops

- for loop
- while loop
- repeat loop

Just for the sake of quick review I will just show you the command. So, under the, these loops we are going to consider these three loops for loop, while loop and repeat loop.

(Refer Slide Time: 02:07)

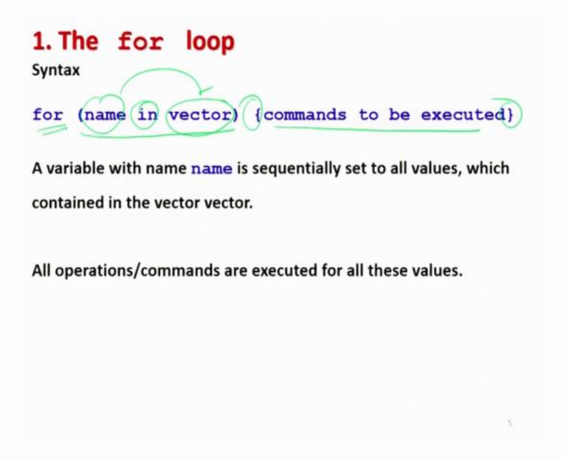


1. The `for` loop

If the number of repetitions is known in advance (e.g. if all commands have to be executed for all cases *i* = 1,2,...,*n* in the data), a `for()` loop can be used.

So, for so the for loop the way it is work is that the for loop is going to be used when the number of repetitions are known in advance; that means, you know that how many times the program has to be repeated.

(Refer Slide Time: 02:21)



1. The `for` loop

Syntax

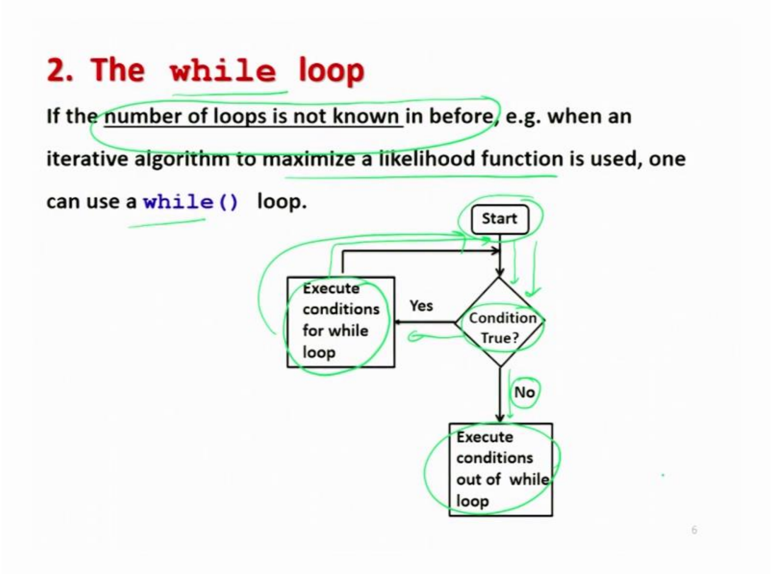
```
for (name in vector) {commands to be executed}
```

A variable with name `name` is sequentially set to all values, which contained in the vector vector.

All operations/commands are executed for all these values.

And then the command for the for loop was like you try to write down here for, that inside the parenthesis you try to write down the controlling variable as name, then in and then the vector of values that you want this name to choose. And then whatever commands you want to execute they are written inside the curly bracket here like this. So, that is what we had done in the last lecture.

(Refer Slide Time: 02:44)



2. The `while` loop

If the number of loops is not known in before, e.g. when an iterative algorithm to maximize a likelihood function is used, one can use a `while()` loop.

Start

Condition True?

Yes

Execute conditions for while loop

No

Execute conditions out of while loop

So, now we try to consider here another loop which is called as while loop so, as the meaning of the loop suggests while; that means, this loop is going to be executed as long as the condition remains TRUE. So, in this case what will happen that the number of times you want to repeat the loop that is not known to us. So, that is why we try to give here a condition in terms of while, right.

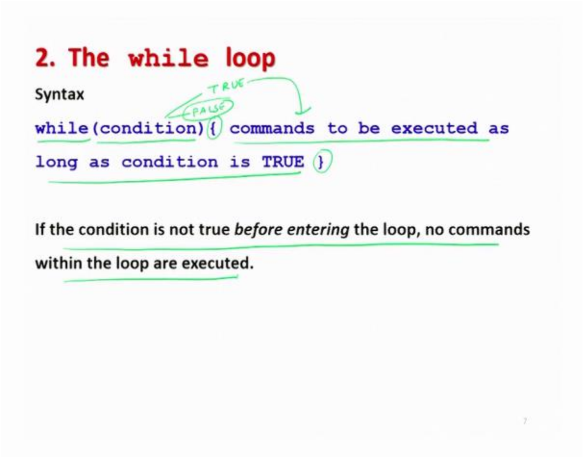
So, one of the basic fundamental things what you have to keep in mind that this while loop is used when the number of loop is not known in advance, right. For example, when we are trying to use some iterative algorithms; that means, the algorithm has to work till we get a convergence. For example, when we try to use the maximum likelihood estimation in the statistic then we try to use this type of algorithm because, we do not know that whether the convergence of the algorithm is going to be achieved after 5000 repetition or 7000 repetitions, right.

So, we try to give there a condition like that, ok try to repeat the algorithm till you get a convergence. So, for that we try to use the while loop. So, the flow chart of the while loop is as follows that we start and then there is a condition and this condition is actually checked in terms of TRUE and FALSE. If this condition is TRUE then the conditions or the statements or the commands that we want to execute they are executed.

And then it will, because the condition is TRUE it will go back to the initial place and then once again it will check whether the condition is TRUE or not. And then in case it comes out to be yes then once again this these statements are executed and we come back to the start point. And then once again it comes to the conditions to be checked whether it is TRUE or FALSE and suppose after some time the condition becomes FALSE. Answer comes out to be a no that case try to execute those statements, which are written outside the while loop and you stop the program or try to do whatever has been asked in the program.

So, you can see here it is not really known before the start of the program that how many times this condition is going to be TRUE. So, that is why we do not specify any particular number for which the program has to be executed, but we try to give that number in terms of a condition that you try to execute the program as long as this condition is TRUE.

(Refer Slide Time: 05:25)



2. The while loop

Syntax

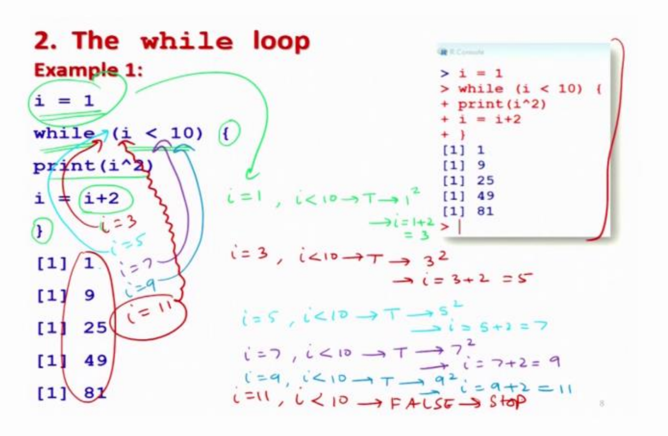
```
while(condition){ commands to be executed as
long as condition is TRUE }
```

If the condition is not true *before entering* the loop, no commands within the loop are executed.

So, the way in which this while loop is written in the R software is as follows. We try to write down here while and then inside the parenthesis we write the conditions. Now this condition is going to be either TRUE or it is going to be FALSE. So, whatever you want to be executed when the condition is TRUE that is written just after this inside the curly bracket, right.

So, all these commands which are written inside the curly bracket they are going to be executed as long as the condition remains TRUE and as soon as the condition becomes here FALSE. Then the control will come out of this commands, which are written inside the curly bracket. And in case if the condition is not TRUE before entering the loop then no command within the loop are executed, right.

(Refer Slide Time: 06:19)



2. The while loop

Example 1:

```
i = 1
while (i < 10) {
print(i^2)
i = i+2
}
[1] 1
[1] 9
[1] 25
[1] 49
[1] 81
```

```
> i = 1
> while (i < 10) {
+ print(i^2)
+ i = i+2
+ }
[1] 1
[1] 9
[1] 25
[1] 49
[1] 81
> |
```

So, let me try to take here some examples and try to explain you that how it works. So, these are very simple examples and I have tried my best to take a similar example what we have taken in the case of for loop. So, that you can understand it better and you can also compare.

So, here also I want to print some numbers you may recall that in the case of for loop we have printed the numbers 1 2 3 4 5 right. So, I try to write here a command which can print all the integers which are smaller than 10. So, I try to take of the integer say i equal to 1, so this is just like a counter actually.

And then I try to write down here the while command while i is less than 10 and then whatever command I want to give they are given inside this curly brackets like this. So now, my commands are print i square and then after printing try to replace i by i by plus 2. That means, you try to add 2 in the value of I that you have used earlier and then try to check whether the new value i is smaller than 10 or not in case if the condition is TRUE then you try to print otherwise not.

So, now let us try to see how it works. So, the first value of i which is going to be considered here is i equal 1, which is given here you that you can see. Now after this it tries to take the condition that i is less than 10 the answer is come comes out to be TRUE. And when it is TRUE this will try to print here the value of i which is here 1 square and then it will try to choose i equal to 1 plus 2 which is equal to here 3, right.

And now after it has chosen here say i equal to here 3, then this control will go back to this while condition. And then it will try to take this value of i and it will try to check whether i is less than 10. So, 3 is less than 10 the condition is TRUE, so it will print here 3 square. And then I will be replaced by here 3 plus 2 which is equal to here 5.

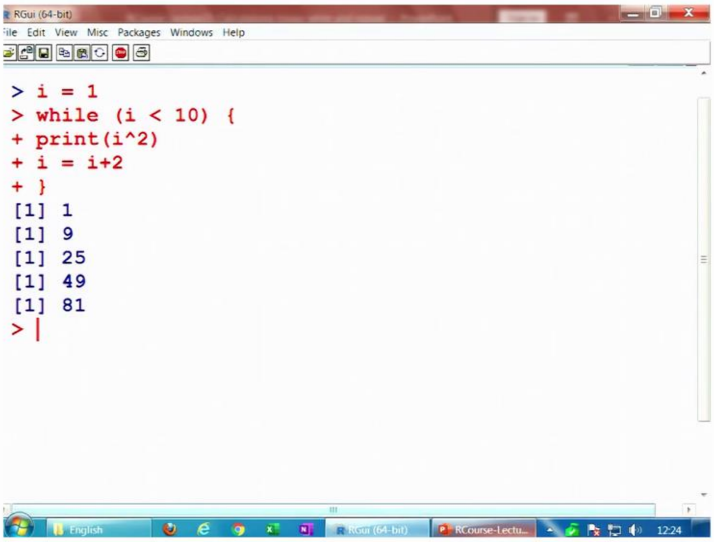
And now this i equal to 5 is here this is once again taking to the while loop and while loop considers here i equal to 5 and it tries to check whether i is smaller than 10, TRUE then it will try to print here 5 square. And i is going to be replaced by 5 plus 2 which is equal to here 7.

Now, you are here at a position where now i is equal to here 7. So now, once again this condition is brought to the while loop and it tries to take here that whether this value of i equal to 7 is smaller than 10. Answer comes out to be here TRUE and you print here 7 square. And i becomes here 7 plus 2 equal to here 9. And now after this next this i equal to 9 is consider and it is transported back to while condition.

So, i equal to 9 is there and it tries to check whether i is smaller than 10 answer comes out to be a TRUE and then nine square is printed. And then i is replaced by 9 plus 2 equal to 11. Now you have to be watchful what happens. Now i here is 11 this value of i is travel is transported to while condition. And now i equal to 11 is checked whether i is less than 10 answer comes out to be here FALSE and FALSE then the program stops here, right.

So, that is how you can see here these values are printed here and this is here the screenshot. So, this is how the while loop works right. So, before I move forward let me try to give you this example on the R console. So, that you become more confident and then I will try to give you here one more example in which I am going to once again ask for your help.

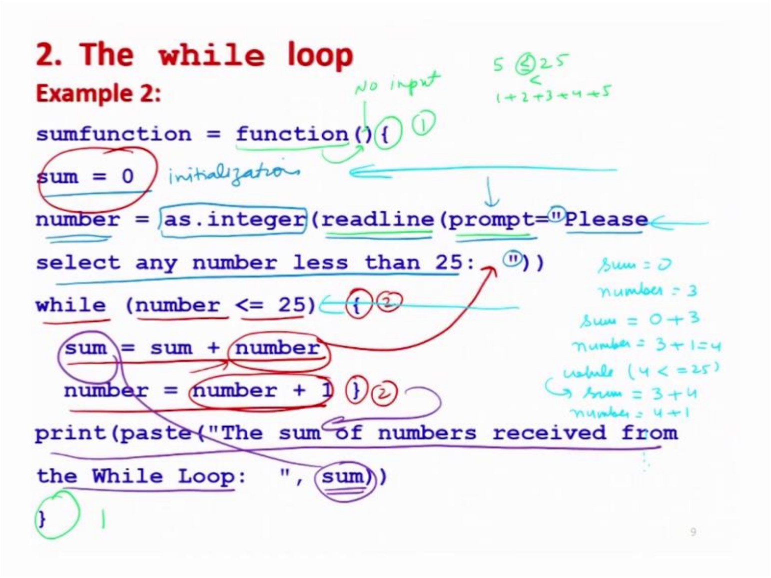
(Refer Slide Time: 10:55)



```
> i = 1
> while (i < 10) {
+ print(i^2)
+ i = i+2
+ }
[1] 1
[1] 9
[1] 25
[1] 49
[1] 81
> |
```

You can see here I am just copying and pasting this command and you can see here the outcome is coming to out to be here like this right. So, that is what is happening.

(Refer Slide Time: 11:08)



2. The while loop

Example 2:

```
sumfunction = function(){
sum = 0
number = as.integer(readline(prompt="Please
select any number less than 25: "))
while (number <= 25){
  sum = sum + number
  number = number + 1 }
print(paste("The sum of numbers received from
the While Loop:  ", sum))
}
```

So, now I try to take here one more example. And yeah, once again I am going to use here the concept of here function which I use in the case of for loop. So, since I have already explained you briefly that how this function works. So, I do not need to repeat it again, but as I promised I will try to take up this function topic in the next lecture, right.

So, what is my objective here my objective here is that I want to input a number and this number has to be less than or equal to 25. And then whatever number I am entering, I want to find out the sum of the number beginning from 0 to that number, right. Say for example, if I enter a number see here 5 which is less than or equal to 25 that is less than than actually.

So now, I want to find out the sum of 1 2 3 4 and 5, right. So now, how to do it well here in this program, I am going to use here simple a some more commands here which I will try to explain you that what happened ok that is command is here read line. So, means you know that when you want to execute a program sometime you have seen that as soon as you go further it will ask you something and then you will read the question and then you will try to give the input.

So, for that job actually this read line command is used along with the prompt right. So, these two commands I will try to consider later on, but here I can just show you that what will happen. And once you see the functioning you will understand it very easily, ok. So, I try to write down here a function. So, I try to write down here a function and then yeah there is no input here because, I am asking the input inside the program.

So, I can leave it blank, but I have to give the parenthesis and whatever I want to write this I am trying to write down in this parenthesis number 1, right. Now after this I try to take here a initial variable sum, so this is initialization, right. And then after that I choose here a number and this number is actually the number which I want someone to enter and the objective is that the sum up to this number is going to be found.

So, this number has to be an integer that is my requirement. So, I try to use here the command here as dot integer now you tell me do I really need to explain you what is the meaning of as dot integer. This is simply trying to consider an input variable as an integer that you know, right, because you have done such types of commands earlier.

So, after this then I use here read line and then I write down here prompt and then within double quotes I try to write down here a statement please select any number less than 25, right. So, what will happen when you try to execute the program the program will stop here and it will try to prompt that you please try to first enter the value.

And after this I try to write down the conditions what I want to execute under the while loop. So, I try to write down here while and I want to give a condition that as long as the number is less than or equal to 25 then try to execute these conditions. So, what are those statements which I want to execute, which are they are written here inside the curly bracket number 2. That the first statement I want to execute is sum is equal to sum plus number.

So, initial value of sum I have taken it to be here 0. So, it will start from 0 and then whatever number I am entering here that is going to be here and then I am trying to execute here number is equal to number plus 1. So, whatever the number has been entered this is used in the first line and then this number is going to be replaced by number plus 1, so 1 is added.

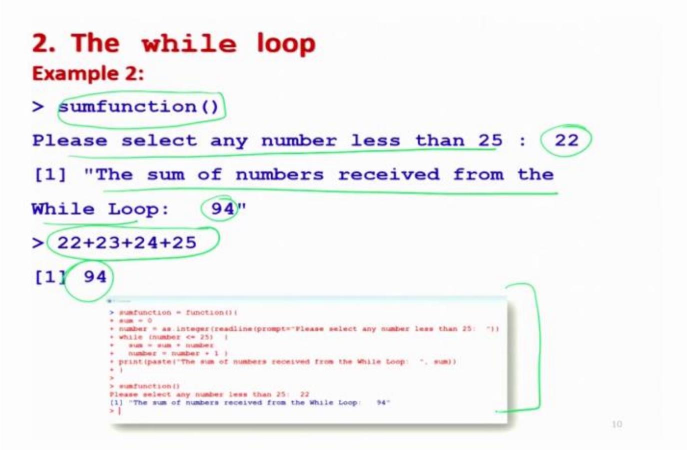
And after that when this job is completed when this as long as this the condition under the while loop is satisfied this is going to be executed, after that it will come out of the while loop and it will try to print here the sum of number receive from the while loop is whatever is the value of here sum this is obtained here. So, well I am using here some more concept of programming and then like as a print paste, but believe me these are very simple commands.

And I promise you that a just after a couple of lectures you will be comfortable with all the things and yeah these are very simple things. So, I am sure that it is not difficult for you to understand them. So, now if you try to understand the functioning of this number what will really happen?

Suppose, if you try to and execute the program, the program will come to the first line and it will try to take here sum is equal to 0. Then it will come to the second line and it will try to ask a number, suppose I try to give here a number see here 3. So, now the condition here is this while the number is less than 25 then try to execute the condition.

And what are those condition there are two statement which I want to execute sum is equal to sum plus number. So, the sum value here is 0 and then the number here is 3 and then this number will become here say number which is here 3 plus 1, 4. So, now this 4 is going to be once again checked, under the while condition that whether while 4 is less than or equal to 25, is this correct? 4 is less than or equal to 25 this is TRUE. So, once again this program will execute and now the sum is going to be means that you have taken earlier the initial value is now here 3 and plus the number, with number now becomes here 4, right. And then after that the number is going to be added with 1. So, number is now here 4 plus here 1 and then this condition will once again go to the while condition and this process will continue till my numbers are less than equal to 25, right.

(Refer Slide Time: 18:18)



2. The while loop

Example 2:

```
> sumfunction()
Please select any number less than 25 : 22
[1] "The sum of numbers received from the
While Loop:   94"
> 22+23+24+25
[1] 94
```

```
> sumfunction = function(){
+ sum = 0
+ number = as.integer(readline(prompt="Please select any number less than 25:  "))
+ while (number <= 25)  {
+   sum = sum + number
+   number = number + 1 }
+ print(paste("The sum of numbers received from the While Loop:  ", sum))
+ }
>
> sumfunction()
Please select any number less than 25:  22
[1] "The sum of numbers received from the While Loop:   94"
> |
```

10

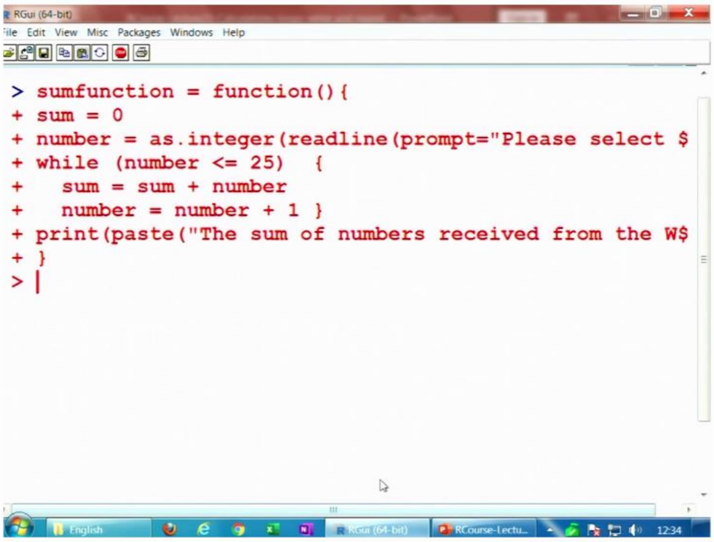
So, for example if I try to execute this program on the R console you can see here what is happening.

I try to just say here some function and parenthesis it ask me please select any number less than 25, so I give here suppose 22. So, now what will happen? It will try to print all the numbers which are more than 22. So, they are 22, 23, 24 and 25 because my condition is that please try to repeat the loop while the number is less than equal to 25.

So now, it will give me the number here the outcome here is this is the sum of number received from while loop it is printed as such. And it and whatever is the value here that is coming out to be this is 94 and it will print here like this and you can verify that the sum of 22, 23, 24, and 25 is 94 and this is here the screenshot.

So, let me try to first execute this program on the R console. So, that you become confident that these things are working and I am trying to copy these commands to save the time.

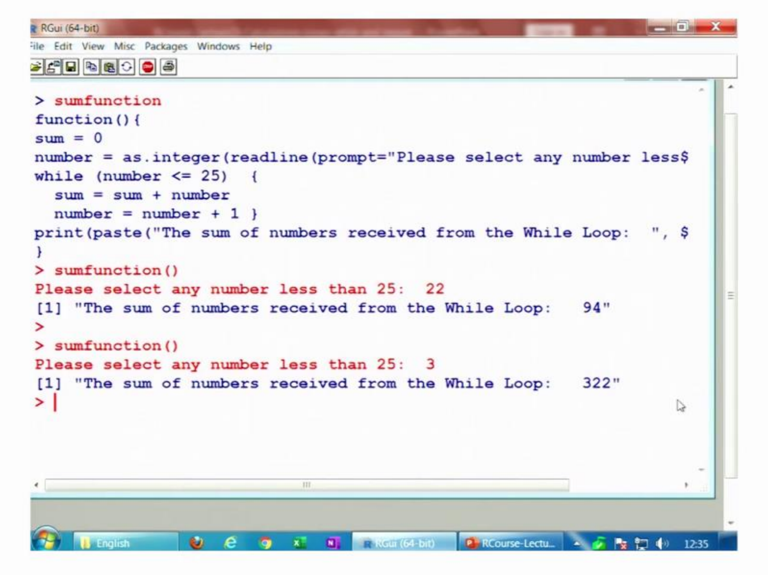
(Refer Slide Time: 19:23)



```
> sumfunction = function(){
+ sum = 0
+ number = as.integer(readline(prompt="Please select $
+ while (number <= 25)   {
+   sum = sum + number
+   number = number + 1 }
+ print(paste("The sum of numbers received from the W$
+ }
> |
```

And you can see here it is copied here.

(Refer Slide Time: 19:29)

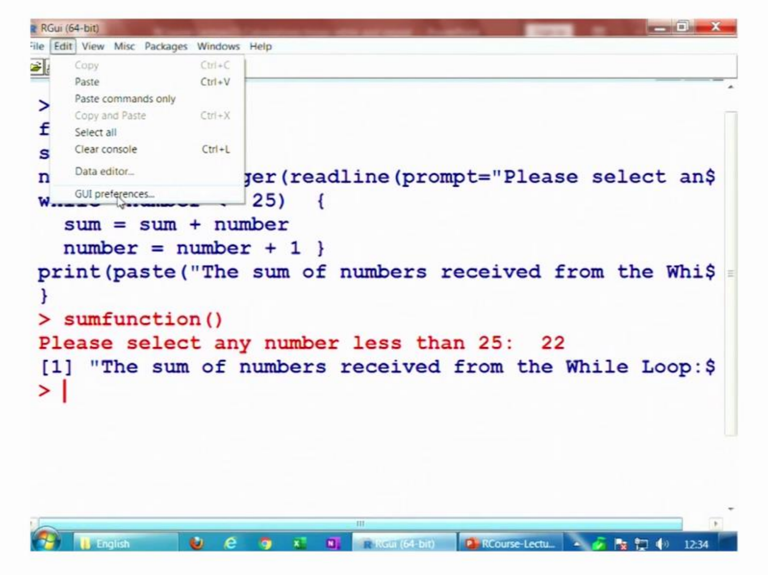


```
> sumfunction
function(){
sum = 0
number = as.integer(readline(prompt="Please select any number less$
while (number <= 25)  {
  sum = sum + number
  number = number + 1 }
print(paste("The sum of numbers received from the While Loop:  ", $
}
> sumfunction()
Please select any number less than 25:  22
[1] "The sum of numbers received from the While Loop:   94"
>
> sumfunction()
Please select any number less than 25:  3
[1] "The sum of numbers received from the While Loop:   322"
> |
```

And if you try to see here this function f u n c t i o n. you can see here this is here like this is what I have printed.

And if I want to execute it suppose I write down here sum function and the parenthesis and I enter it is asking me please select any number less than 25. So, I say 22 and this gives me an answer here the sum of number received from the while loop I try to make here this font is smaller.

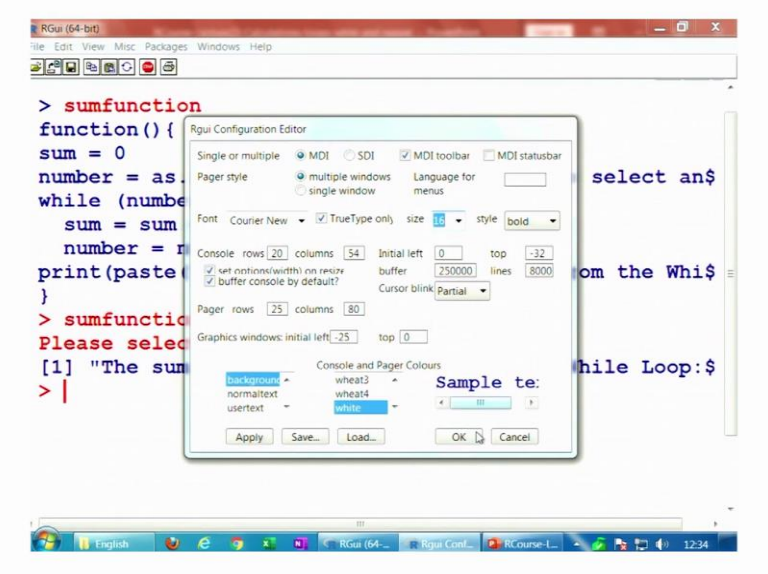
(Refer Slide Time: 19:58)



```
>
f
s
n                  ger(readline(prompt="Please select an$
w                   25)   {
  sum = sum + number
  number = number + 1 }
print(paste("The sum of numbers received from the Whi$
}
> sumfunction()
Please select any number less than 25:   22
[1] "The sum of numbers received from the While Loop:$
> |
```

So, that you can see the screen.

(Refer Slide Time: 19:59)


RGui (64-bit)
File Edit View Misc Packages Windows Help
Rgui Configuration Editor

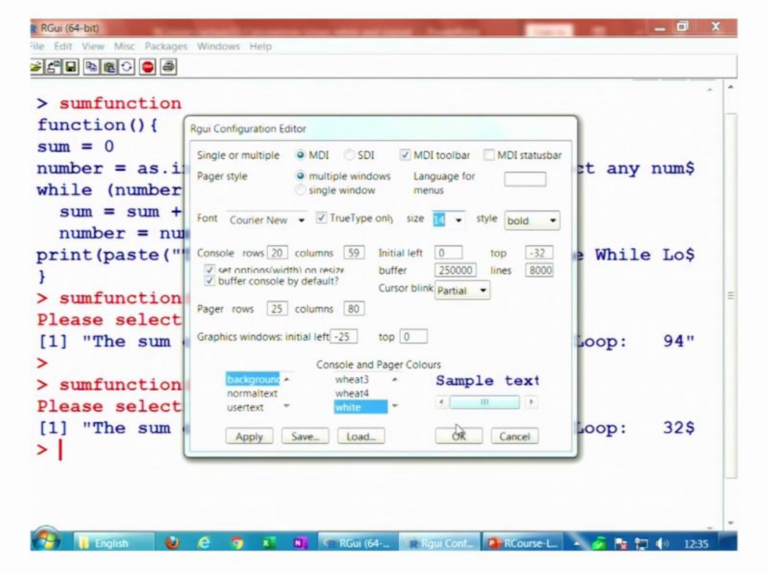
```
> sumfunction
function(){
sum = 0
number = as.                                    select an$
while (numbe
   sum = sum
   number = r
print(paste(                                    om the Whi$
}
> sumfunctic
Please selec
[1] "The sum                                    hile Loop:$
> |
```

Single or multiple ● MDI ○ SDI ☑ MDI toolbar □ MDI statusbar
Pager style ● multiple windows ○ single window Language for menus
Font Courier New ☑ TrueType only size 16 style bold
Console rows 20 columns 54 Initial left 0 top -32
☑ set options(width) on resize ☑ buffer console by default? buffer 250000 lines 8000 Cursor blink Partial
Pager rows 25 columns 80
Graphics windows: initial left -25 top 0
Console and Pager Colours
background normaltext usertext — wheat3 wheat4 white — Sample te:
Apply Save... Load... OK Cancel

It is here 94, right. And similarly if you try to repeat this function or this program once again you try to give here this number please select any number less than 3. So now, you can see what is the use of this read lines and prompt I give here 3.

So, this here this number is here like this. So, I have to make this screen more smaller.

(Refer Slide Time: 20:28)

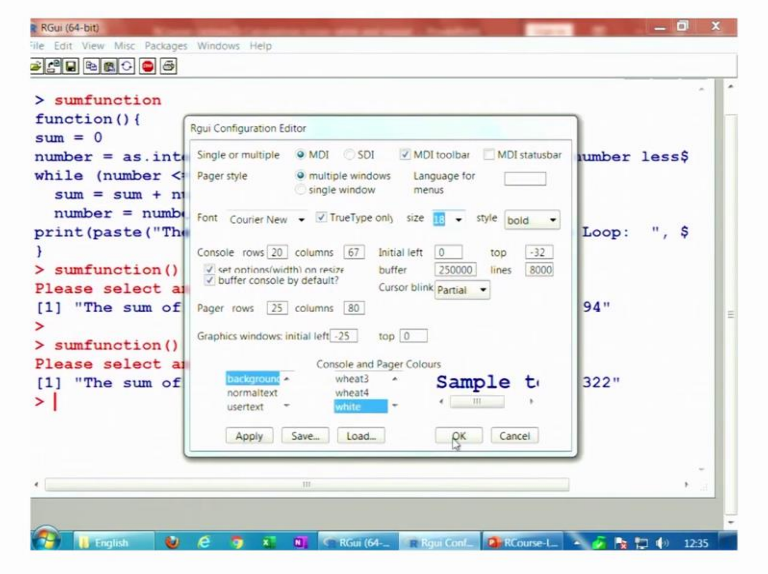

RGui (64-bit)
File Edit View Misc Packages Windows Help
Rgui Configuration Editor

```
> sumfunction
function(){
sum = 0
number = as.i                                   t any num$
while (number
   sum = sum +
   number = nu
print(paste("                                   e While Lo$
}
> sumfunction
Please select
[1] "The sum                                    Loop:    94"
>
> sumfunction
Please select
[1] "The sum                                    Loop:    32$
> |
```

Single or multiple ● MDI ○ SDI ☑ MDI toolbar □ MDI statusbar
Pager style ● multiple windows ○ single window Language for menus
Font Courier New ☑ TrueType only size 14 style bold
Console rows 20 columns 59 Initial left 0 top -32
☑ set options(width) on resize ☑ buffer console by default? buffer 250000 lines 8000 Cursor blink Partial
Pager rows 25 columns 80
Graphics windows: initial left -25 top 0
Console and Pager Colours
background normaltext usertext — wheat3 wheat4 white — Sample text
Apply Save... Load... OK Cancel

Because this number is bigger; so, you can see here this number comes out to be a 322 right. So, this is how this program will work now, right.

(Refer Slide Time: 20:44)



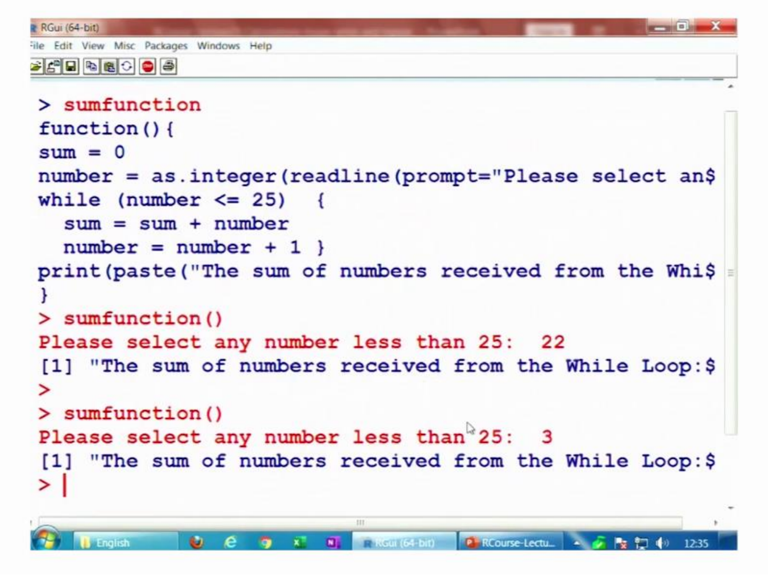
```
> sumfunction
function(){
sum = 0
number = as.inte                                    umber less$
while (number <=
  sum = sum + nu
  number = numbe
print(paste("The                                    Loop:  ", $
}
> sumfunction()
Please select an
[1] "The sum of                                     94"
>
> sumfunction()
Please select an
[1] "The sum of                                     322"
> |
```

Rgui Configuration Editor

Single or multiple ● MDI ○ SDI ☑ MDI toolbar □ MDI statusbar
Pager style ● multiple windows ○ single window Language for menus
Font Courier New ☑ TrueType only size 18 style bold
Console rows 20 columns 67 Initial left 0 top -32
☑ set options(width) on resize buffer 250000 lines 8000
☑ buffer console by default? Cursor blink Partial
Pager rows 25 columns 80
Graphics windows: initial left -25 top 0
Console and Pager Colours
background normaltext usertext wheat3 wheat4 white Sample t

So, now we come to another aspect of this thing and I can make this.

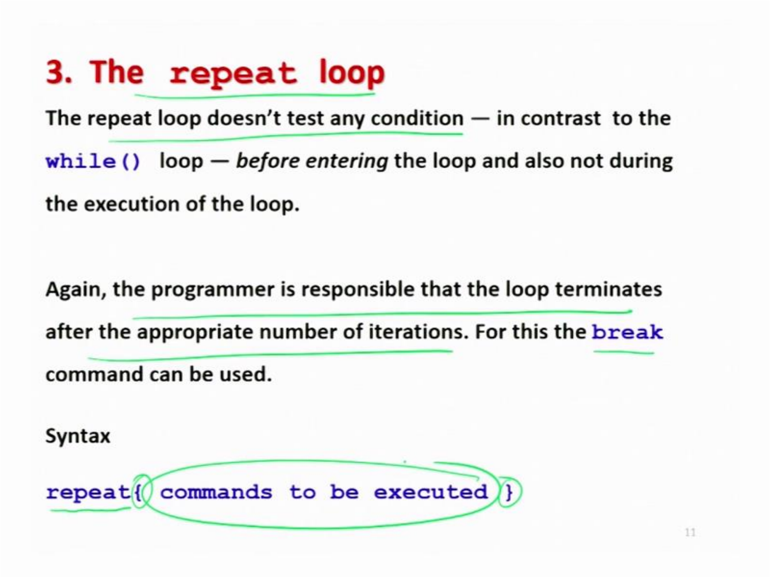
(Refer Slide Time: 20:47)



```
> sumfunction
function(){
sum = 0
number = as.integer(readline(prompt="Please select an$
while (number <= 25)  {
  sum = sum + number
  number = number + 1 }
print(paste("The sum of numbers received from the Whi$
}
> sumfunction()
Please select any number less than 25:  22
[1] "The sum of numbers received from the While Loop:$
>
> sumfunction()
Please select any number less than 25:  3
[1] "The sum of numbers received from the While Loop:$
> |
```

Font size clear. So, that next time when I try to show you the things are looking clear on the screen right.

(Refer Slide Time: 20:54)



3. The repeat loop

The repeat loop doesn't test any condition — in contrast to the `while()` loop — *before entering* the loop and also not during the execution of the loop.

Again, the programmer is responsible that the loop terminates after the appropriate number of iterations. For this the `break` command can be used.

Syntax

```
repeat{ commands to be executed }
```

11

.

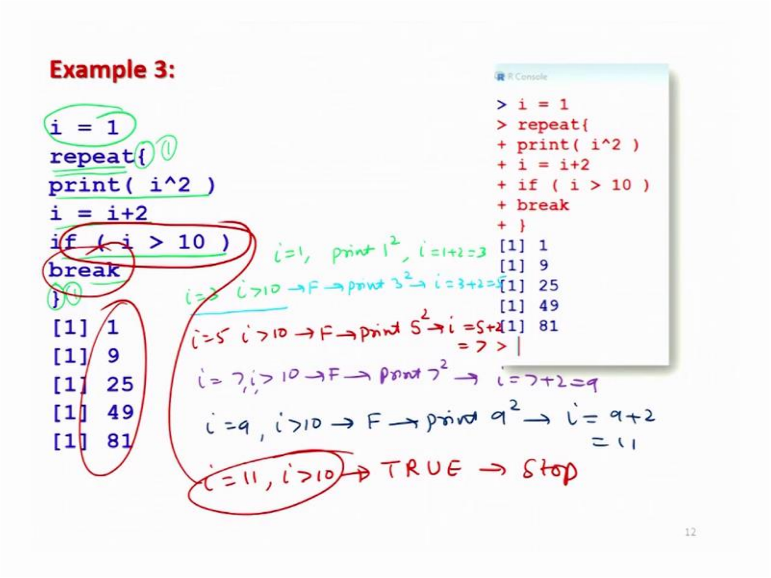
After this I come to our next loop which is here the repeat loop now in the case of for and while you have seen that they are dependent on the condition. That how many times the function has to be or the commands have to be repeated that is known or unknown. Now this repeat loop is independent of these condition right and repeat loop does not test any condition for example, in the case of while loop, right.

But it is dependent on the programmer that how many times the program has to be repeated or the commands have to be repeated. And in this case we do not also know that at which point of iteration the condition is going to be satisfied or not. So, what we try to do here we simply try to close our eyes and we simply try to ask the program to repeat, right.

And for that we can define here inside the program that how the program has to be terminated and that is the job of the programmer that the programmer has to take a call that when the loop is going to be stopped. And for this actually we try to use here the commands break for example, you know we have discussed about the break and next command.

So, in case if you want to use the repeat loop the command in the R software is like this you try to repeat use here the word repeat r e p e a t all in lower case. And within this parenthesis try to write all the command that you want to execute that is all, very simple.

(Refer Slide Time: 22:37)



Example 3:

```
i = 1
repeat{ ①
print( i^2 )
i = i+2
if ( i > 10 )
break
} ①
[1] 1
[1] 9
[1] 25
[1] 49
[1] 81
```

```
> i = 1
> repeat{
+ print( i^2 )
+ i = i+2
+ if ( i > 10 )
+ break
+ }
[1] 1
[1] 9
[1] 25
[1] 49
[1] 81
>
```

$i=1$, print 1^2, $i=1+2=3$

$i=3$ $i>10 \rightarrow F \rightarrow$ print $3^2 \rightarrow i=3+2=5$

$i=5$ $i>10 \rightarrow F \rightarrow$ print $5^2 \rightarrow i=5+2=7$

$i=7, i>10 \rightarrow F \rightarrow$ print $7^2 \rightarrow i=7+2=9$

$i=9, i>10 \rightarrow F \rightarrow$ print $9^2 \rightarrow i=9+2=11$

$i=11, i>10 \rightarrow$ TRUE $\rightarrow$ Stop

So, let us try to take here one example and try to understand how it works and this will give you more clarity. Suppose I try to take here the same example that I consider in the while loop that I want to print the integers starting from 1 as long as their value is smaller than 10 and or less than equal to 10 on and then after that I want to break it, right.

So, I try to write down here i equal to 1 that is a initial value now you have understood it. Then I try to repeat and repeat whatever I want to repeat this I am trying to give it inside the curly bracket number 1. And then I am simply asking that print i square and then try to replace i by i plus 2 and then if i comes out to be greater than 10 then you break we stop that program.

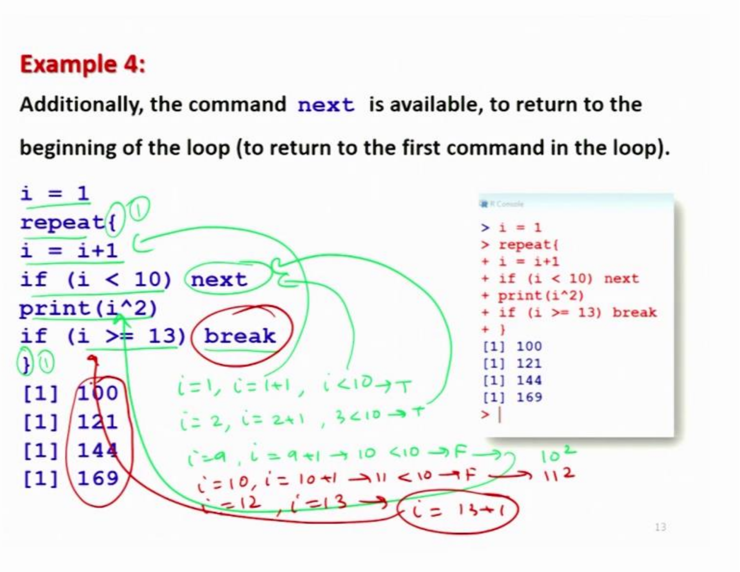
So, let us try to understand how it actually works. So, it will try to take here the first value here i equal to 1 and then it will here print 1 square. And then I will become 1 plus 2 equal to 3 and it will try to check whether i is greater than 10 because now i equal to here 3. Now the answer of this condition that i equal to 3 and whether i is greater than 10 this is here FALSE. So, it will once again execute the command print 3 square and then i will become here 3 plus 2 equal to 5. And now the new value of i is going to be here i equal to 5.

Now it will try to check whether 5 is greater than 10 or not answer is FALSE. So, it will try to print here 5 square and then I will become here 5 plus 2 equal to 7. Now, after this it will try to check i equal to 7 is greater than 10 or not the condition comes out to be here FALSE and it will try to print here 7 square. And then I will become here 7 plus 2 equal to 9.

Then after this it will try to see once again whether i equal to 9 whether i is greater than 10 or not the answer comes out to be here FALSE and it will try to print here 9 square and then I will be replaced by i plus 2 that is 9 plus 2 equal to 11. Now, if you try to see what happens now i is equal to here 11 and then it tries to check whether i is greater than 10.

Answer comes out to be here TRUE and so the program will stop here because you have given that you try to repeat the program as long as i is smaller than or less than equal to 10 and s i becomes more than 10 try to break it. So, the same outcome is shown here. So, you can see here that is not a very difficult thing to understand after you have understood the for and while loop.

(Refer Slide Time: 25:47)



Example 4:

Additionally, the command `next` is available, to return to the beginning of the loop (to return to the first command in the loop).

```
i = 1
repeat{
i = i+1
if (i < 10) next
print(i^2)
if (i >= 13) break
}
[1] 100
[1] 121
[1] 144
[1] 169
```

```
> i = 1
> repeat{
+ i = i+1
+ if (i < 10) next
+ print(i^2)
+ if (i >= 13) break
+ }
[1] 100
[1] 121
[1] 144
[1] 169
> |
```

And similarly I can also show you here in the similar program that how you can use this next and break both together right. So, I try to choose here a initial value i equal to 1 and I try to use here the command repeat and whatever I want to repeat I am trying to give it under the curly bracket 1. Then i will become here i plus 1 now I have a condition that if i is less than 10 then use the command next and print i square and if i is greater than or equal to 13 then break the program.

So, what will happen here try to understand the functioning that it will try to choose here i equal to 1 and then i becomes here i equal to i plus 1 that is 1 plus 1 2. And then it tries to this is happening here and then it tries to check here whether i is less than 10 this is here. So, i equal to here 2 2 is less than 10, so it is TRUE. So, you can see here i less than 10 is TRUE.

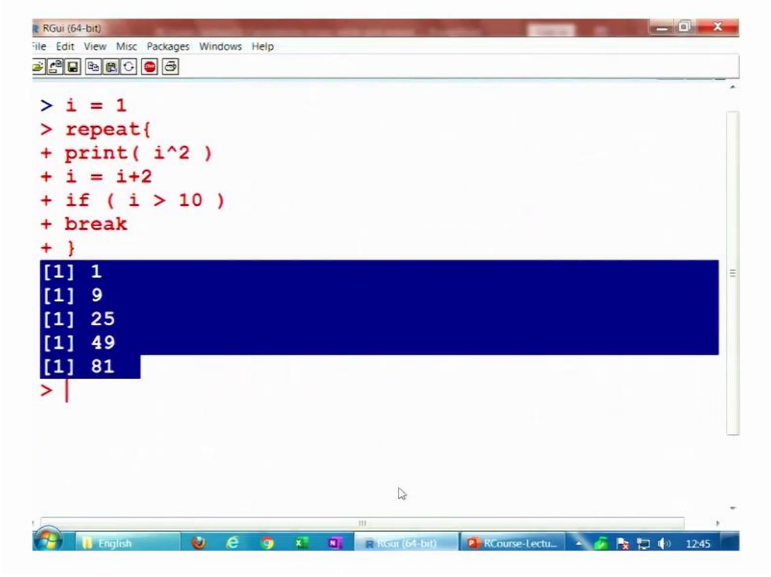
So, now what will happen after this? It will come to here next. And then it will not do anything then i will become here now here i equal to here 2. And then once again now the new i will become here 2 plus 1 equal to 1 which is equal to 3 and 3 is less than 10 yes this is 2. So, once again it will say next and it will keep on doing up to here i equal to 9.

Now, then it will become here i equal to 9 plus 1 equal to here 10 and then this condition is actually checked whether 10 is less than 10 this becomes here FALSE. And then whatever is the condition given here that print i square that will be executed here and 10 square will be printed. And then after this it will try to consider here the next value of i from 10 to it will become here 10 plus 1 11 and 11 is less than 10 this is FALSE.

So, it will try to print here 11 square and then it will continue to here i equal to 12 and i equal to 13 it will continue and then as soon as it become i equal to 13 then I will become here 13 plus 1. And then this condition is going to be here checked i is greater than or equal to 13 yes, and the function will break here and you will get here essentially the outcome as 10 square 11 square, 12 square and 13 square.

So, this is how you can see the things will work in the this repeat loop also. So, let me try to show you these examples on the R console also, but now you know that these are very very simple thing to execute. So, I try to copy this command and try to give it on the R console.

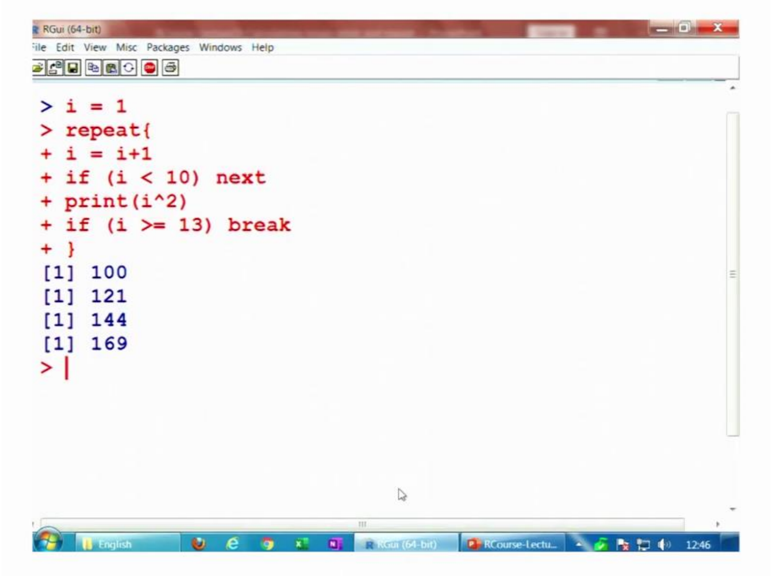
(Refer Slide Time: 28:54)



```
> i = 1
> repeat{
+ print( i^2 )
+ i = i+2
+ if ( i > 10 )
+ break
+ }
[1] 1
[1] 9
[1] 25
[1] 49
[1] 81
> 
```

You can see here this is here like this and you get here the same outcome which I shown you and explain you here, right, you can see this is the same outcome. And similarly if you try to use here the command here break and next together in this program in this example 4.

(Refer Slide Time: 29:13)



```
> i = 1
> repeat{
+ i = i+1
+ if (i < 10) next
+ print(i^2)
+ if (i >= 13) break
+ }
[1] 100
[1] 121
[1] 144
[1] 169
> 
```

So, you can see here if I try to place this commands here, it will give you the same outcome which is here like this, right.

So now, we come to an end to this lecture and you can see here that I have explained you here the concepts of repeat and while loops. And you can see that these are not very difficult things to understand. The main thing is that you have to understand what is happening inside the loop when the loop functions, when the loop works. Because based on that you have to take a decision in your programming that which of the loop are you going to really use for loop, while loop or repeat loop.

So, now why do not you take some examples try to think about some conditions under which you would like to use this while repeat and for loop. And try to write a small program one line, two line program and then try to execute it and by the time you do it I will also cover the topic of functions. So, the after that you will be in the conditions to write smaller programs also. So, you try to practice it and I will see you in the next lecture where, I will try to give you a brief explanation about the functions.

So, till then goodbye.