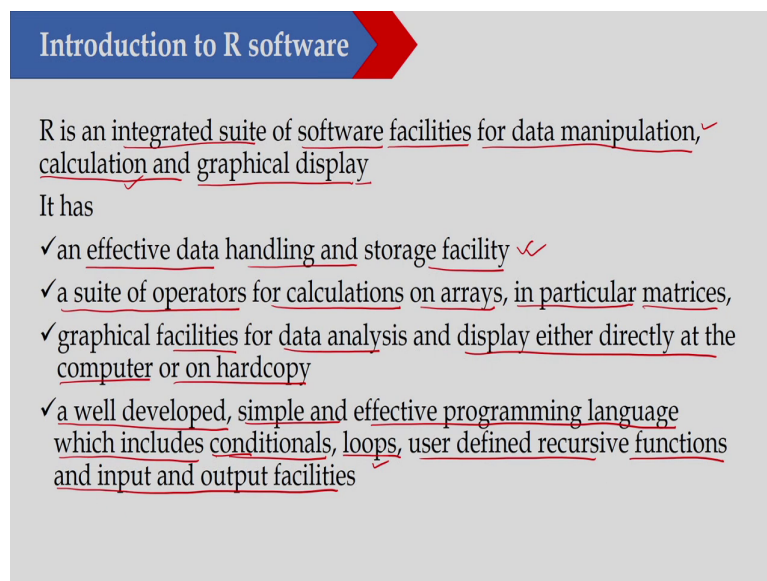


Optimization Methods for Civil Engineering
Prof. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 22
Introduction to R Software

Hello student, welcome back to the course on Optimization Methods for Civil Engineering. So, today I will introduce you R software ok. So, in this class, so, we will try to learn the basics of R, how what is R, how you can download R and then how to write simple lines how to do simple mathematical calculation so that I will discuss in this class.

(Refer Slide Time: 00:57)



Introduction to R software

R is an integrated suite of software facilities for data manipulation, calculation and graphical display ✓

It has

- ✓ an effective data handling and storage facility ✓
- ✓ a suite of operators for calculations on arrays, in particular matrices,
- ✓ graphical facilities for data analysis and display either directly at the computer or on hardcopy
- ✓ a well developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities ✓

Now, what is R? R is an integrated suite of software facilitates for data manipulation, calculation and graphical display. So, it is an integrated suite and where you can do the data manipulation, you can do the calculation and you can also do the graphical display of your

results. So, it has an effective data handling and storage facility. So, in R you have effective data handling and storage facility, a suite of operators for calculation on arrays in particular matrices.

So, it has the graphical facilities as I have said graphical facilities for data analysis and display either directly at your computer. So, you can display on your computer screen or on a hard copy. So, you can also take a print out of that one. It has a well developed simple and effective programming language which includes conditional, loops user defined recursive functions and input and output facilities.

So, it has the facilities so where you can do the programming you can write the logical statement you can write the conditional statement you can use loops and you have also user defined recursive function and has the input and output facilities.

(Refer Slide Time: 02:44)

Introduction to R software Developers of R software

- ✓ R is a free software ✓
- ✓ Developed by the R Development Core Team
- ✓ Available at www.r-project.org ✓
- ✓ It supports many free packages which helps the data scientist and analyst

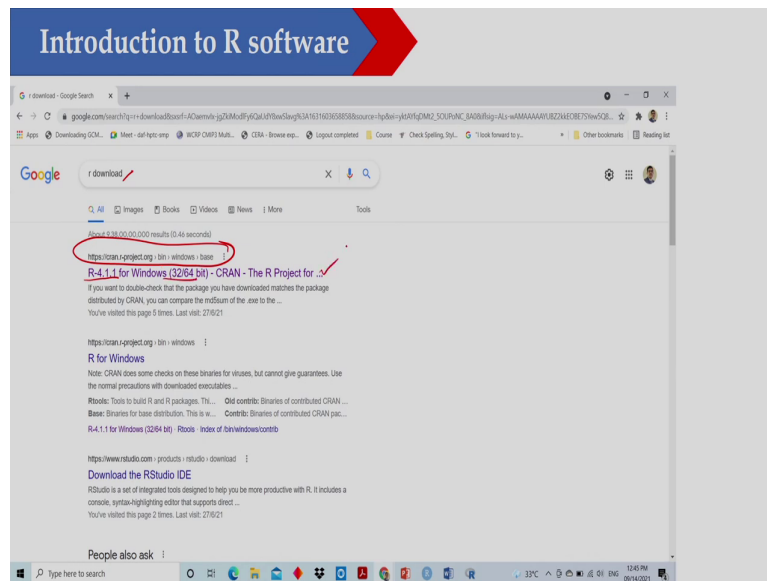
You may install R in a windows by downloading from the Comprehensive R Archive Network (CRAN) website: ✓
www.r-project.org or directly from http://cran.r-project.org ✓
https://cran.r-project.org/bin/windows/base/ ✓

Now, this is a free software. So, this is the main advantage of using R. So, it is a free software. So, you need not purchase it, you can simply download from the net and you can use it. So, today I will show you how you can download this one. Then developed by R development core team and available in this particular website, I will show you this website how you can download that one, it supports many free packages. So, which help us the data scientist and data analyst?

So, I can also install the free packages. So, I will also show you there are some packages. So, that you can install and you can use for your work ok. So, you may install R in a Windows by downloading from the Comprehensive R Archive Network CRAN website. So, you can download from this particular website. So, this is the URL of this particular website. So, you can go up go to this and you can download it. So, I have shown I have given you the URL. So, you can visit this URL and you can download the software.

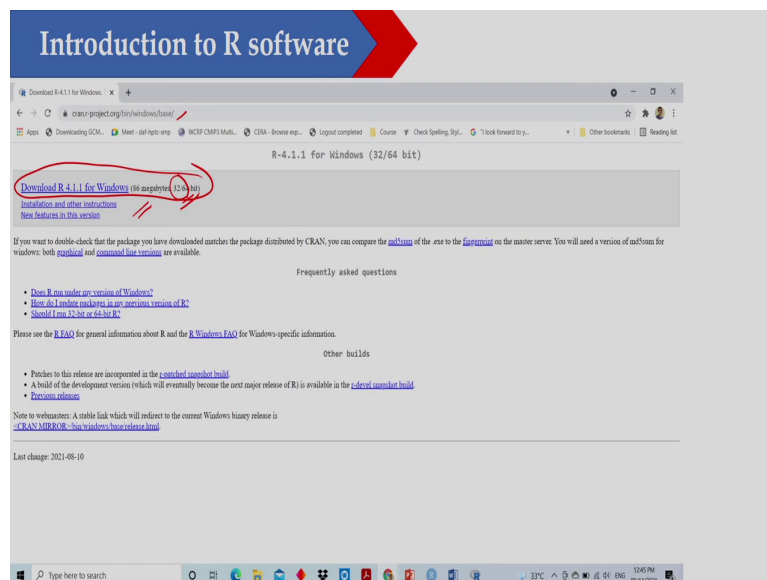
So, today I will show you how you can download this on a Windows platform, but if you are using Mac or if you are using UNIX then also you can install it. So, the free version of R can be installed on a UNIX system as well as on your Macbook.

(Refer Slide Time: 04:22)



So, what we can do basically directly you just write R download. So, in that case you go to this particular, Google will show you this one. So, we have that R 4.1.1. So, this is the Windows version. So, this is the website.

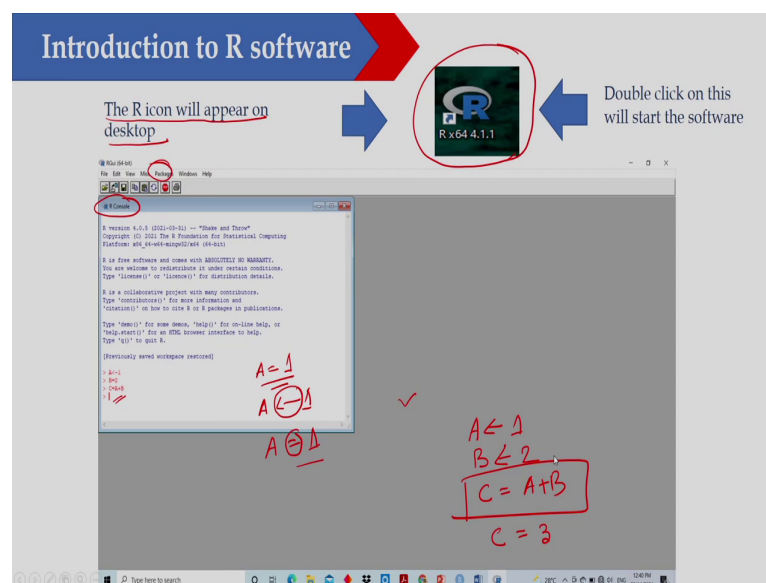
(Refer Slide Time: 04:41)



So, if you are clicking here, so, you will be getting this particular website. So, here from where you can download the software. So, I am using this one. So, I am downloading that 4.1.1 for Windows and it is for 32 and 64 bit. So, I will be using 64 bit version yeah. So, I can download.

So, you click here and you can download the executable file and installation is not that difficult. So, you just follow the instruction and you can install the software. So, I am not showing here, but this is not that difficult. So, you can do that.

(Refer Slide Time: 05:24)

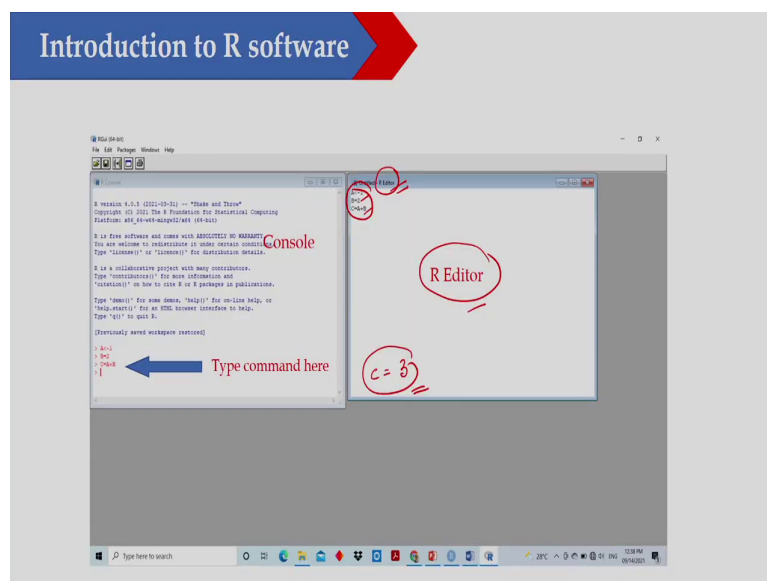


Now, once you are installing the software, so, this R icon, so, this is the R icon. So, R icon will appear on your desktop ok. So, it will be there and you just double click on this particular icon then R will be open. So, this is the R Window and you can see that this is the R console. So, R console here you are getting and then you have the other option, you can open the file, you can edit, you can view and you can see also packages. So, there are different options. So, I am not discussing here, but this is the R console you are getting and here you can actually do your programming.

So, you can execute some of the; suppose here in the first line I am assigning that A is equal to 1. So, I am assigning this value. So, how I am writing that A you can write it; you can write it the A equal to 1 or you can write A equal to 1 also. So, either you can use this symbol or you can also use equality symbol for assigning a value to a variable. So, you can do that.

Similarly, here what I have done that A equal to 1 then B equal to 2 and then I am calculating C equal to A plus B. So, this is very simple. So, it will give the value of C and C will be equal to 3 ok because A equal to 1, B equal to 2 and C equal to 3. So, I can do the calculation here itself. So, even I can also write a complicated statement here and get the result.

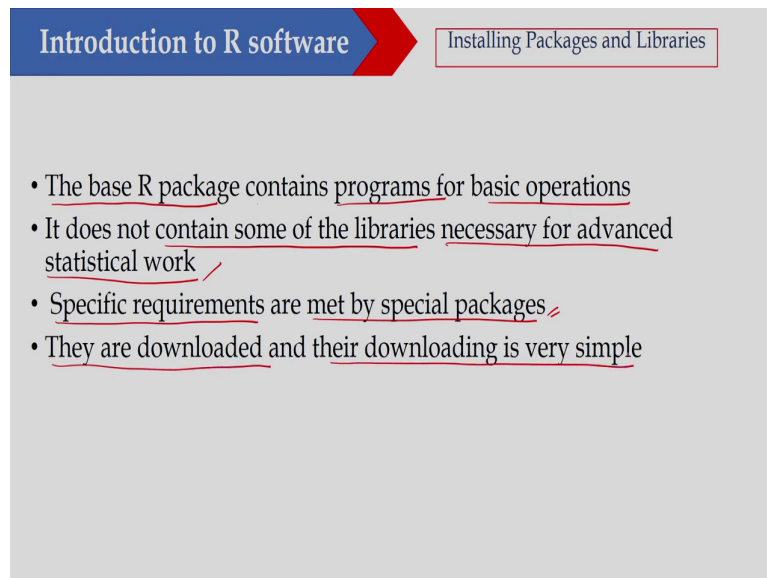
(Refer Slide Time: 07:18)



But what I can do basically? I can also use the R editor. So, how to open that editor? You go to file and open the R editor. Here I have open the R editor and the same calculation same calculation that A equal to 1, B equal to 2 and C equal to A plus B I have written here and once I am executing these 3 lines, so, I will get the value of C and which is equal to 3. So, I am getting the value of C which is equal to 3 in this case.

So, I can write a complicated program here also in using R editor, but R editor is not that efficient ok. So, R editor is not that efficient. So, therefore, I can use some other software. So, that also I will discuss like R Studio I can use. So, that I will discuss here.

(Refer Slide Time: 08:14)



Introduction to R software

Installing Packages and Libraries

- The base R package contains programs for basic operations
- It does not contain some of the libraries necessary for advanced statistical work ✓
- Specific requirements are met by special packages ✓
- They are downloaded and their downloading is very simple

Now, the R base package. So, whatever we have installed, so, this is the base package contains programs of basic operation. So, it contains the program of basic operation. So, it does not contain some of the libraries necessary for advanced statistical work. So, when you are installing R base package, so, it will not contain the library some of the libraries, so, which libraries are required for advanced computing statistical work.

So, we can download that in the end of this particular your lecture. I will also show you how you can download the libraries ok develop for different purposes. So, you can download that

libraries and you can install that one and then basically you can use that one. So that I will show you, but whatever basic R package so, that only contains program for basic operation.

Just like you can write a simple code using different conditional loop, you can assign, you can write the equation, you can write the function, but you will not be able to do some other advanced work. So, that for that you can download the libraries. Specific requirements are met by specific packages. So, as I said that they are there are different packages. So, I will also show you suppose GA is one package. So, where genetic algorithm is there. So, in this class I will show you how you can download GA package ok. So, that is for specific purpose.

So, specific requirements are made by specific packages. They are downloaded and they are downloading is very simple. So, easily you can download that software ok. So, or that package ok. So, you can download that one.

(Refer Slide Time: 10:12)

Introduction to R software Cleaning up the Windows

- We assign names to variables when analyzing any data
- It is good practice to remove the variable names given to any data frame at the end each session in R
- rm() command removes variable names
- For example, `rm(A, B, C)` removes the variables A, B and C

Handwritten notes:

$A \leftarrow 1$
 $B \leftarrow 2$
 $C = A + B$
 $rm(A)$
 $rm(A, B, C)$

$A = 1$
 $B \leftarrow 2$
 $C = A + B$
 $\downarrow \quad \downarrow$
 $1 \quad 2 = 3$

We assign names to variable when analyzing any data, ok. So, I will show you. So, suppose in the I have shown you that I would like to put that A equal to 1, so; that means, A is a variable and I am assigning a value equal to 1, B is basically 2. So, I am assigning a value 2 to the variable B and similarly I am calculating C equal to A plus B. So, here what I am doing?.

I am assigning a value 1 to A, similarly I am assigning a value 2 to the variable B then I am executing this particular line that is C equal to A plus B. So, therefore, the value of C will be equal to 3 in this case because A equal to 1 and B equal to 2. So, you should get 3 ok, it is good practice. So, when you are assigning value to a variable, so, it is a good practice to remove the variable names given to any data frame at the end of each session ok. So, end of each session you should remove that one. So, you can remove that one by using rm command this command will remove the variable. So, what I can do?.

Suppose I have assign A equal to 1 ok, B equal to ok 2, then C equal to A plus B ok. So, I will get the value C equal to 3. Now, I would like to remove the variable A. So, what I will do? I will write rm A. So, it will remove the variable A. Similarly, if I write rm A comma B comma C, so, it will remove all these 3 variables.

(Refer Slide Time: 12:16)

Introduction to R software

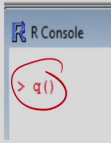
How to clear the screen in R

Press ctrl + L to clear the screen of R console.

 and 

- How to quit in R

Type `q()` to quit R.

Now, how to clear the screen in R? So, you can put control L to clean the screen. So, that I can do control L and similarly I can quit R by writing q ok. So, by writing q. So, by writing q I can quit R. So, you can close the R program by writing q.

(Refer Slide Time: 12:43)

Introduction to R software

Working with R Studio

- ✓ Some free software like R Studio, Tinn-R, etc. are also available to work with R software
- ✓ They are the interface between R and us and help in running the R software
- ✓ This software make coding and execution of programmes easier
- ✓ R Studio is available at <https://www.rstudio.com/>
- ✓ Rstudio is written in C++ programming language
- ✓ Rstudio is a free and open-source integrated development environment (IDE) for R

Now, as I said the basic R whatever we have installed, so, it has an editor. So, where you can write your program, so, you can write the line and all the lines you can execute, but it is not that efficient. So, what we can do? We can also download R Studio. So, this is also a free software, you can directly download that one from the net.

So, there are some free software like R Studio as I said. So, I will download R Studio, but there is another one Tinn-R. So, that also you can download etcetera are available to work with R software ok. So, this is a platform actually. So, this is actually interacting between R and basically us.

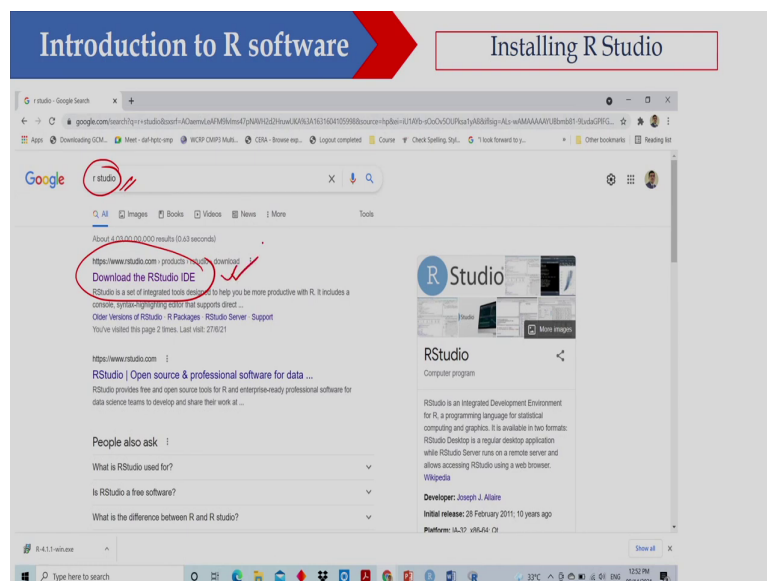
So, this is an interface and this interface can also be used and that is little bit more efficient than the R editor. So, they are interface between R and us and help in running the R software.

So, it is basically an interface and it will help you in running the R software. So, this software makes coding and execution of the programming easier.

So, you can as I said you can use the R editor, but you can also use R Studio and that is actually is really it is efficient in making the coding in executing and displaying the results. So, even graphical display also is better than R software ok. So, R Studio is available at this particular link.

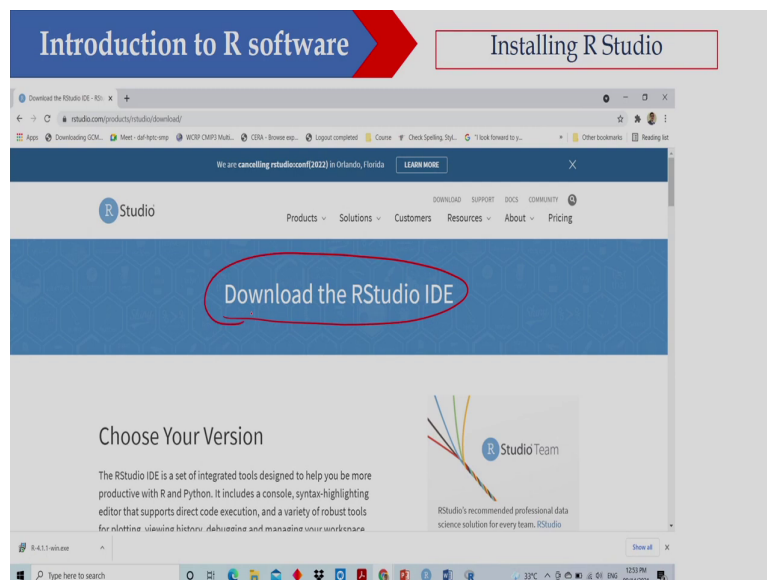
So, you can go to that one that is www.rstudio.com. So, from here you can download that one. So, RStudio is written in C plus plus programming language. RStudio is a free software as I said and open source integrated development environment for R ok. So, this is a free software and you can download it and you can use it.

(Refer Slide Time: 14:49)



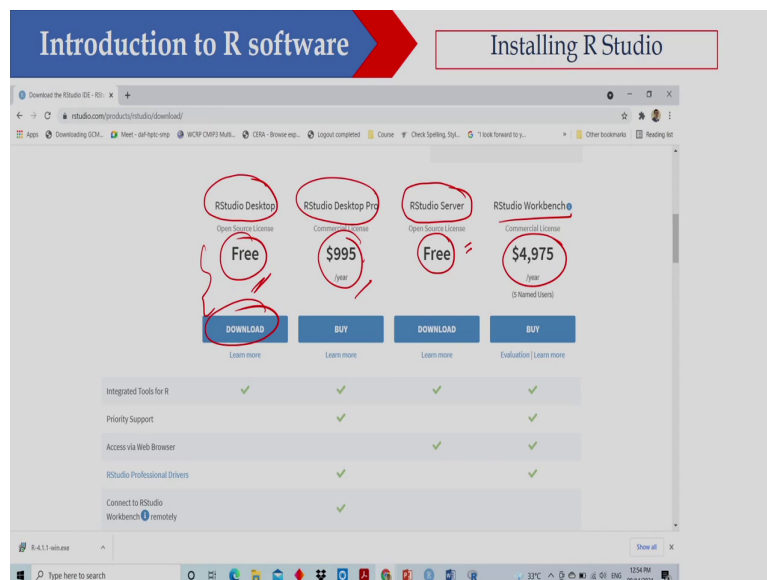
So, what you have to do? You again search in Google that is R Studio. Now, I am getting this link, that is download the RStudio ok. So, you please click in this particular link.

(Refer Slide Time: 15:06)



So, once you are clicking then you will get this particular page where download the RStudio, ok. So, download the RStudio.

(Refer Slide Time: 15:14)



So, if you scroll down then you will get this one. So, there are different version. So, this is RStudio Desktop and this is Desktop Pro then you can also download the Server one if you want to install in server and there is another RStudio Workbench ok. So, now this is free, this is free and this is also free, but if you want to purchase the Pro, so, it has some other facilities.

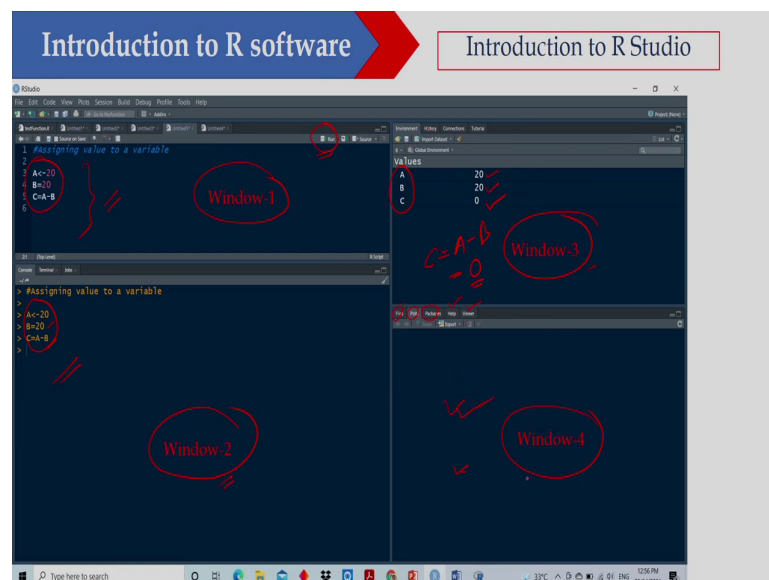
So, I do not want to use this one, but if you are suppose if you need this one then you can also you have to purchase this one, but for our work the free version is sufficient enough. So, I am not downloading the RStudio Server that is also free, but I will download the RStudio Desktop version ok. So, this is you can download.

So, you click on the download button and you can download that one and then you can install this, installation is also not difficult you just click on the exe file and then follow the

instruction given ok. So, you can install it very easily. So, here you should note one thing that so, we are working on R Studio ok. So, what you will do?.

You have to what you have to install R engine first ok the R engine and then you have to install R Studio, but when you are working. So, you will work on R Studio, but R engine should be there on your computer. So, therefore, first you have to install the R engine. So, as I have shown you and then you install the R Studio and then we will work on R Studio.

(Refer Slide Time: 17:03)



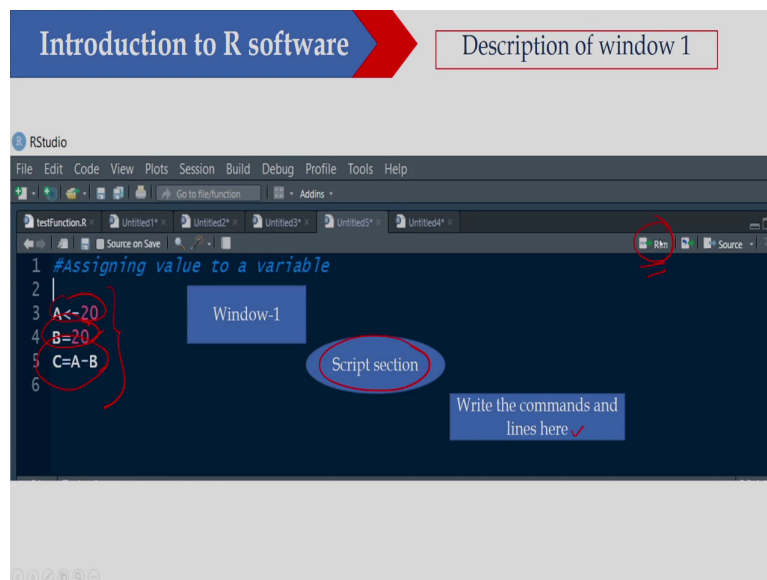
Now, once you are opening R Studio, so, you are getting this particular interface ok. So, we have here the Window 1, then Window 2, Window 3 and Window 4. So, we have total here 4 windows ok. Now, in Window 1 this is basically the editor. So, what I can do? I can write my coding or I can do the coding in this particular window, ok.

So, this is the editor and here I can do the or I can write this, but I can run this particular line using the Run button ok. So, using the Run button, so, you select that one and just by clicking at Run, so, these lines will be executed ok. So, once you are executed, so, this is the Console. So, here you will see that this lines has been executed ok. So, that A equal to 20, B equal to 20, A minus B. So, that has been executed and you are getting at Window 2.

And Window 3, so, you can under environment you can see all the variables here. So, in this case I have 3 variables here that is A, B, C. So, you can see that A equal to 20 I have assigned 20, B equal to 20 I have assigned 20 and C equal to; C equal to A minus B. So, therefore, C value is equal to 0. So, C is 0. Similarly, I can also see the history. So, if you click on History button, so, you can see the history the connection then tutorial. So, that also you can do it.

Now, if you look at the last one, so, last one you have actually, so, you can see the files, you can see the plots, you can see the packages. So, you can download the packages using this particular window and Help is there and Viewer is there ok, so, that you can see. Suppose if I plot something if I plot a curve, so, you can you will get your plot in this particular window.

(Refer Slide Time: 19:11)



So, here this is the Window 1. So, as I said in Window 1, this is the script section. So, I can write my script here. So, write the commands and lines here. So, suppose I have written A equal to 20. So, it will not execute, so, I then B equal to 20 and C equal to A minus B. Now, if I want to execute this particular line, so, what I will do?.

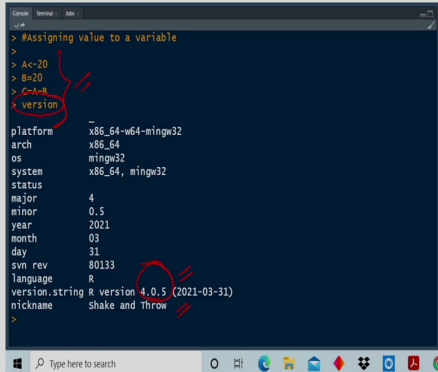
I have to select this first this line and then I have to use the Run ok. Similarly, then if I want to execute this particular line then again I will select that one and I will click Run and similarly if I execute the third line then I will select that one and I can execute that particular line using the Run button ok. Or if I want if I select all of them then I can and then I use Run button then it will all these three lines will be executed ok. So, this is Windows 1.

(Refer Slide Time: 20:15)

Introduction to R software

Description of window 2

Calculations take place in console window. One can write programs in console also but it is hard to make corrections and experiments with the coding.



The screenshot shows an R console window with the following text:

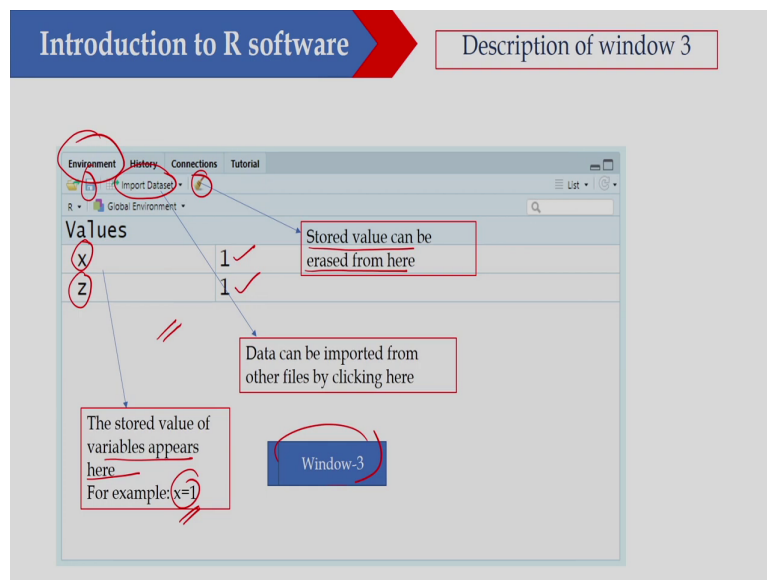
```
> #Assigning value to a variable
>
> A<-20
> B=20
> C=100
> version
platform      x86_64-w64-mingw32
arch          x86_64
os            mingw32
system       x86_64, mingw32
status
major         4
minor        0.5
year         2021
month         03
day           31
svn rev       80133
language      R
version.string R version 4.0.5 (2021-03-31)
nickname      Shake and Throw
>
```

Red annotations in the original image include a bracket on the right side of the first four lines of code and a circle around the 'version' command.

Now, Windows 2 as I said, so, whatever you are executing, so, this will be there in console ok. Then whatever you are executing the result you can see in this particular window. Suppose I would like to see what version we have actually. So, I am writing the version and it is showing the version ok.

So, this is the version I have basically. So, you can see. So, whatever R version is there that is 4.0.5 and this is the version I have your downloaded ok or this is the version I have installed here. So, whatever you are executing. So, you can you will get your result in this particular window.

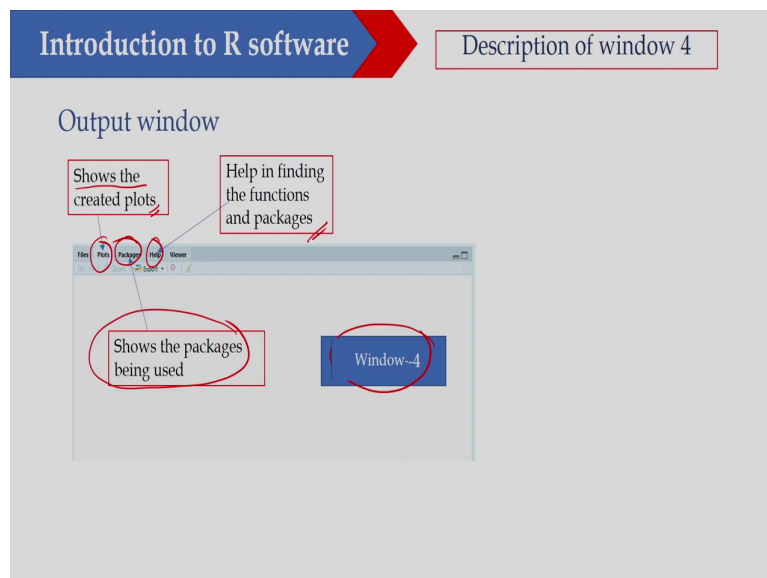
(Refer Slide Time: 20:59)



Then in the Window 3, so, this is under environment. So, under environment you can see the variables. Suppose you have x and z variable, x equal to 1 and z equal to 1, so, that you can see. The store value of variables appeared here for example, x equal to 1. So, whatever variable is there you can see in the on the environment window. And similarly, you can import database using that one. Suppose, I would like to import some data to this particular environment, so, I can import that one.

And similarly if I want to erase all these data if I want to delete all these data, so, whatever store value can be erased from here, so, by clicking at this particular button, so, I can actually delete all these variables. So, I can clean the environment ok using this one. Similarly, I can also save the environment ok that also I can do and there are some other option. So, I am not discussing here, but you can also explore those option.

(Refer Slide Time: 22:09)



Now, the output window that is the Window 4, ok. So, as I said in the output window, so, it shows the created plot. Suppose if you have created some plots. So, you can see here similarly if you want to see what are the packages you are using, so, you can go to Package and then Help button will give you if you want if you need any help.

So, you can go to Help button and you can search for help. So, that also you can do. And you can install package using this particular your window also. So, I would like to install suppose a particular software that also you can install using this particular window.

(Refer Slide Time: 22:55)

Introduction to R software

Basic calculation with R (Using R Studio)

Assignment operators are the left arrow dash (<-) or equal sign (=)

The screenshot shows the R Studio interface. The console on the left contains the following code:

```
# Assigning value to a variable  
A <- 20  
B <- 20  
C <- A - B  
#
```

Handwritten red annotations include: a bracket around the first three lines, the formula $C = A - B$, and circled assignments $A <- 20$ and $B <- 20$. The Environment pane on the right shows the values of the variables:

Variable	Value
A	20
B	20
C	0

Handwritten red annotations next to the screenshot include the left arrow dash (<-) and the equals sign (=).

The basic calculation with R so, using R Studio. As I said I will work mainly on R Studio, so, but I have downloaded the R engine and then R Studio, but I am working on R Studio. I am not opening the R engine, but I am opening the R Studio. Now, assignment of operator are left arrow and dash.

So, I can assign a value to a variable using this particular left arrow and dash. So, this is left arrow and dash or I can also use the equality sign both are acceptable here. So, you can write suppose here I have written that A is equal to 20 or I can write B equal to 20, ok. So, both will assign; that means, A is a variable and the value of that variable is 20, B is a variable and value of B is 20 and C is a line where I am doing some operation that C equal to A minus B.

So, you will be getting the value equal to 0, ok. So, once I am executing this, so, I am getting this one. Now, if you are writing a hash in front of any line suppose here I am writing hash,

so, this is giving you a comment that particular line will not be executed. So, therefore, this has not been executed ok this line, but it is showing here and this is a comment you are getting.

(Refer Slide Time: 24:34)

Introduction to R software

Basic calculation

- ✓ The character `#` marks the beginning of a comment
- ✓ All characters until the end of the line are ignored

```
# Assigning value to a variable
A<-20
B<-30
# C<-A+B
C<-A+B
C
```

Output:

```
> A
[1] 20
> B
[1] 30
> C
[1] 50
```

Environment:

Variable	Value
A	20
B	30
C	50

So, as I said the character has marks the beginning of a comment; that means, if I am if I put any hash, so, this line will not be executed and it is basically you are writing some comment. All characters until the end of the line will be ignored. So, all will be ignored. Suppose I can also write a hash here and I can write a comment here ok.

So, in that case this portion will be executed, but this portion will not be executed because I am putting this hash here ok. So, now, in this case, so, what I am doing? I am assigning A equal to 20. So, you can see that A equal to 20 B equal to 30. So, B equal to 30 C equal to a plus B that means, I should get 50. So, I am getting C value is equal to 50. So, here what is

happening? If I write A and I execute this one. So, it will display the value of A in this particular window.

Similarly, if I write B here and then if I execute then I will get the value of B. So, I am getting the value of B is 30, value of A is 20 and value of C is equal to 50 and you can also see here. So, on the environment window you can see that A equal to 20, B equal to 30, C equal to 50. So, you can get that one on the environment window also.

(Refer Slide Time: 26:06)

The slide has a blue header bar on the left with the text 'Introduction to R software' and a red arrow pointing right. To the right of the arrow is a white box with a red border containing the text 'Calculations with Data Vectors'. The main content area is light gray and contains two lines of text: 'How R behaves with data vectors?' and 'What happens when a scalar is added/subtracted/Multiplied/divided in a data vector'. The words 'added', 'subtracted', 'Multiplied', and 'divided' are underlined, and red arrows point from the word 'scalar' to each of these four underlined words.

Introduction to R software

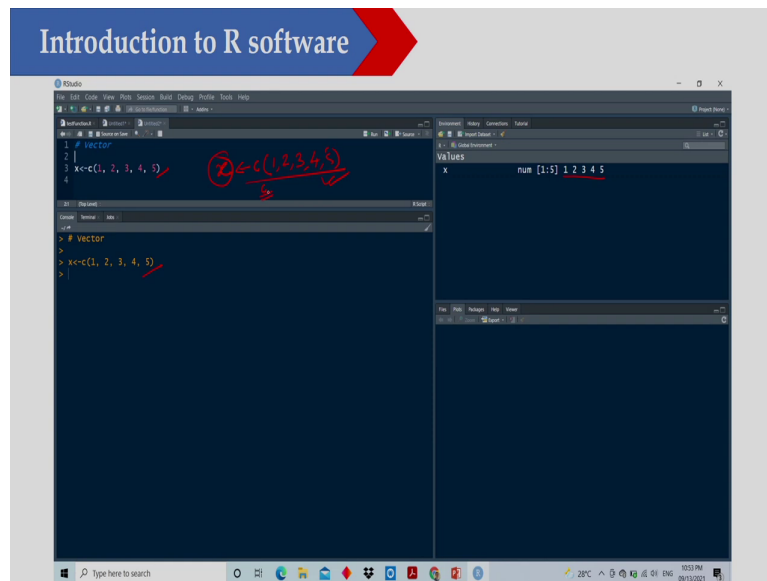
Calculations with Data Vectors

How R behaves with data vectors?

What happens when a scalar is added/subtracted/Multiplied/divided in a data vector

Now, let us see how R behaves with data vector ok. So, calculation with data vector. So, what happen? When a scalar is added or subtracted or multiplied or divided in a data vector ok. So, if I add a scalar to a data vector or if I subtract a scalar from a data vector if I multiply, if I divide it, so, what will happen? So, let us see that one.

(Refer Slide Time: 26:41)



Now, a vector is defined using this command, so; that means, how I will define that using C. So, C is suppose I am writing here C equal to 1, 2, 3, 4, 5. So, using this one and I am assigning that one to x. So, x is a vector actually here and the values are 1, 2, 3, 4, 5s ok. So, I am defining here. So, if I execute that one, so, I am getting this one and if I see what is x? x equal to so, it is 1 to 5; 1, 2, 3, 4, 5 ok. So, I am defining if I can define a vector using C ok. So, this is I am defining.

(Refer Slide Time: 27:30)

Introduction to R software

Addition in Data Vectors: $x + y$

$c(2,5,6,7)+3$

$2+3, 5+3, 6+3, 7+3$

x y

$c(2,5,6,7) + c(-2,-6,-7,1)$

$2+(-2), 5+(-6), (6-7), (7+1)$

```
R Console 1
> x=c(2, 5, 6, 7)+3
> x
[1] 5 8 9 10

R Console 2
> x=c(2, 5, 6, 7)+c(-2, -6, -7, 1)
> x
[1] 0 -1 -1 8
```

Now, let us see what will happen that C is a vector ok. So, this is a vector that is 2, 5, 6, 7; 2, 5, 6, 7 and I am adding another scalar that is 3 with this particular vector. So, what I will do basically? So, I am writing here this is C ok and C equal to 2, 5, 6, 7 and this is a vector and with that I am adding 3. So, what will happen?.

3 will be added to all these values; that means I will be getting 2 plus 3, 5 plus 3, 6 plus 3, 7 plus 3; that means, 3 will be added to this particular vector. So, all the elements of the vector. So, therefore, I will be getting 2 plus 3 that is 5; 5 plus 3 that is 8; 6 plus 3 that is 9 and 7 plus 3 that is 10 ok.

So, what I am doing here? So, I am adding this is a vector ok. Suppose this is x and this is another vector this is y. So, what I am doing? x plus y and you can see that how many elements are here? We have 4 elements and we have 4 elements here. So, in that case what

will happen? The corresponding element will be added; that means, first element of this x and first element of y will be added that is 2 plus minus 2.

Similarly, second element of x and the second element of y that is 5 plus 6, similarly third element of x and third element of y that is 6 minus 7 and the fourth element of x and fourth element of y will be added. So, you can see here. So, I have added 2 vector. This is the first vector, this is x and this is the second vector this is y. And what I am getting? That is 2 minus 2. So, I am getting 0, 5 plus 6 this is 11, 6 minus 7 minus 1 7 plus 1 is 8. So, I am getting this particular result. So, if I execute this particular line.

(Refer Slide Time: 29:48)

Introduction to R software

$c(1,2,3,4) + c(6,7)$

(4) (2)

1+6, 2+7, 3+6, 4+7 ✓

$c(1,2,3,4,5) + c(6,7)$

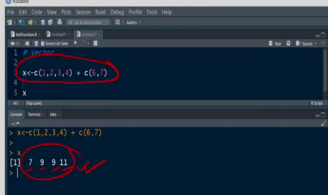
(5) (2)

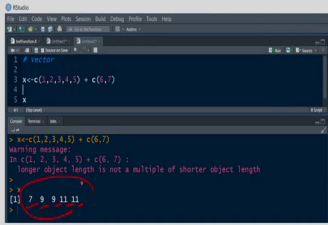
1+6, 2+7, 3+6, 4+7, 5+6 ✓

Warning message: ✓

In $c(1, 2, 3, 4, 5) + c(6, 7)$: longer object length is not a multiple of shorter object length ✓

Addition in Data Vectors: x + y





Now, if I add these 2 vector, so, this is x and this is y. Now, we just see the difference here. So, we have total 4 elements and here we have 2 elements ok. So, therefore, what will happen? So, it will add the first element. So, what will happen? The 6 will be added to 1 and

7 will be added to 2 ok. So, 6 is added to the first element, 7 is added to the second element then it will repeat again with third element it will the 6 will be added and fourth element the 7 will be added.

So, here what is the; we have 4 here and we have 2 here. So, I can also show you here. So, this is I have done in R Studio and you can see the first element is 7, second element is 9, third element is 9 and fourth element is 11 ok, so; that means, 1 plus 6 that is 7 2 plus 7 that is 9 3 plus 6 that is 9 again 4 plus 7 that is 11. So, I am doing that one. Now, you just see here what is happening, we have total 5 and this is 2 ok.

So, in this case what will happen? The one will be 6 will be added to 1, 7 will be added to 2 then it will repeat. 3 the 6 will be added to 3 and 7 will be added to 4 and 5 and on the last element 6 will be added to 5, but then 7 is again. So, there is there should be another element. So, in that case it will execute that one, but it will give a warning ok.

So, what is the warning here? So, warning here is that longer object length is not a multiple of shorter object length. So, it is not a multiple. In this case the longer one is a multiple of shorter one ok. So, therefore, there is no error, but in this case the longer one is 5 and shorter one is 2. So, it is not a multiple.

So, therefore, there will be an error message ok. So, otherwise this will also calculate ok, you can see. So, I am getting 7, 9, 9, 11, 11. So, I am getting this particular results.

(Refer Slide Time: 32:16)

Introduction to R software

Multiplication in Data Vectors: $x * y$

RStudio interface showing the console output of the command `x = c(1,1,1,1,1,1)^2`. The output is `[1] 2 4 6 8 10`. Handwritten notes in red ink include $a \cdot b$, $x = a * b$, and $1 \times 2, 2 \times 2, 3 \times 2, 4 \times 2, 5 \times 2$.

$1 \times 2, 2 \times 2, 3 \times 2, 4 \times 2, 5 \times 2$

Now, if I multiply 2 vector ok, so, what I am doing here? I am multiplying a vector with scalar. So, here this is suppose a is a vector and b is a scalar and x equal to $a \star b$ ok. So, now what will happen? b is a scalar here. So, what will happen? All the elements will be multiplied by 2. So, therefore, what I am getting? This will be 1 into 2, then 2 into 2, 3 into 2, then 4 into 2 and then 5 into 2 ok.

So, how much you are getting? That 2 into 2. So, I am getting 2 then 2 into 2 I am getting 4 then 3 into 2 I am getting 6 then 4 into 2 I am getting 8 and 5 into 2 I am getting 10. So, all the elements will be multiplied by this particular scalar. So, results are also shown here and you can see that is 2, 4, 6, 8 and 10 I am getting this one. So, here I have multiplied a vector by a scalar.

(Refer Slide Time: 33:34)

Introduction to R software

Multiplication in Data Vectors: $x * y$

```
x <- c(1, 2, 3, 4, 5) * c(5, -4, 3, 2, -1)
y <- c(1, 2, 3, 4, 5) * c(5, -4, 3, 2, -1)
x * y
```

Handwritten red annotations: $x = a \times b$, 1×5 , $2 \times (-4)$, 3×3 , 4×2 , $5 \times (-1)$

Console output: $[1] 5 -8 9 8 -5$

Now, you just see here in this case I have multiplied 2 factor of equal length. So, here it is a here it is b. So, a is also a vector and b is also a vector ok. So, I am getting the value of x here. So, here if you are multiplying 2 vector ok, so, in that case what you will get? So, the corresponding element will be multiplied. So, here what you are getting?.

One is the first element of vector a and 5 is the first element of vector b; so, 1 into 5 2 into minus 4 3 into 3 4 into 2 and 5 into 1. So, what is the result here? That 1, then 8, 9, 8, 5 ok, so; that means, if you are using a star b then this is a multiplication of 2 vector. So, you are getting this particular results. So, I hope this is clear to you.

(Refer Slide Time: 34:43)

Introduction to R software

Multiplication in Data Vectors: $x * y$

```
# Create vector x
x <- c(1, 2, 3, 4) * c(1, -4)
x
# [1] 1 -8 12 -16

# Create vector y
y <- c(1, 2, 3, 4) * c(1, -4)
y
# [1] 1 -8 12 -16
```

The screenshot shows the RStudio interface. The console on the left displays the execution of R code. The environment pane on the right shows the variables x and y, both of type numeric, with their values displayed. The values for x and y are 1, -8, 12, and -16. A red circle highlights the values in the environment pane. Below the screenshot, the calculation is summarized as $1 \times 5, 2 \times (-4), 3 \times 5, 4 \times (-4)$.

$1 \times 5, 2 \times (-4), 3 \times 5, 4 \times (-4)$

Now, so, here what I am doing? Here a is a vector and b is also vector, but length of the vector are not equal. So, length is 4 here and this is 2 here, but this is the longer one and this is the shorter one, but longer one is a multiple of the shorter one. So, what I will get? So, in this case the 5 will be multiplied to 1 then minus 4 then this will repeat again this is 5 and this is minus 4 ok. So, results are 5 minus 8 15 minus 16 you are getting and results are also stored here. So, there is no issue ok.

(Refer Slide Time: 35:28)

Introduction to R software

Multiplication in Data Vector: $x*y$

$$1 \times 5, 2 \times (-4), 3 \times 5, 4 \times (-4), 5 \times 5$$

Warning message: In `c(1, 2, 3, 4, 5) * c(5, -4)` :
longer object length is not a multiple of
shorter
object length ✓✓

```
# vector
> x=c(1,2,3,4,5)*c(5,-4)
[1] 5 -8 15 -16 25
```

So, here what we are doing? So, here length of the vector is 5 and length of the vector is 2. So, in this case the longer object length is not multiple of the shorter object length. So, in this case the longer object length is not multiple of shorter object length. So, therefore, I will get the result. So, this is the result I am getting, but I am also getting a warning ok. So, this is the warning you will get.

So, here what you are getting? 1 into 5, then 2 into minus 4, then 3 into 5, 4 into minus 4, 5 into 5 and basically so, it is a as the length is not multiple of the shorter one. So, I am getting a warning, but I am getting; I am getting the results here that is 5, minus 8, 15, minus 16, plus 25.

(Refer Slide Time: 36:28)

Introduction to R software

Division in Data Vectors: x / y

```
x <- c(1,2,3,4,5)
y <- c(1,2,3,4,5)
x/y
```

1 1 1 1 1

$1 \div (1), 2 \div (2), 3 \div (3), 4 \div (4), 5 \div (5)$

So, here I am dividing x by y ; that means division in data vector. So, that is x is a vector and y is also a vector. So, x by y I am doing. So, this is suppose a and this is b . So, what I am calculating? x equal to a by b . So, if you are doing that one, so, in this case what will happen? The corresponding element will be divided ok. So that means, what will happen?

1 divide by 1, then 2 divided by 2, 3 divide by 3, 4 divide by 4, 5 divided by 5. So, this is the same vector that here it is a equal to b . So, therefore, the result is 1 1 1 1; 1 1 1 1 1 1. So, this is the result I am getting ok. So, this is the result I am getting and the corresponding element will be divided ok.

(Refer Slide Time: 37:24)

Introduction to R software

Power operator in Data Vectors

$x = c(1, 2, 3, 4, 5)$
 $y = c(1, 2, 3, 4, 5)$

$1^1, 2^2, 3^3, 4^4, 5^5$

x^y

$x = c(1, 2, 3, 4)$
 $y = c(2, 3)$

$1^2, 2^3, 3^2, 4^3$

x^y

R Console 1:

```
x = c(1, 2, 3, 4, 5)
y = c(1, 2, 3, 4, 5)
x^y
[1] 1 4 27 256 3125
```

R Console 2:

```
x = c(1, 2, 3, 4)
y = c(2, 3)
x^y
[1] 1 8 9 64
```

Now, if I want to calculate power, so, power operator in data vector. So, I have taken a vector. Suppose this is x and this is y ok. Now, what you will do basically? So, if you are using this power, so, it will be 1 to the power 1, 2 to the power 2, 3 to the power 3, 4 to the power 4, 5 to the power 5. So, what you are getting? You are getting 1, 4, 27, 256 and 3125 ok, so that means, 5 to the power 5 is 3125.

Now, if I am writing, so, here I am writing this is one vector and this is another vector. So, here the longer one is multiple of the shorter one. So, what I will get? 1 to the power 2, 2 to the power 3, 3 to the power 2 and 4 to the power 3. So, I am getting that one. So, this is the result I am getting here that is 1, 8 and this is 3 square 9 and 4 to the power 3 64.

(Refer Slide Time: 38:37)

Introduction to R software

Power operator in Data Vectors

$$c(1,2,3,4,5)^{c(2,3)}$$

$1^2, 2^3, 3^2, 4^3, 5^2$

Warning message:
In $c(1, 2, 3, 4, 5)^{c(2, 3)}$:
longer object length is not a multiple of shorter
object length

```
RStudio
File Edit View Help Session Tools
1 # vector
2 x=c(1,2,3,4,5)^c(2,3)
3 x
4 [1]
5 1
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647

```

(Refer Slide Time: 39:19)

Introduction to R software		Built-in Commands and Plots
<u>abs()</u>	Absolute value	$\text{abs}(x) = x \text{ if } x > 0$ $= -x \text{ if } x < 0$
<u>sqrt()</u>	Square root	$\text{sqrt}(4) = 2$ $\text{sqrt}(9) = 3$
<u>round()</u> , <u>floor()</u> , <u>ceiling()</u>	Rounding up and down	
<u>sum()</u> , <u>prod()</u>	Sum and product	$x = 1:10$ $\text{sum} \leftarrow \text{sum}(x)$
<u>log()</u> , <u>log10()</u> , <u>log2()</u>	Logarithms	
<u>exp()</u>	Exponential function	
<u>sin()</u> , <u>cos()</u> , <u>tan()</u> , <u>atan()</u> , <u>asin()</u> , <u>acos()</u>	Trigonometric function	$x = 1:10$ $\text{max}(x) \rightarrow 10$ $\text{min}(x) \rightarrow 1$
<u>max()</u> , <u>min()</u>	Getting maximum from the vector	

Now, in R, so, there are some built in command ok. So, that commands are absolute, suppose so, what you can do? You can calculate the absolute values. Suppose x is a variable and I would like to calculate the absolute value of x. So, what I will do? I will write ab as x ok. So, it will give you. What it will give? It will give x, if x is positive; it will also give x, if x is negative ok, it will give negative. So, it will get.

So, similarly I can calculate square root ok the square root I can calculate. Suppose if I write sqrt and I am writing 4. So, I will be getting 2 ok, sqrt this is 9. So, I will getting 3 ok. So, I can find out the square root of any variable using that one. Similarly, I can use round, floor, ceiling ok. So, rounding up and down basically. So, you can use that one then if you want to use sum and product ok.

So, suppose x equal to 1 to 10 ok and I can write that if I want to sum ok, sum is equal to that is sum, this is x . So, it will add all these at the vector ok. So, you can calculate, similarly you can also calculate the product that I that you can do product of two. Similarly, I can also calculate logarithmic.

So, log, then $\log_{10}$, $\log_2$. So, that you can do. Similarly, exponential function that you can write $\exp$ then trigonometric function that $\sin x$, $\cos x$, $\tan x$, atan , asin , acos . So, you can use that one and max and min function is also there to get the maximum from a vector. Suppose, if x is a vector which is 1 to 10 then if I write $\max x$. So, what I will get? I will get the value 10.

If I write $\min x$, so, what I will get? I will get 1. Is not it? So, its a vector x is a vector which will have 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. So, maximum is 10 and minimum is 1. So, I can use max and min function to calculate maximum value or minimum value of the vector ok. So, I can use this one.

(Refer Slide Time: 42:21)

Introduction to R software

Logical Operator

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

✓TRUE and FALSE are reserved words denoting logical constraints

Operator	Executions
>	Greater than ✓
>=	Greater than or equal to
<	Less than
<=	Less than or equal
==	Exactly equal to
!=	Not equal to

$$x = y$$
$$x == y$$
$$x != y$$

10 20

So, there are the logical operators. So, in programming I also need the logical operator the following table shows the operation and function for logical comparisons TRUE or FALSE. TRUE and FALSE are reserved word denoting logical constraints ok. So, what is this? This is greater than and this is greater and equal then this is less than this is less than equal. Then if you it is equal, suppose if I say x equal to y, so, what does it mean?.

That the value of y will be assigned to x, but if I say x equal to y, so, this is a logical statement; that means, variable x is equal to y if it is equal then it will return TRUE and if it is not equal it will return FALSE ok. And not equal. So, not equal; that means, if I write not equal to y. So, if it is not equal suppose x value is 10 and y value is 20. So, this is not equal. So, this operator will give you TRUE ok.

And if it is equal then it will give you FALSE ok. So, I can use this operator in my coding. So, I can use it if there is any logical operation greater than then greater than equal to this type less than less than equal then equal not equal to so that I can use this symbol.

(Refer Slide Time: 44:00)

Introduction to R software

Matrices

- A matrix is a collection of data elements arranged in a 2-dimensional grid (rows and columns).
- As with vectors, all the elements of a matrix must be of the same data type
- A matrix can be generated in several ways ✓
- The function matrix creates a matrix from a given set of values ✓
- A Matrix can be created using the matrix() function

Syntax: The basic syntax for creating a matrix in R is

```
matrix(data, nrow, ncol, byrow, dimnames)
```

Let us see how you can define matrices in R. A matrix is a collection of data element under 2-dimensional grid column and rows ok. As with vectors, all elements of a matrix must be of same data types ok. So, it must be of same data type then a matrix can be generated several ways. So, we have different ways to generate a matrix. So, I am explaining here how you can generate a matrix. The function matrix create a matrix from a given set of values.

So, you can use the matrix function you can use the matrix function to create a matrix. So, that I will show you and if from a given values. So, I can use this matrix function to create a

matrix from a given set of values. A matrix can be created using matrix function as I said. So, this function you can use.

The argument of this function is data. So, you have to; you have to put the data here and data will put as a vector the number of rows number of column then by rows actually. So, by default it will be by column I will explain you and dimnames.

(Refer Slide Time: 45:33)

Introduction to R software

Creates a matrix

By default, the function matrix fills in the matrix column by column. Set the argument byrow = TRUE to fill in the matrix row by row.

$A <- \text{matrix}(20:25, \text{nrow} = 3, \text{ncol} = 2)$

```
A <- matrix(20:25, nrow = 3, ncol = 2)
```

Environment: int [1:3, 1:2] 20 21 22 23 24 25

Values:

	int [1:3, 1:2]
A	20 21 22 23 24 25

So, let us create a matrix here. So, as I said I can use this matrix function. So, here I am putting the data is a vector that is C. So, here I am putting the data suppose this is a equal to 20 to 25; that means, it will have 21, 22, 23, 24 and 25 ok. So, it will have these data and number of rows equal to 3, number of column equal to 2 ok.

So, if I execute, so, I have written this particular line here and whatever matrix I have created. So, I am assigning this matrix to the variable A. So, now, variable A is a matrix and its dimension is 3 by 2; that means, number of rows equal to 3 then number of columns equal to 2 and this is the data.

So, you can see this is the matrix A here and by default it is basically by column. So, you can see that 20, 21, 22, 23, 24, 25. So, what will happen? First it will go to the 1st column and then it will go to the 2nd column; that means, the first element will be 1st row 1st column, second element will be 2nd row 1st column then third one will be 3rd row 1st column and then it will fill the 2nd column.

So, what I have done here? I have not written byrow ok, so, byrow. So, by default byrow is FALSE ok. So, by default byrow is FALSE. So, if you are not writing byrow if you are not writing byrow; that means, it will consider the byrow is equal to FALSE ok. So, I am getting this matrix and matrix is also stored here.

(Refer Slide Time: 47:38)

Introduction to R software

Creates a matrix

```
A<-matrix(20:25, nrow = 3, ncol = 2, byrow = TRUE)
```

The screenshot shows the RStudio interface. The console displays the command `A<-matrix(20:25, nrow = 3, ncol = 2, byrow = TRUE)` and the resulting matrix `A`. The matrix is printed as:

```
      [,1] [,2]
[1,] 20 21
[2,] 22 23
[3,] 24 25
```

Handwritten red annotations are present:

- A red circle highlights `byrow = TRUE` in the command.
- Two red boxes with arrows point to the matrix output. The top box contains `20 21` and `22 23`, with an arrow pointing to it labeled `byrow = F`. The bottom box contains `20 21` and `22 23`, with an arrow pointing to it labeled `byrow = T`.

Now, is the C the I am writing byrow through basically. So, I can write I just I have just added this one just see the difference. So, here what is happening? byrow TRUE that means, it will fill byrow; that means, 20, 21, 22, 23, 24, 25. So, just look at. So, earlier one it was 20, 21, 22, 23, 24, 25. So, this was the matrix ok.

Now, this one is 20, 21, 22, 23, 24, 25, ok. So, this is; so, here what I have done? The byrow equal to TRUE ok. So, here I am not defining anything. So, if you are not defining then in that case byrow is equal to FALSE ok. So, byrow FALSE. So, it will fill the column y that 1st column then 2nd column then 3rd column something like that I hope this is clear. So, I can create a matrix.

(Refer Slide Time: 48:48)

Introduction to R software

Creates a matrix

- ✓ Other functions to define matrices are cbind and rbind
- ✓ The function cbind creates a matrix by binding two or more vectors as column vectors
- ✓ The function rbind creates a matrix by binding two or more vectors as row vectors

The screenshot shows the R Studio interface. In the console, the following code is entered and executed:

```
a <- 1:4  
b <- 5:8  
A <- cbind(a,b)
```

Handwritten red annotations on the slide include:

- A circle around the code `a <- 1:4`.
- A circle around the code `b <- 5:8`.
- A circle around the code `A <- cbind(a,b)`.
- A handwritten matrix representation of A: $A = \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$.
- Handwritten vectors: $a = [1, 2, 3, 4]$ and $b = [5, 6, 7, 8]$.

The R Studio environment shows the following:

- Environment:** A (int [1:4, 1:2]) 1 2 3 4 5 6 7 8
- Values:**
a int [1:4] 1 2 3 4
b int [1:4] 5 6 7 8

So, there are some other function to define a matrix. So, this is cbind and rbind. So, this function also you can use to create a matrix. The function cbind create a matrix by binding two or more vector as a column vector ok. So, I will show you what cbind is doing and similarly rbind create a matrix by binding two or more vectors as a row vector.

So, here you just see. So, what I am doing here? I am defining two vector; one is a one is b. So, what is a? a is 1, 2, 3, 4 ok. And what is? b is 5, 6, 7, 8 ok. Now, what I have done here? So, I am defining a, I am defining b. So, a is equal to 1, 2, 3, 4. So, this is a vector and b equal to 5, 6, 7, 8.

So, now, what I have done? I am using cbind. So that means, what it will do basically? So, it will create a matrix by binding two or more vector as a column vector. So, in this case what is

happening? It is a column vector. So, you just see you are getting that is you are getting a matrix which is A and this is a column vector.

So, 1, 2, 3, 4, then 5, 6, 7, 8. So, this is a column vector. So, what I am using here? I am using the command cbind ok. So, you are getting the result here ok. So, this is the matrix I have created using cbind.

(Refer Slide Time: 50:46)

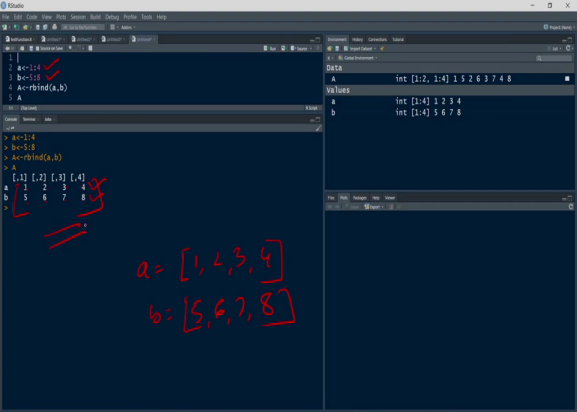
Introduction to R software

Creates a matrix

✓ The function **rbind** creates a matrix by binding two or more vectors as row vectors

```
a<-1:4
b<-5:8
A<-rbind(a,b)
A
```

a = [1, 2, 3, 4]
b = [5, 6, 7, 8]



```
1 a<-1:4
2 b<-5:8
3 A<-rbind(a,b)
4 A
5
```

```
1 a
2 [1] 1 2 3 4
3
4 b
5 [1] 5 6 7 8
6
```

```
1 A
2      [,1] [,2] [,3] [,4]
3 a      1    2    3    4
4 b      5    6    7    8
```

	test [1:2, 1:4]
a	test [1:4] 1 2 3 4
b	test [1:4] 5 6 7 8

Similarly, I can also use rbind ok. So, now, the same problem ok. So, same vector I have used that a vector that is 1 to 4 then b vector that is 5 to 8. So, a equal to this is a equal to 1, 2, ok 3, 4 and similarly b equal to 5, 6, 7, 8 ok, so, 8. So, you can see now I have used rbind. So, rbind a, b. So, I am getting this particular matrix. So, we have two rows now two rows and there are four columns; 1, 2, 3, 4; this is the 1st vector; 5, 6, 7, 8 this is the 2nd vector you are getting the matrix operation. So, here I have added two matrix A plus B.

(Refer Slide Time: 51:35)

Introduction to R software

Matrix operation

A<-matrix(1:6, nrow = 3, ncol = 2, byrow = FALSE) ✓
B<-matrix(7:12, nrow = 3, ncol = 2, byrow = FALSE) ✓
C<-A+B
C

```
# R Studio Console Output
> A<-matrix(1:6, nrow = 3, ncol = 2, byrow = FALSE)
> B<-matrix(7:12, nrow = 3, ncol = 2, byrow = FALSE)
> C<-A+B
> C
      [,1] [,2]
[1,]  8  14
[2,] 10  16
[3,] 12  18
```

Handwritten annotations in the script editor:

- Red checkmarks next to the matrix creation commands.
- Red circles around the variables A, B, and C in the assignment statements.
- Red circles around the variables A, B, and C in the addition command C<-A+B.
- Red circles around the output of the C command in the console.
- Red handwritten text "A+B" next to the console output.

So, first line is creating one matrix that is A then this line is creating another matrix that is B and I have written A and B just to print that one. So, you can see this is the first matrix I have created then I have printed here and this is the matrix A that is 1, 2, 3, 4, 5, 6. So, this is the first matrix and this is the second one 7, 8, 9, 10, 11, 12 and then I have added these two matrix ok. So, that is A plus B.

So, if you are writing A plus B A and then what you are getting? Corresponding element will be added; that means, the first one is 1 plus 7 that is 8, the second one is 2 plus 8 that is 10 then third one is 3 plus 9 that is 12, similarly 14, 16 and 18. So, I am adding two matrix here.

(Refer Slide Time: 52:37)

Introduction to R software

Matrix operation

```
A<-matrix(1:6, nrow = 3, ncol = 2, byrow = FALSE)
A
B<-matrix(7:12, nrow = 3, ncol = 2, byrow = FALSE)
B
C<-A*B
C
```

The screenshot shows the R Studio interface. The console window displays the following commands and output:

```
> A<-matrix(1:6, nrow = 3, ncol = 2, byrow = FALSE)
> A
     [,1] [,2]
[1,] 1    4
[2,] 2    5
[3,] 3    6
> B<-matrix(7:12, nrow = 3, ncol = 2, byrow = FALSE)
> B
     [,1] [,2]
[1,] 7    10
[2,] 8    11
[3,] 9    12
> C<-A*B
> C
     [,1] [,2]
[1,] 7    40
[2,] 16   55
[3,] 27   72
```

The Environment pane on the right shows the contents of matrices A, B, and C:

```
A      int [1:3, 1:2] 1 2 3 4 5 6
B      int [1:3, 1:2] 7 8 9 10 11 12
C      int [1:3, 1:2] 7 16 27 40 55 72
```

Handwritten red annotations include 'C = A * B' and checkmarks.

Now, if I use A star B the corresponding element will be multiplied. So, here 1 into 7 I am getting 7, 2 into 8 I am getting 16, 3 into 9 I am getting 27, similarly I am getting 40, 55 and 72. So, the first line is creating matrix A then this line the third line is creating matrix B and then the fifth line that is basically executing C equal to A star B ok. So, this is and I am getting this particular matrix C.

(Refer Slide Time: 53:17)

Introduction to R software

Matrix operation

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 7 & 14 \end{bmatrix}$$

$\frac{A}{A} \quad \frac{B}{B} \quad A \times B$

$A <- \text{matrix}(1:4, \text{nrow} = 2, \text{ncol} = 2, \text{byrow} = \text{TRUE})$
 A
 $B <- \text{matrix}(c(1,1,2,2), \text{nrow} = 2, \text{ncol} = 2)$
 B
 $C <- A \% \% B$
 C

$$\begin{bmatrix} 1 \times 1 + 2 \times 1 & 1 \times 2 + 2 \times 2 \\ 3 \times 1 + 4 \times 1 & 3 \times 2 + 4 \times 2 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 7 & 14 \end{bmatrix}$$

Now, if I want to do matrix multiplication; that means, this is the matrix A and this is B, I would like to multiply this one A into B ok. So, in that case what I have to do basically? So, the result is 3, 7, 6, 14. So, how I am getting? That 1 into 1 ok. So, this I am getting 1 into 1 plus 2 into 1 ok then I am getting 1 into 2 plus 2 into 2, I am getting 3 into 1 plus 4 into 1 and I should get this is 3 into 2 plus 4 into 2 ok.

So, this value is this is 1 plus 2 that is 3 this is 3 plus 4 that is 7 then this is 2 plus 4 that is 6 and this is 8 plus 6. So, that is 14 ok. So, I should get that one. So, to do that, so, what I have to do? I have to use this particular symbol A star percentage B ok. So, I can do the A cross B. So, finally, I am getting this particular solution. So, this is the matrix A, this is matrix B and I am getting the matrix C.

(Refer Slide Time: 55:04)

Introduction to R software

Function

Functions

- A function is a set of statements organized together to perform a specific task. R has a large number of in-built functions and the user can create their own functions.
- An R function is created by using the keyword **function**. The basic syntax of an R function definition is as follows -

```
function_name <- function(arg_1, arg_2, ...) {  
  Function body  
}
```

Handwritten notes:
- "my fun" written above "function_name"
- "max", "min", "abs" written inside a circle
- "plot" written below the circle

Now, in R I can also write the function ok. A function is a set of statements organized together to perform a specific task. R has large number of inbuilt function. So, it has already inbuilt function suppose you are using max, you are using minimum, you are using absolute abs absolute. So, these are all function inbuilt function and there is a function called plot also ok. So, I can plot it. So, these are all inbuilt functions and the user can create their own function. So, I can create my own function and once I am creating I can use this function just like an inbuilt function.

An R function is created by using keyword function. So, this is the keyword I can use to create a function. The basic syntax of R function defined as follows ok. So, what you have to give? You have to give a name of the function. So, this is function name then I have to use

this particular keyword. So, I cannot change that one. So, this is the keyword I have to I have to use that is the function, but this name you can give anything ok.

So, this is suppose I want to give my function myfun ok. So, this is whatever name you can put it and then this is the argument what is coming into the function ok, argument 1, argument 2, argument 3, something like that. So, whatever the argument you want to give. So, you can give the argument. So, this variables ok will basically will come to this particular function and then there is a function body.

Then I will use second bracket within bracket I can write the function ok. Suppose for example, I can define a function suppose anyway. So, I will go to the next slide and I will basically discuss how to create a function ok. So, this is user defined function that means I would like to create a particular function.

(Refer Slide Time: 57:14)

Introduction to R software **Function**

The different parts of a function are

- **Function Name** This is the actual name of the function. It is stored in R environment as an object with this name
- **Arguments** An argument is a placeholder. When a function is invoked, you pass a value to the argument. Arguments are optional, i.e., a function may contain no arguments. Also arguments can have default values.
- **Function Body** The function body contains a collection of statements that defines what the function does.
- **Return Value** The return value of a function is the last expression in the function body to be evaluated.

So, as I have discussed the different parts of the function are the function name. This is the actual name of the function. So, you can define this, but you cannot define this function as if suppose that name is reserved. Suppose I cannot define a function like max or min or something like that because this is the default function already it is available in R. So, it is stored in R environment as an object with this name ok. So, this name you can define, but some of the name you will not be able to take.

Then there is a argument. As I said an argument is a placeholder ok. So, when a function is invoked you pass a value of the arguments. So, what we are doing? We are passing the value of the argument to the function and arguments are optional ok. So, it means that there may be argument and there may not be argument; that means a function may contain no arguments also.

So, there may not be any argument on a particular function also arguments can have default values. So, sometime this arguments can have also default values. Then second part is function body. The function body contains a collection of statements that defines what the function does ok. So, as per our need, so, I can define this.

So, whatever the objective of this particular function, so, I can define the statements here and the return value the return value of a function is the last expression in the function body to be evaluated. So, it will return the last expression value. So, that will return by this particular function.

(Refer Slide Time: 59:10)

Introduction to R software

User-defined Function

- ✓ We can create user-defined functions in R
- ✓ They are specific to what a user wants and once created they can be used like the built-in functions

```
# Test function
> myfunction<-function(x){
+   for (i in 1:x){
+     y=i^2
+     print(y)
+   }
+ }
> myfunction(7)
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
[1] 49
> |
```

So, let us see an example. So, as I said the we can create a user defined function in R. They are specified to what a user wants and once created they can be used like a inbuilt function built in functions ok. So, once you are creating, so, you can use it as a built in function. So, suppose here I am defining one function. So, this is a test function. So, this is the comment line and name of the function is myfunction.

So, you can give any name. So, I have given myfunction and this is I cannot change, this is the function and this is the argument, argument is x ok. So, x is entering. So, x in this case this is a scalar value and x is entering into the function. So, we are passing the value of x to this particular function and then I am using this second bracket. So, within second bracket I am writing this particular body ok.

So, what I am doing here? For i equal to; I will also discuss this for loop later on, but right now you just see the for i in 1 in x ok. So, 1 to x basically that x is suppose x value is 3; that means, 1, 2, 3 ok. So, 1 to x that y equal to i square; that means, it will square the value of i and then I am printing.

Now, if I execute that one, suppose now the argument I am argument is x and the value I am putting is 7; that means, what you will do? I will go from I will go from 1 to 7; that means, 1, 2, 3, 4, 5, 6, 7. So, what you will do? Suppose first time it will come 1, then 1 square that is 1, then 2 square that is 4, 3 square 9, 16, 25, 36, 49 and then print command will print the value of y ok. So, this function will return the value of y ok. Or here what you see? It is printing the value of y.

(Refer Slide Time: 61:27)

Introduction to R software

Loops in R

For Loop in R

```
for (vector counter)
{
  Statements
}
```

If else statement

```
if ( statement1 )
statement2
else statement3
```

While Loop in R

```
while (constraint condition) // while is a
keyword
//returns bool (true/false) value
{ //opening curly brackets
  //Statements
} // closing curly brackets
```

Now, let us see how you can execute the loop in R. So, here we can execute loop in R using for loop ok. So, this is one option. I can also use while loop and I can also use else if statement ok conditional statement. So, here what we are doing for this is a for loop for vector count and then within second bracket this is the statement. So, this will be executed for this particular condition.

Similarly, if I write while that is the condition I am writing here what constraint condition I am writing here then it will return, so, what you are basically writing within this second bracket ok. And similarly I can also write the conditional statement if else statement that if this statement is TRUE then you execute this one else you execute the statement 3 ok. So, that is the if else statement.

(Refer Slide Time: 62:36)

The screenshot displays the RStudio interface. At the top, a blue banner reads "Introduction to R software" with a red arrow pointing right. To its right, a box contains the text "Examples of Loop". The main editor window shows the following R code:

```
1. for (i in 1:7){  
2.   y<-i^2  
3.   print(y)  
4. }
```

The code is executed, and the console shows the output:

```
> for (i in 1:7){  
+   y<-i^2  
+   print(y)  
+ }  
[1] 1  
[1] 4  
[1] 9  
[1] 16  
[1] 25  
[1] 36  
[1] 49  
+ }
```

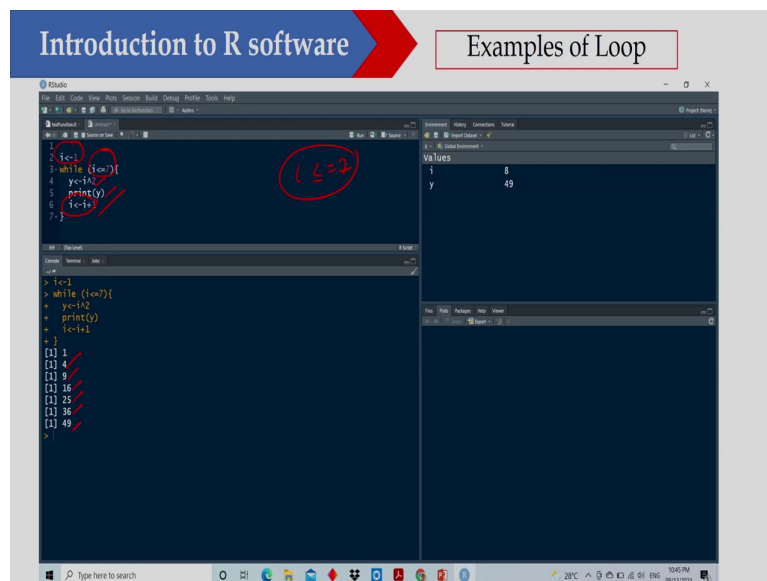
The right-hand pane shows the "Values" section with a table:

i	7L
y	49

Below the "Values" section, the "Functions" section shows "myfunction" as a "function (x)".

So, what I can do? Suppose I would like to execute I would like to print the value of the same thing ok. I would like to print the value of y to do y which is equal to i square; that means, from i equal to 1 to 7 ok. So, now this is a for loop here. So, i in 1 to 7. So, y equal to i square and I am printing that one. So, you just see I am getting 1, 4, 9, 16, 25, 36, 49. Here I can use for loop.

(Refer Slide Time: 63:12)



The screenshot shows the RStudio interface with a slide titled "Introduction to R software" and a sub-header "Examples of Loop". The main editor displays a for loop script:

```
1 for(i in 1:7){
2   i<-i
3   while(i<=7){
4     y<-i^2
5     print(y)
6     i<-i+1
7   }
}
```

Handwritten red annotations include a circle around the condition `i<=7` and a checkmark next to the `print(y)` statement. The console shows the output of the loop:

```
> for(i in 1:7){
+   i<-i
+   while(i<=7){
+     y<-i^2
+     print(y)
+     i<-i+1
+   }
+ }
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
[1] 49
```

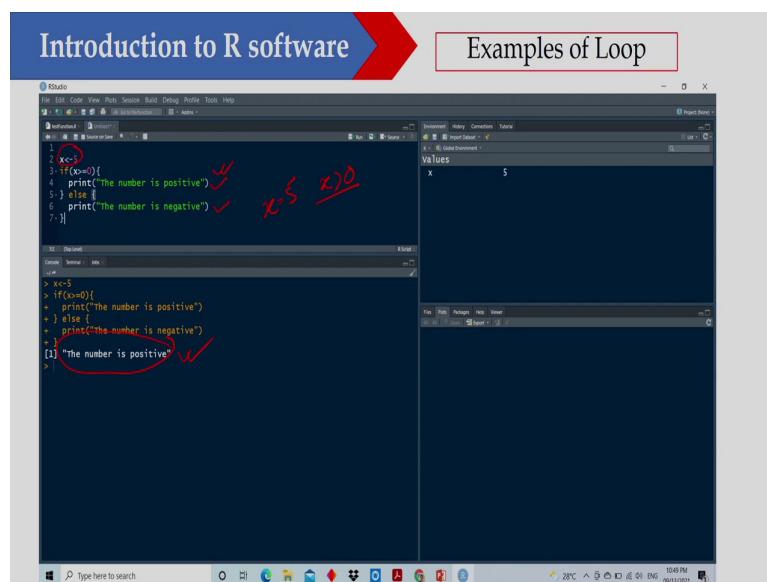
The Environment pane on the right shows the current values of the variables:

Variable	Value
i	8
y	49

I can also use the while loop to do the same thing here. So, I would like to print y which is equal to i square. So, what I have done here? That i equal to 1 and then while this condition is TRUE; that means, i is less than equal to 7; that means, i is less than equal to 7 this condition is TRUE then what I am doing? i equal y equal to i square then print y and then there is a increment that i equal to i plus 1.

So, I am getting the same result; so, 1, then 4, 9, 16, 25, 36, 39. So, I can also use the while loop to execute this particular problem.

(Refer Slide Time: 63:58)



The screenshot shows the RStudio interface. The top-left pane displays the following R code:

```
1 x<-1
2
3 if(x>0){
4   print("The number is positive")
5 } else {
6   print("The number is negative")
7 }
```

Handwritten red annotations include a circle around line 1, a checkmark next to line 4, and the text "x=5" and "x>0" next to line 5. The bottom-left pane shows the console output:

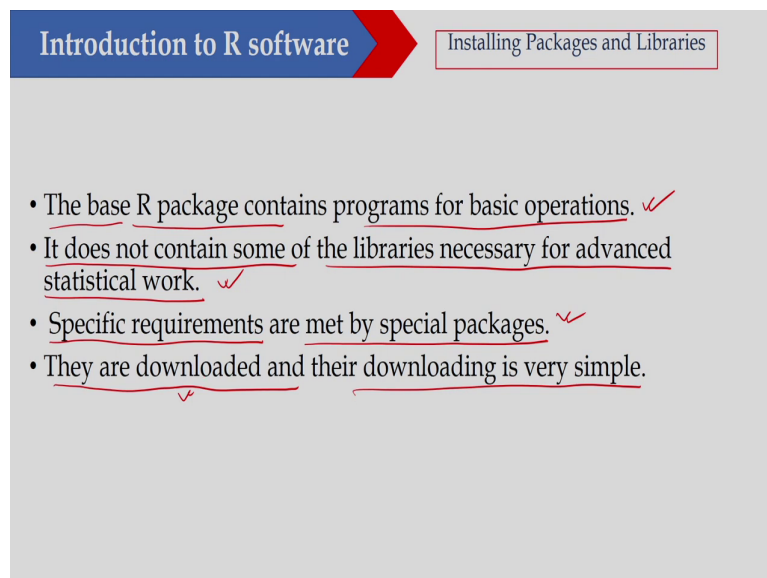
```
> x<-5
> if(x>0){
+   print("The number is positive")
+ } else {
+   print("The number is negative")
+ }
[1] "The number is positive"
```

Handwritten red annotations include a circle around the output string and a checkmark. The top-right pane shows the "Values" environment with a variable x assigned the value 5.

Now, let us see the if else statement. So, here what I am doing here? I am defining a value x equal to 5. Now, what I am doing? If x is positive then I am printing the number is positive else the number is negative. Now, if x is positive ok, so, in that case it will print positive else it will print negative.

So, in this case the I am putting x equal to 5. So, x equal to 5. So, this is positive. So, therefore, I am getting the number is positive. So, therefore, I can also use these if else statements to execute some of the statement like this.

(Refer Slide Time: 64:40)



Introduction to R software

Installing Packages and Libraries

- The base R package contains programs for basic operations. ✓
- It does not contain some of the libraries necessary for advanced statistical work. ✓
- Specific requirements are met by special packages. ✓
- They are downloaded and their downloading is very simple. ✓

Now, as I said so, whatever I have; whatever I have discussed here, so, this is the basic R engine ok. So, as I said earlier the some of the advanced facilities are not available. Suppose I would like to work on GA ok. So, GA is not available here so, but GA package is available ok.

So, you can download that package you can install that package and you can include in your R platform and you can use it. So, the base R packets contain program of basic operations, so, that as I have discussed. So, it contains only the basic operation. You can write your own code. It does not contain some of the libraries necessary for advanced statistical work. So, that is not there.

Specific requirements are met by specific packages ok. They are downloaded and they are downloading is very simple. So, I can download these packages and downloading is also very simple and then I can install it ok.

(Refer Slide Time: 65:46)

Introduction to R software

Installing Packages and Libraries

- The base R package contains some necessary libraries only. Other libraries are required for advanced statistical work which are downloaded and installed as and when required. Run the R program, then use the `install.packages()` command to download the libraries
- Examples
 - `install.packages("GA")` : installs package GA
 - `install.packages("Nlcoptim")` : installs package Nlcoptim
 - `install.packages("deconstructSigs")` : installs package deconstructSigs

The base R package contains some of the necessary libraries only. Other libraries are required for advanced statistical work which are downloaded and installed as and when required. So, as and when required, so, you need not do because sometime you need that one. So, at that time you can download it and you can install it. And run the R program then use install package.

So, this is the command I can use to install a particular package ok. So, this command I can use. Suppose in the console itself I can write or in the program itself I can write that I would like to install this particular package. So, for example, I can suppose I would like to install

GA package. So, name of the package. So, this you have to know what is the name of genetic algorithm package ok.

So, what is the name of this particular package? So, name is GA. So, what you have to do? You have to write this particular command that is install packages and within code so, you have to write GA and this will install GA on your computer. Similarly, there is another package which is non-linear optimization toolbox actually automation package. So, this is Nlcoptim.

So, it will you can install using this one and there is another one deconstructSig. So, that also you can use. So, you have to use the install package command to install the packages. So, what will happen? So, in the program itself you can write that one.

(Refer Slide Time: 67:32)

Introduction to R software Libraries in R

- ✓ To use a library, type the library function with the name of the library in brackets.
- ✓ Thus to load the GA library type library(GA)

Similarly,

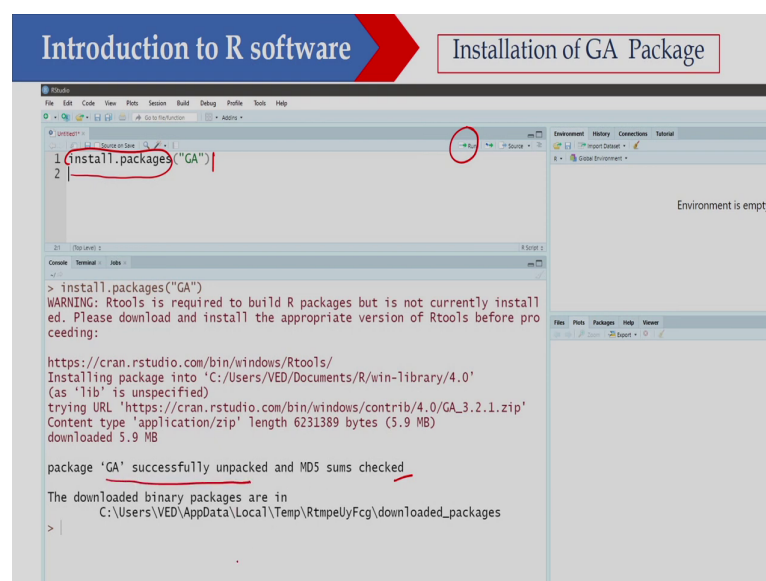
- ✓ library(GA) : loads package GA ✓
- ✓ library(Nlcoptim) : loads package Nlcoptim ✓
- ✓ library(deconstructSig) : loads package deconstructSig ✓

library (GA)

So, it will install the package and after installation what you have to do? You have to include that library to use the library type the library function with the name of the library in the brackets. So, thus to load the GA library type: library GA. So, in the program itself, so, what I will do? So, in the program itself I will write library GA ok. So, GA and this will include the library on your program and you can use it.

So, similarly, so, load this is GA, this is Nlcoptim and this is deconstructSig ok. So, this is. So, you have to know the name of the library. So, you cannot say in suppose this is in capital letter you have to use capital letter and here it is S is capital. So, you have to use capital letter otherwise it will not work ok. So, you can include the library here ok.

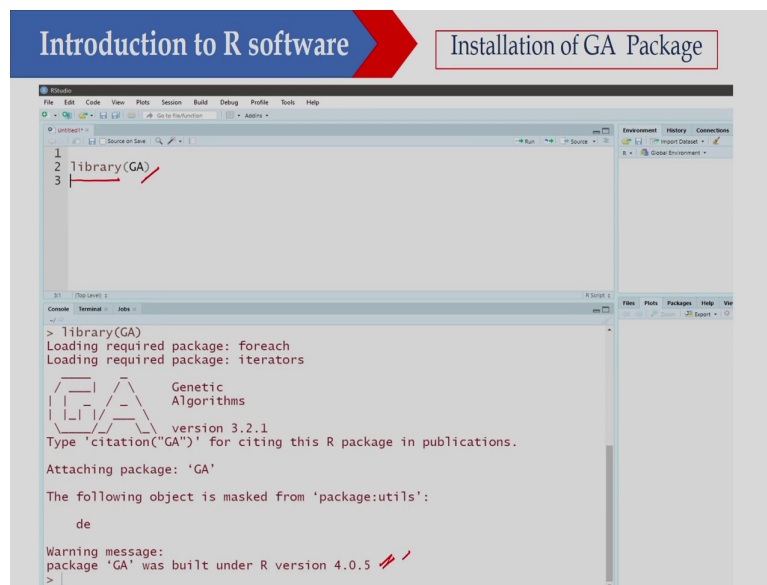
(Refer Slide Time: 68:37)



So, here you can see that as I said. So, I can install the GA package by using these particular command. So, install GA. So, if I execute this particular line using the Run button, so, you

can see that it will automatically. So, if you have the net connection on your computer, so, it will automatically download that one and it will install in on your computer. So, you can see the package GA successfully unpacked and it has been installed ok.

(Refer Slide Time: 69:07)



Now, then when we are using that one, so, I have to include that library. So, how I am doing that one? I am writing library GA. So, this will basically use or this will include the library on your program ok. So, on that particular program this will be or on your working environment this will be added. So, GA has been included here.

So, thank you very much. So, in this particular class, so, I have introduced you what is R, how you can download R, then again as I said that we are not working on R engine, but we are working using R studio. So, both are freely available. So, first you download the R engine and then you download R studio, but we are working on R Studio.

Then we have discussed the simple operation how you can create a vector, how you can create a matrix, how you can do the vector manipulation, how you can do the matrix manipulation so that we have discussed we also discussed how to create user defined function how you can create a function and this function once you are creating. So, you can use this function as a built in function. So, that you can use it and also we have discussed the different packages.

So, when I will solve the optimization problem using some of these packages, suppose I will solve the genetic algorithm using R. So, there I will show you how you can install the genetic algorithm package and then how you can include that one in your program and how you will use that one.

So, that I will discuss and so, this packages you have to download or you have to download and you have to install before using that one before including that one. So, if you are writing library, suppose without installation if I write library GA, so, in that case it will show you error because you did not install the library GA library on your computer. So, first you have to install it and then you can use that one. So, let us stop here.

Thank you.