

Marketing Analytics
Dr Swagato Chatterjee
Gupta School of Management
Indian Institute of Technology, Kharagpur
Lecture 01: Introduction to R Programming

Hello everybody. Welcome to Marketing Analytics course. This is the first module- Introduction to R Programming and my name is Dr. Swagato Chatterjee, Vinod Gupta School of Management, IIT Kharagpur. I will be taking this course for you.

Introduction to R Programming; so first we have to discuss before we jump in this thing that why are we doing R programming. We will do hands-on of marketing analytics. Now hands-on can actually be done using various software. But we have chosen Excel for some of the smaller problems and R programming for a bigger problem.

Now Excel is something that is almost inevitable in today's management era. All people, I would say all managers actually use spreadsheet modeling. And even the academicians who teach managers, future manager will also be required to know the intricacies of Excel. And we will actually majorly focus on how the various features of Excel can be used in marketing problem solving.

On the other hand, the problem with Excel is that it becomes a little bit of limited when the data size is big. If the data size, the number of rows is more than 6 lakhs, Excel actually have some problem. So it is better to use a software which is handy to deal with a little bit of larger size of data.

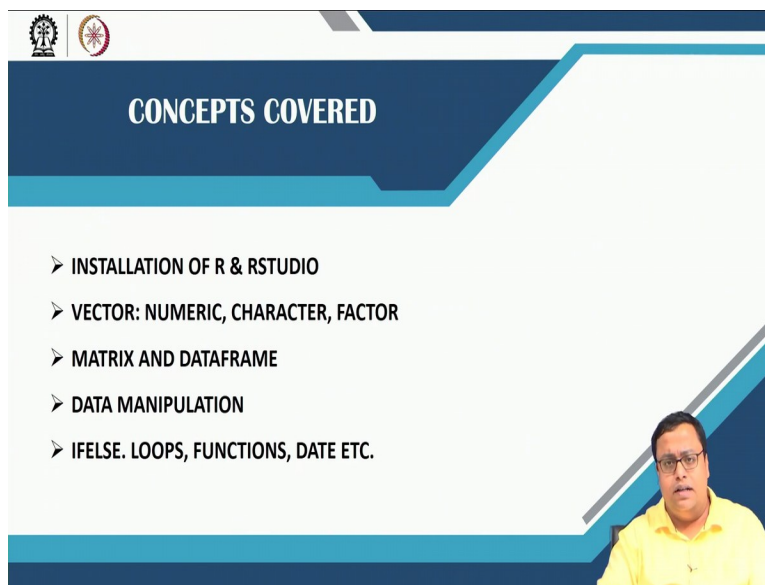
Now we have multiple software propriety and non-proprietary, open source which we have in our hand. But we have chosen R, because one reason is, it is open source. The second reason is it has huge support. So there are lots of resources available online. You can learn it on your own. We will be actually teaching you a little bit. But you can do on your own as well.

Another option is Python, obviously. But Python is more used in; for the deployment; when you create a software which will be deployed for an automated kind of problem

solving. But R is more good for research oriented work. And Marketing Analytics often backend research oriented work, so we will focus on R Programming.

Now R I told that it is online. It is the open source, available online, freely downloadable. So before we jump in, we will have to learn how to download it and how to install the R Programming. We will have a few sessions on R Programming, before we jump in the actual marketing analytics, so that you become handy with the software.

(Refer Slide Time: 02:56)



The slide features a dark blue header with the text 'CONCEPTS COVERED' in white. Below the header, a list of topics is presented in a light blue font, each preceded by a right-pointing arrowhead. In the bottom right corner, there is a small video inset showing a man in a yellow shirt speaking.

- INSTALLATION OF R & RSTUDIO
- VECTOR: NUMERIC, CHARACTER, FACTOR
- MATRIX AND DATAFRAME
- DATA MANIPULATION
- IFELSE. LOOPS, FUNCTIONS, DATE ETC.

So the first thing is, this particular presentation will show you how to install R and R Studio and I will discuss about R Studio also. And then today we will cover these aspects like vector matrix, data manipulation and little bit if-else functions in a few probably one or two sessions.

(Refer Slide Time: 03:16)

INSTALL R

- Go to: <https://cran.r-project.org/bin/windows/> and click on “install R for the first time”
- Download the latest version available. Currently it is R3.6.1
- Install

NPTEL Online Certification Courses
IIT Kharagpur

Now installation of R that is the first job. If you want to install R, you have to go to this particular link. You can also go to Google and search for Cran R or something like that or R download or something like that. It will ultimately lead you to this particular link. I have chosen Windows because I use Windows. If you use some other ways, then you can use corresponding R software.

So you have to go to this particular link and click on “install R for the first time”. It will be written there, you have to click on that. That will actually direct you to the latest version of R available, currently the latest version is R3.6.1. So once you go there, you actually download R3.6.1. When we actually see this video, probably more recent version of R might be available. If it is available, download that. Now after downloading it, you have to install by double clicking on the installer like you do for any other exe file and it will get install.

Now we actually use R Studio over R. R has its own UI- user interface also. But we use RStudio because I have seen; it is my personal opinion and probably many of, many other people will actually agree with me that RStudio’s UI is more user-friendly than R’s UI. There are lots of more options available, lots of more drag and drop kind of options or

click kind of options are available which makes it easier. And, so RStudio is also free, it is open source software. So it is better to use that.

(Refer Slide Time: 05:02)


INSTALL RSTUDIO

- Go to: <https://rstudio.com/products/rstudio/download/> and choose RStudio desktop (Free version)
- Download the latest version available depending on your OS.
I prefer Windows and the choice is "RStudio 1.2.5001 - Windows 10/8/7 (64-bit)".
- Install

NPTEL Online Certification Courses
IIT Kharagpur

So we go to this particular link to download R Studio. So when you go to the link, there are various versions of RStudio available and we will use this RStudio desktop which is the free version. And again in the free version for various OS lots of different kinds of installers are available. We will use the RStudio, the latest version for Windows 10 or 8 or 7 because I am doing it for Windows. So the latest version currently available is RStudio 1.2.5001. But when you do it, further latest version can come, whichever is the latest version available, install that.

(Refer Slide Time: 05:53)



INSTALL RSTUDIO – 32 bit system

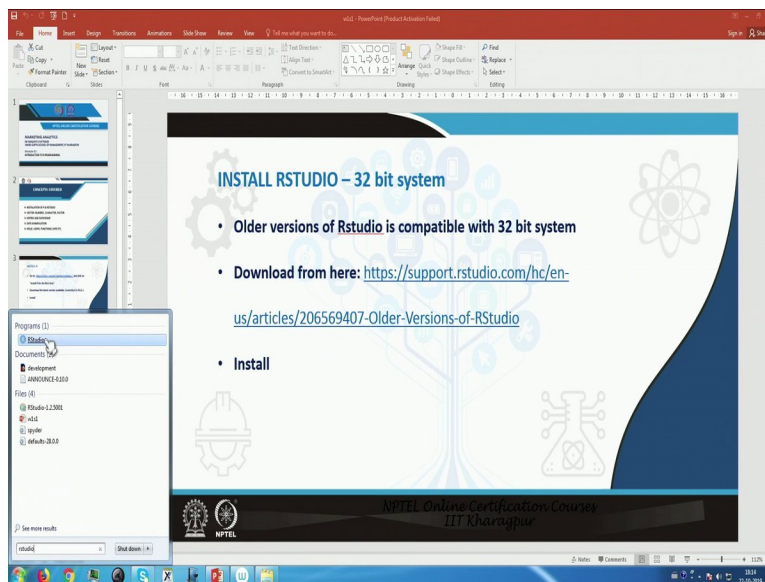
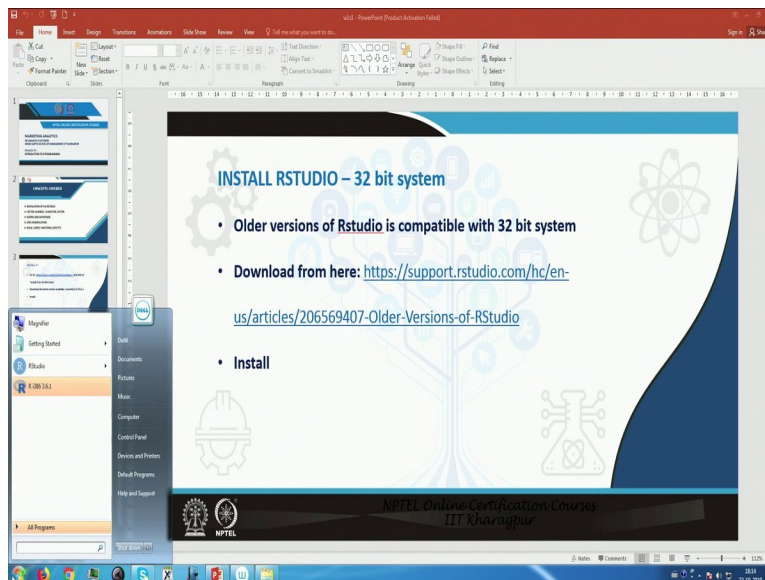
- Older versions of Rstudio is compatible with 32 bit system
- Download from here: <https://support.rstudio.com/hc/en-us/articles/206569407-Older-Versions-of-RStudio>
- Install

The slide features a background with various icons including gears, a tree, a hard hat, and a circuit board. A video inset in the bottom right corner shows a man in a yellow shirt speaking. The footer contains the NPTEL logo and the text 'NPTEL Online Certification Courses IIT Kharagpur'.

Now problem with RStudio, little bit of problem with RStudio is that newer versions are more focused towards 64 bit system. Now by chance if you have a 32 bit system, you have to go to this link. And then the same thing, here the older versions are there. The version which is most recent but older, but at the same time handles 32 bit system can be downloaded from here.

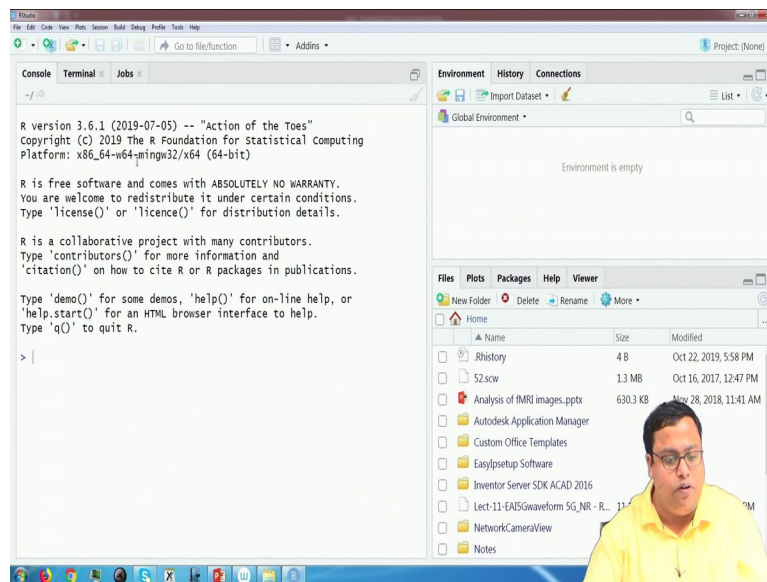
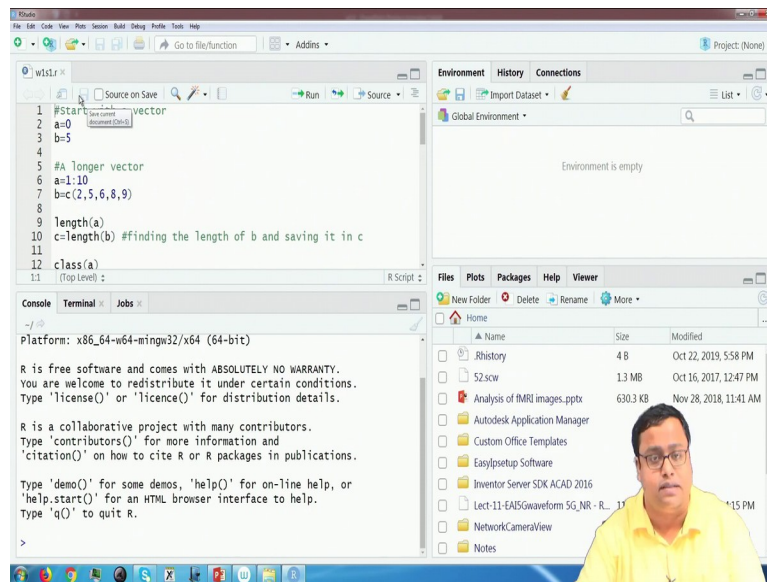
So you can download that and install that. For the Marketing Analytics purpose, both of these RStudio whether it is 64 bit or compatible with 32 bit, both of them will work. So we have no problem. Install any one of them. So once you have installed this, we will be able to go ahead with learning basic R Programming which will be used in Marketing Analytics course as we proceed.

(Refer Slide Time: 06:48)



Now I will show you how to start with the RStudio. So once you have installed it, you can go and click on this Start button. RStudio might be shown there or you can just search for RStudio. And RStudio will come, you can click on that. And once you click on that, something like this open.

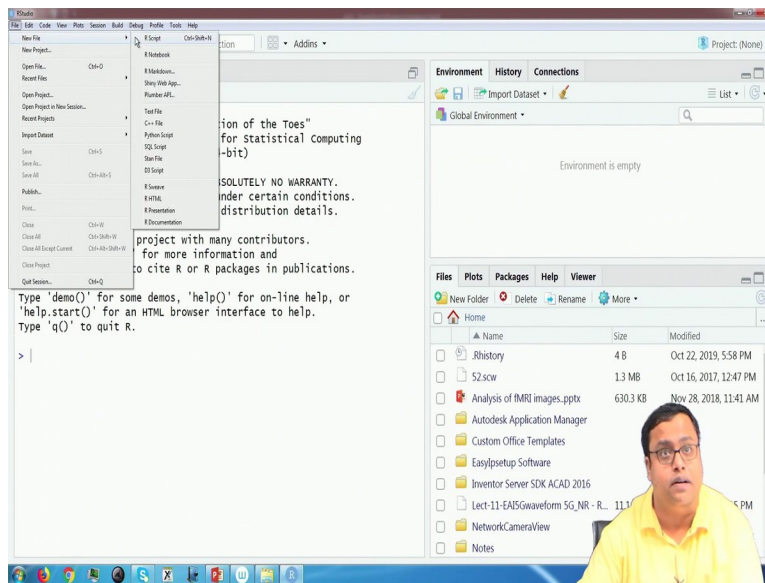
(Refer Slide Time: 07:02)



Now for your case; for my case there was already one file was open. But most of you, if you are doing for the first time, this is something that you will see. This is a view or probably something similar to this view is what you will see. Now this is an UI, as I told that it is more friendly than R's own UI and there various aspects of this particular UI. Currently you can see there are three boxes. Two here in this part and one is this side. And I will actually explain step by step, what are the jobs of these particular boxes.

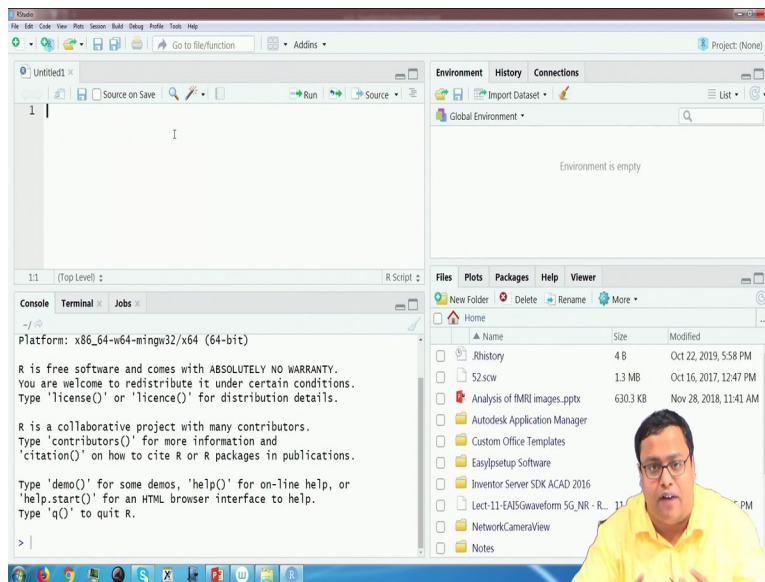
Now in any software, whether is it let us say, Microsoft Word or Excel or whatever, you want to first start with a blank page, where you will write anything and save that particular file. Now here also we will start with a blank page that is the first job that we have to do open a blank page.

(Refer Slide Time: 08:15)



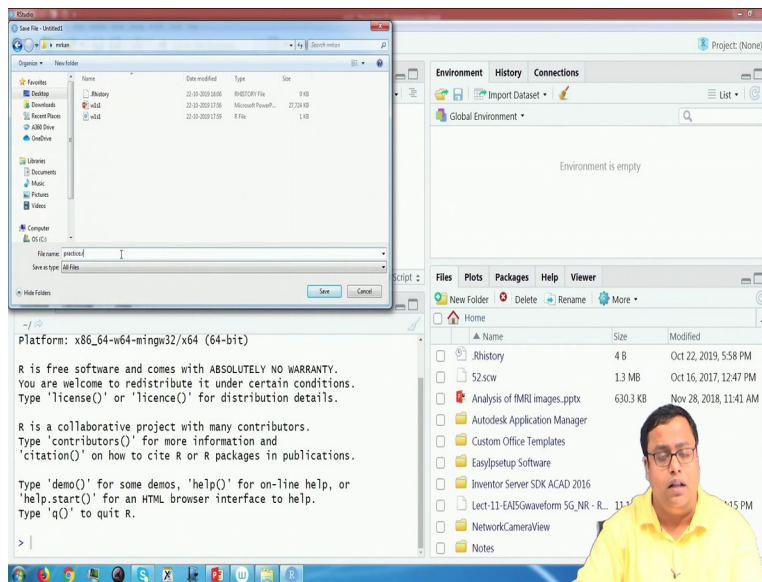
So to do that what I will do is, I will you to the file at the corner, left corner you will see file and then new file and R script. File, New File, R Script, you can also click on Ctrl+Shift+N. So that will open something like this.

(Refer Slide Time: 08:25)



It is a new untitled file. That means it is a new file where no name is there. Now it is a good practice that I believe, that when you have opened a new file in let us say Microsoft Word or Excel, it is a good practice that you save it. You save it, so that later point of time if you write something, you just press Ctrl S and it get saved. So otherwise, by chance your computer hangs or by chance; sometimes it is a programming language, so if you run a program, sometimes it can get probably disturbed. So that you do not lose your data, you do not lose your code, it is a good practice to start saving from the very first.

(Refer Slide Time: 09:12)



So what I will ask you to do is to click on this button, which is a blue button you can see. It is a floppy disc kind of button. If you can click, it is a save button actually. If you click on that it will ask you where to save. And you choose any where. So I will probably save it on my desktop. There is something called, I have created a folder. And I have probably write something like “practice.r”. Now once I write that and save that, this particular file gets saved.

Now you are ready to use RStudio. So there are lots of options and not all the option can be taught on the first day. We will slowly see that what are the various options available, how we can go ahead as per our requirement.

So first of all, there are four boxes that you can see. So one box is here, one box is here, one box is here and another box is here. These are the four boxes that you can see. Now these four boxes, I love to say that they are four quadrants of my screen. Now each of these quadrants has some job to do.

The second quadrant which is “practice.r” where there is written is an editor. This is where you write your code. This is where you save your code, so that you can use it at a later point in time. So this is where you write your code.

On the other hand, this part there are three tabs that you can see console, terminal, and jobs. We will talk about console. Console is a place where your code runs. So when you run the code, all your results come in the console. Now in the right hand side, there is first quadrant and fourth quadrant.

The right hand side, at the top what I love to call as first quadrant is where the environment is there, history is there, and etc. So I will focus on the environment part. So history is also there and connections is also there. But environment part I will focus on.

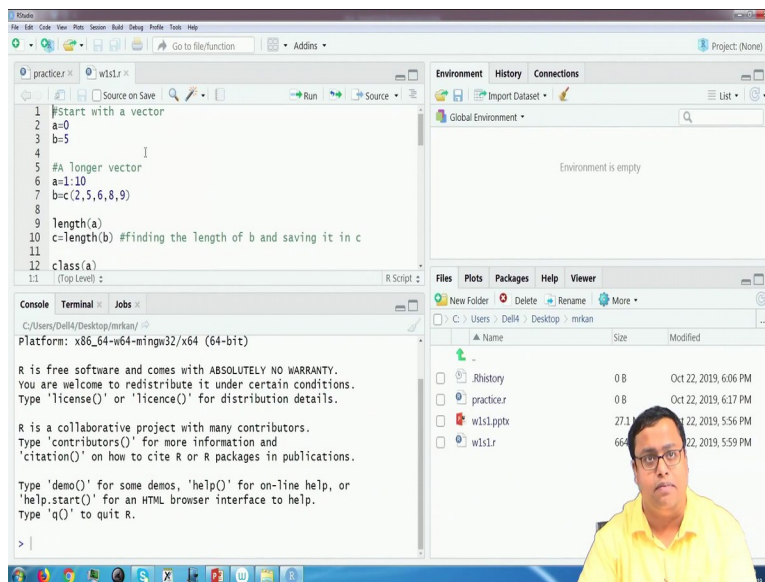
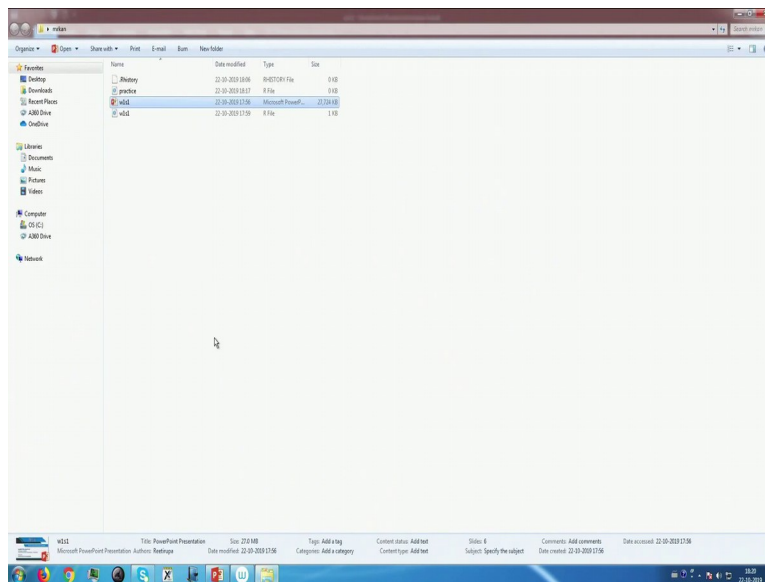
Environment is where, whatever data set, variable, matrix that you want to save, so that you can use it later can be stored. For example, if you have ever done any coding, let us say in C or C++, you have done “int”, int i is equal to 0. So that i is equal to 0 is something that in the name of i you are saving some value. So that is that i value will be saved in this global environment.

And in the fourth quadrant, there are lots of tabs like files is one tab, and then plots is another tab, packages is another tab, and at the right time I will discuss about all these tabs. So these are the four quadrants as I told. Here I will write and here I will run the code.

Now, I have already written a set of codes. But I would ask you that, it is a good practice that you copy the code from there or type it on your own and run the codes. Otherwise you can follow the code that I have shared with you. But it is very good practice that you type on your own because when you type on your own, you do the mistakes.

And when you do the mistake, you learn from those mistakes. It is very important to do mistakes as until and unless you do mistakes in coding, you will not learn how to code. So it better that you type on your own. Do mistakes; learn what kind of things you have done. Because in code, I will write all of these things are which are right and that will not teach you anything. So it is better you write the code on you own. Whatever I am showing there, you write on the editor manually on your own.

(Refer Slide Time: 13:00)

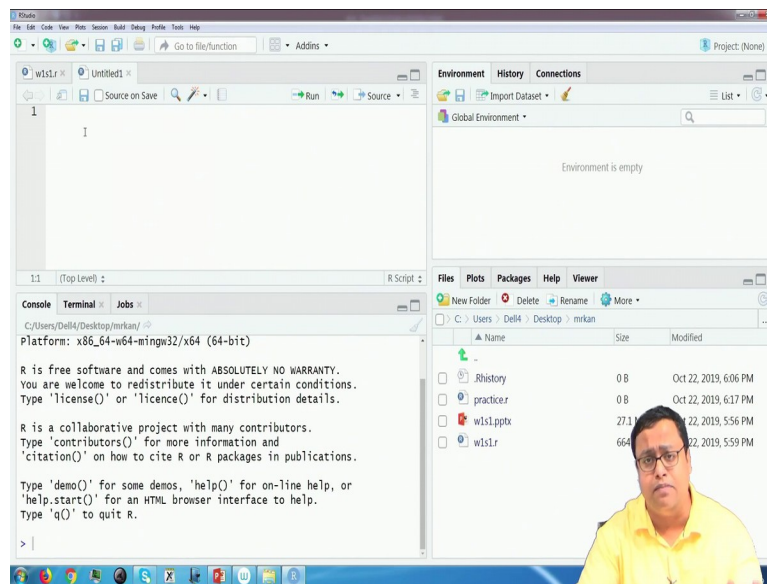


So in the file section, you find out there is file called w1s1.r, so week1session1.r. So I am double clicking on that file and it will open something like this. So other than that file, I have opened practice.r previously. I am closing it. So at this moment I am closing all the things.

It is again a good practice. So you will see that some people when they work on Word, multiple Word files remains open. Even if some file which he is not writing right now,

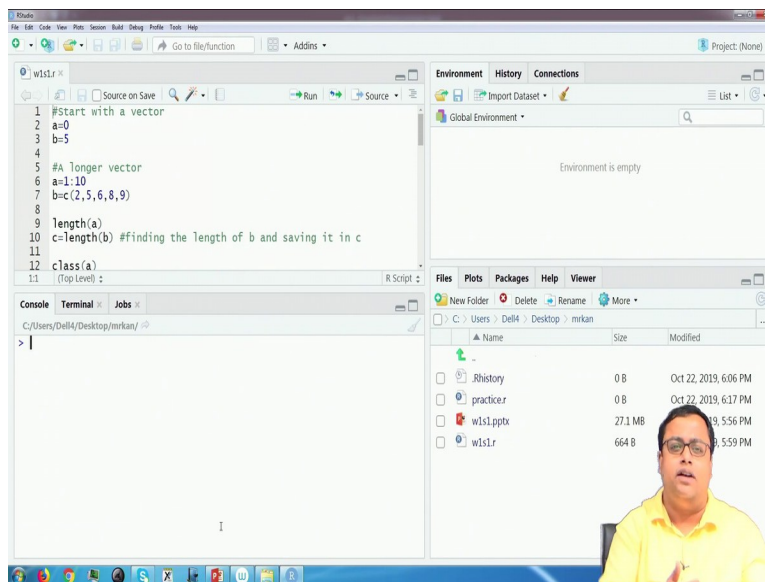
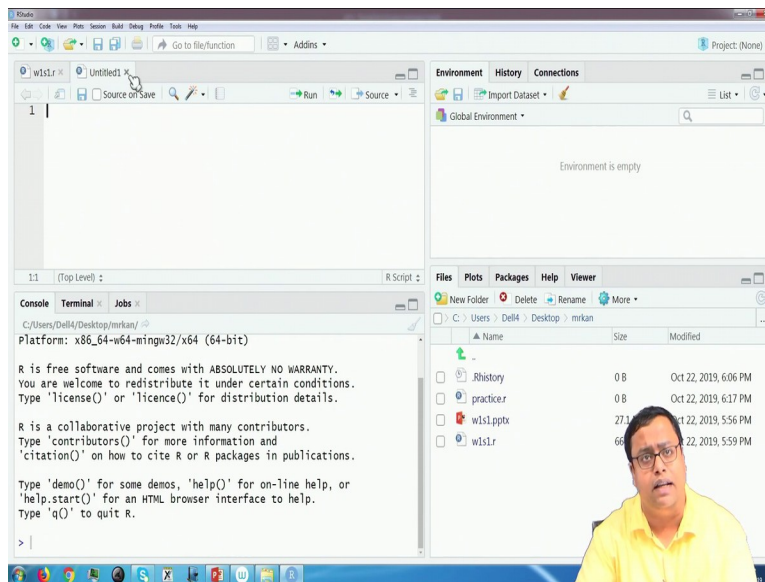
still remains open. What happens is, by chance once the word hangs, then all of those things will have a problem, all the files will have a problem. So it is a good practice to keep only that file open which you are using and close all other files. So closing is nothing, just clicking on this cross sign. So if you by chance have any other tabs open here, close that and `wls1.r` is something that we will work on.

(Refer Slide Time: 14:12)



Now those who want to code on their own as I told it is better if you want to type on you own, so then it is better that you open a new file, then save this file and type it here. So whatever is written here, you can copy it one by one and type it here and then run it. So then you will know what kind of mistakes you are doing. Sometimes it is better not to copy or probably type on your own, then you will know what kind of mistakes you are doing and how you can rectify those mistakes.

(Refer Slide Time: 14:39)



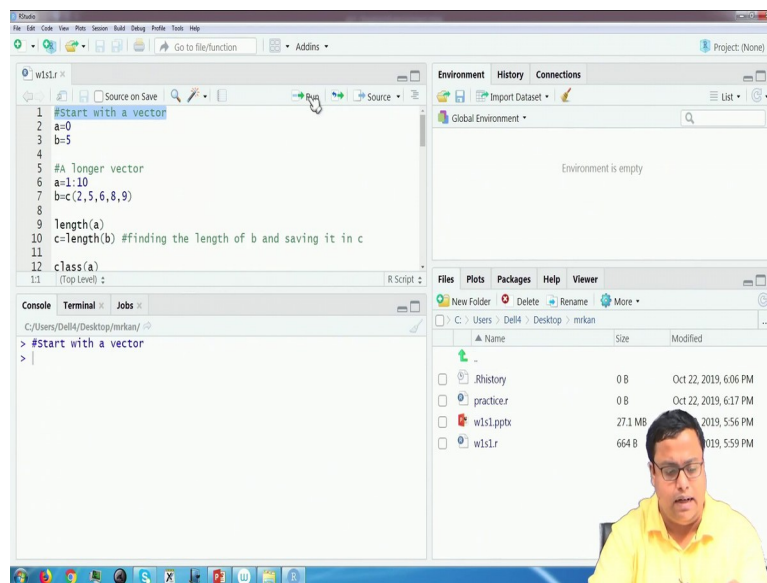
Now let us say you have written these codes. And I will come one by one. So first good job is to cleanliness, cleanliness is another very important thing for any coding because then you will get less confused. So do not write down anything, whatever I am telling now, do not write down anything. Again, it is a good practice to learn coding by practicing rather than by memorizing. It should come from your inside that okay this kind, or if it is not coming from the inside, you should have a resource to fall back on, and that should not be something which is your notes. So do not write down.

So the first thing that I will do is I will clean my console and to do that what I will do is I will press Ctrl+L. So I just press Ctrl+L to clean the console at any point of time. So often time we actually write lots of codes in the editor and run the codes. And we run multiple codes to see that which code is actually working, which code is actually giving out the outputs that I want. Now previous ones I do not need, so then I will just press Ctrl+L to clean my console. Console is where the codes run.

So now, let us start with this thing. So the first thing that you have to understand in R is that R has certain objects like one object is called vector, another object is called matrix; another object is called datasets. So depending on dimensions, depending on the contents, depending on various other aspects, the objects differ. So the basic object, the most basic object of R is called a vector or a variable.

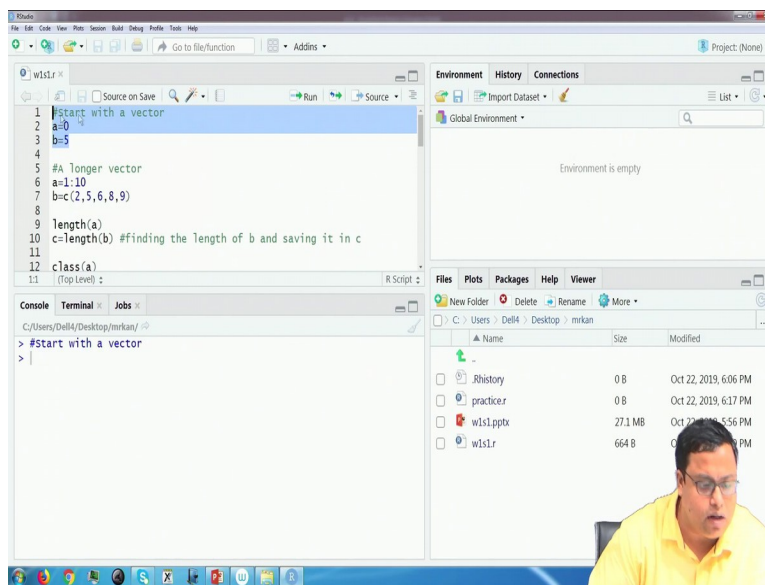
So you can imagine, vector or variable is in Excel, it is one column of an Excel or one row of an Excel. So it is better to imagine one column of an Excel which has the name at the top of the column, and then there are certain values in it. It can have one value; it can have multiple values. So even if a particular, some name contains only one value that is a vector, multiple values also that is a vector.

(Refer Slide Time: 17:25)



So for example I have written this “Start with a vector”. So this is a comment. Anything that starts with a hash sign is a comment. So comment means it will run but it will not give any result. So how to run it? There are two ways of running it one is you can select the area that you want to run, the code that you want to run, select and press on this run button. See the moment I click run, it came down in the console “Start with a vector”. It got run but it is a comment. It will give no output. Nothing changed. So that is the first step.

(Refer Slide Time: 18:00)

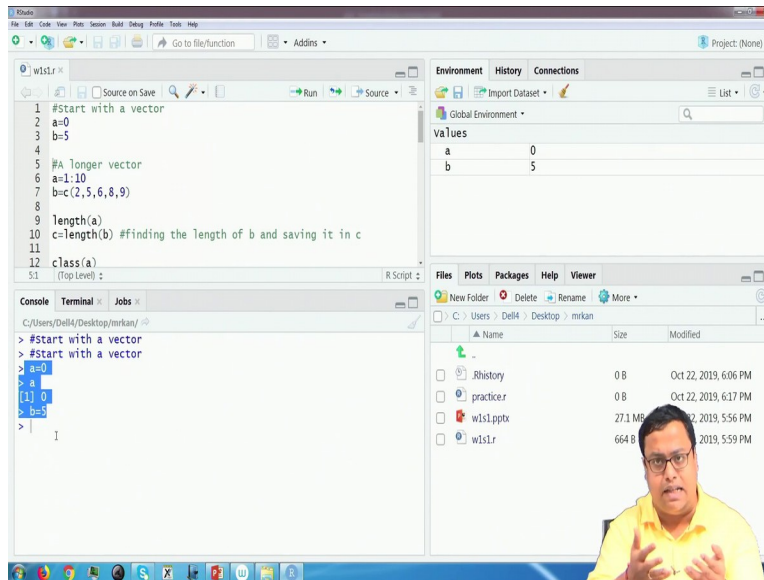


Now, if you want to run two-three lines sometimes together, one at a time two-three lines, you have to select probably the whole area; that is, I want to run all these three and then click on run button, it will also run. Now a good practice is to select the area. Again, I believe that a good practice is to select exactly the portion of the code that you want to run. Sometimes we want to run line by line, so instead of selecting the whole line, I can just put my cursor and then press run also.

If I just put my cursor and press run, it will run one single line of code. So here, the second line of code is a is equals to 0, that will get run. So it gets run, a=0, and the moment I run that a=0, you will see that in the global environment, 0 gets saved in the

value of a, means the name is a; there is a vector gets created whose name is 'a' and the corresponding value that gets created, that gets saved is 0. How will I use that?

(Refer Slide Time: 19:02)



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
1 #Start with a vector
2 a=0
3 b=5
4
5 #A longer vector
6 a=1:10
7 b=c(2,5,6,8,9)
8
9 length(a)
10 c=length(b) #finding the length of b and saving it in c
11
12 class(a)
13
```

The Environment pane on the right shows the following values:

Variable	Value
a	0
b	5

The console shows the following output:

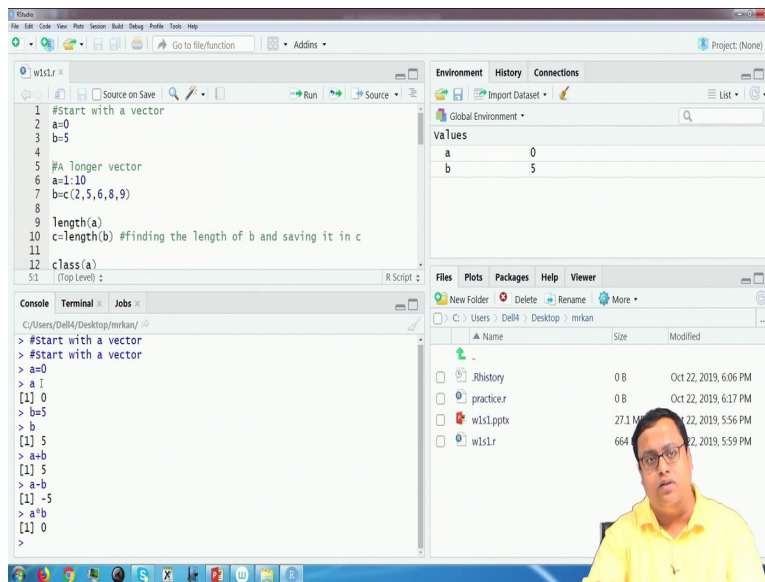
```
> #Start with a vector
> #Start with a vector
> a
[1] 0
> b
[1] 5
```

If I write in my console, just 'a' and then press an enter. I have written 'a' and press an enter, it gives me 0. It gives me the output as 0. It is like print A; it gives me the output as 0. There is something written in the third bracket of 1, I will discuss this later. So whatever is this guy is I will discuss about it later; at later right point, I will discuss.

Similarly, if you have done that how to store a value of 5 in 'b', so I have written b= 5 and if I have written b= 5, I will run this. And see here, in 5 gets saved in the name called b.

See; just check these 3-4 lines, these 3-4 lines. So the first line is a =0. So 0 gets saved in 'a' but nothing comes as output. In the next line, I have written 'a', and pressed an enter. Now I am calling 'a'. So whatever value is in 'a', comes out. Next again, I have written b=5. So 5 get saved to 'b' but no output is there. It is just some code that has run, and something has been done, but nothing comes out. Now, if I ask the output, 'b' and press an enter, then only it gives me the output of 'b'.

(Refer Slide Time: 20:27)



The screenshot displays the RStudio interface. The script editor on the left contains the following R code:

```
1 #Start with a vector
2 a=0
3 b=5
4
5 #A longer vector
6 a=1:10
7 b=c(2,5,6,8,9)
8
9 length(a)
10 c=length(b) #finding the length of b and saving it in c
11
12 class(a)
13
```

The console on the bottom left shows the execution of the first two lines of the script:

```
> #Start with a vector
> #Start with a vector
> a=0
[1] 0
> b=5
[1] 5
> a+b
[1] 5
> a-b
[1] -5
> a*b
[1] 0
>
```

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

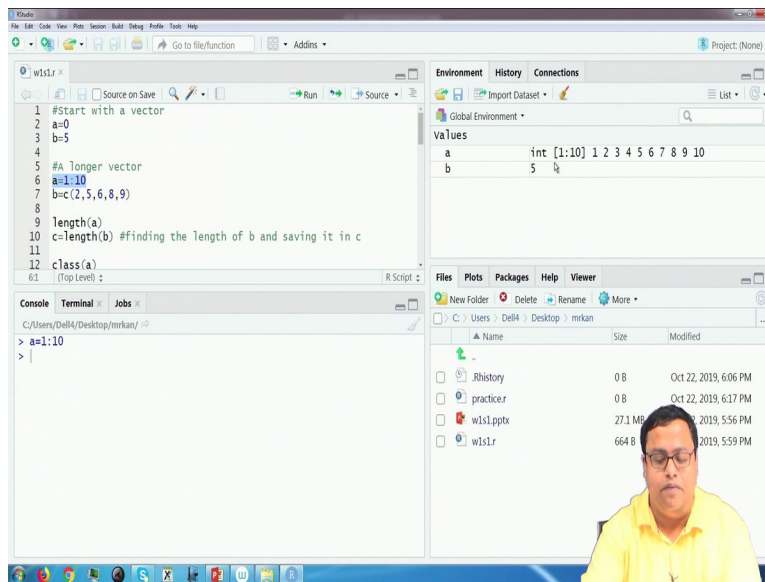
Variable	Value
a	0
b	5

The Files pane on the bottom right shows a directory listing for the user's desktop:

Name	Size	Modified
..		
..history	0 B	Oct 22, 2019, 6:06 PM
practice.r	0 B	Oct 22, 2019, 6:17 PM
w1s1.pptx	27.1 MB	Oct 22, 2019, 5:56 PM
w1s1.r	664 B	Oct 22, 2019, 5:59 PM

How can I use this? Let us say, I want to know what is $a+b$; what is the summation of $a+b$? I will write $a+b$ and press an enter. It gives me the output of $a+b$, so a is 0, b is 5 adds up. Similarly $a-b$, it gives me -5 and so on. $a * b$, will give me 0 and so on. So all of these things $a + b$, $a - b$, $a * b$; all of these things are actually giving me certain outputs. But a is equal 0 and b is equals to 5 is actually storing some value in 'a' and 'b'. So that is the first step. So these 'a' and 'b' are two vectors.

(Refer Slide Time: 21:29)



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
1 #Start with a vector
2 a=0
3 b=5
4
5 #A longer vector
6 a=1:10
7 b=c(2,5,6,8,9)
8
9 length(a)
10 c=length(b) #finding the length of b and saving it in c
11
12 class(a)
6:1 (Top Level):
```

The Environment pane shows the following values:

Variable	Value
a	int [1:10] 1 2 3 4 5 6 7 8 9 10
b	5

The Console shows the following output:

```
> a=1:10
>
```

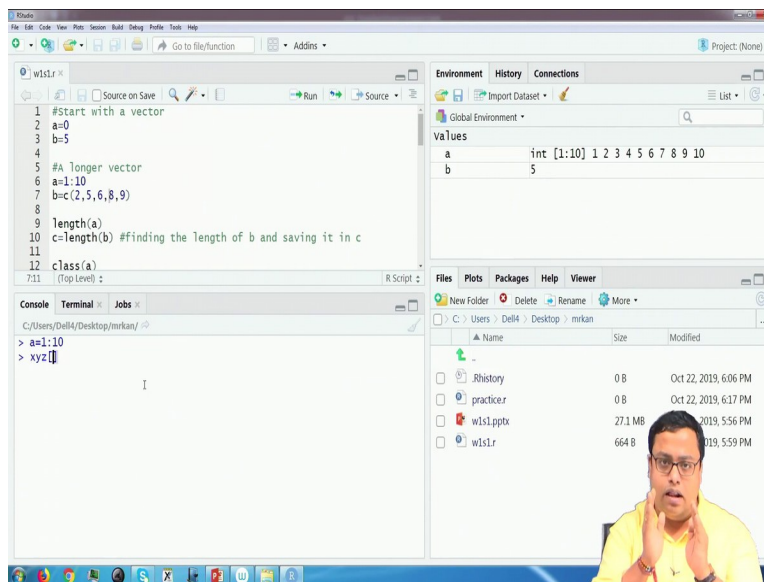
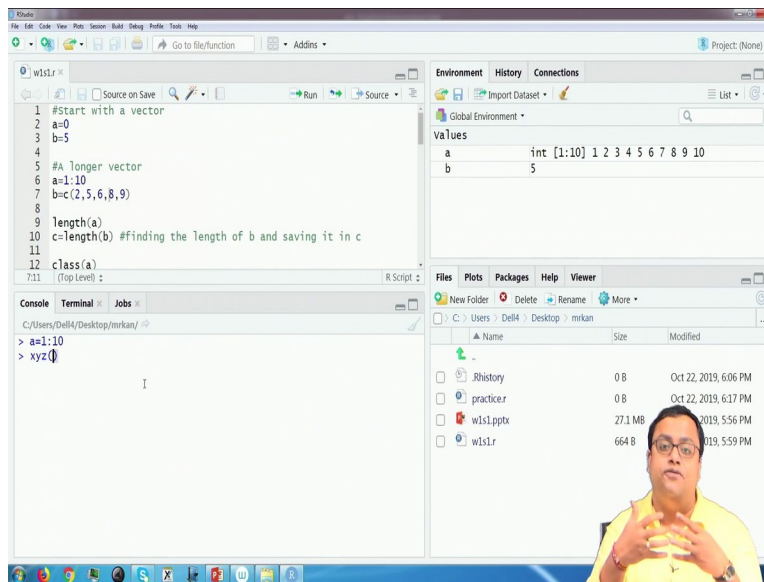
The Files pane shows a directory listing for 'mrkan' on the Desktop:

Name	Size	Modified
..		
Rhistory	0 B	Oct 22, 2019, 6:06 PM
practice.r	0 B	Oct 22, 2019, 6:17 PM
w1s1.pptx	27.1 MB	2019, 5:56 PM
w1s1.r	664 B	2019, 5:59 PM

Now vectors can be little bit longer also. In real life situations, our columns in Excel are longer than one single value. So here I have created two ways. One is, I have written `a=1:10`. So I will select this, and then I will run. So the moment I run, you see that here the previous value of 'a' gets changed. So 'a' is equal to; a is an integer. It has been written 'a' is an integer. Int stands for integer. This 1:10 is that it is saying that it has 10 units, 10 values, 10 I would say addresses. And those values are 1 to 10.

So even if these values were something different, it would have written 1:10 only. This 1:10 and this 1:10 has no meaning. They are not the same. This 1 :10 is saying the address starts from 1 and goes up to 10. There are 10 addresses, 10 locations, 10 cells where some value is given, and then those values are 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10. Similarly, if I run 'b' is equal to 'c' something something something. Now, this is what we have to understand, I think carefully.

(Refer Slide Time: 22:44)

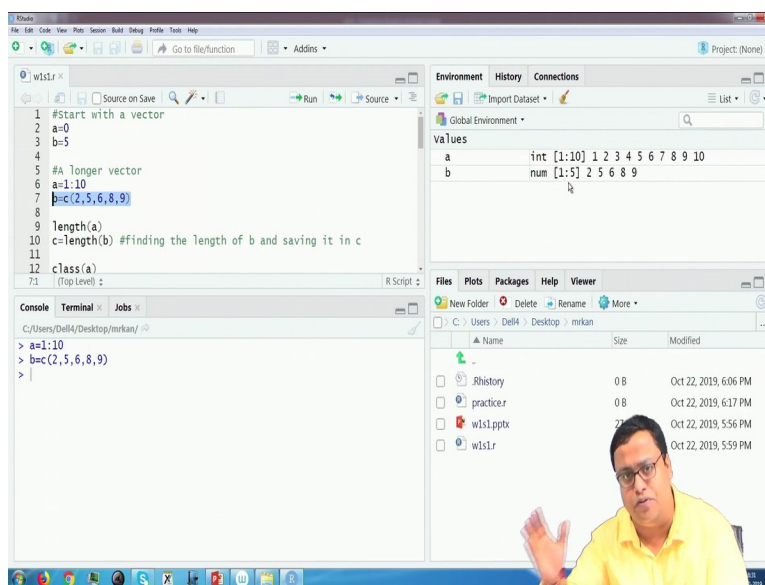


So there are two types of things we will see in R. one is something written, some xyz is written. Let us say I have written xyz, some name, I do not know. And then a first bracket. The moment I write this you will see something like this, where something is written and a first bracket, a first bracket generally signifies a function. What is a job of a function? It gives certain input and it works something and gives an output. So that is a job of a function. On the other hand, by chance if we have seen xyz and then a third bracket, it talks about a location, an address, a cell; mostly an address, not always a cell,

an address. So third bracket talks about an address. First bracket talks about a function which has some job.

Similarly, here if you see, I have written `b=c()`. So that means there is a function called `c`. what is its job? Its job is to give you a vector combination of whatever inputs you give. So here the inputs are 2, 5, 6, 8, 9. These five numbers are the inputs. It converts; this `c` function, converts the inputs to a vector form.

(Refer Slide Time: 24:05)



The screenshot shows the RStudio interface. The script editor contains the following R code:

```
1 #Start with a vector
2 a=0
3 b=5
4
5 #A longer vector
6 a=1:10
7 b=c(2,5,6,8,9)
8
9 length(a)
10 c=length(b) #finding the length of b and saving it in c
11
12 class(a)
```

The Environment pane shows the following values:

Variable	Value
a	int [1:10] 1 2 3 4 5 6 7 8 9 10
b	num [1:5] 2 5 6 8 9

The Console shows the following output:

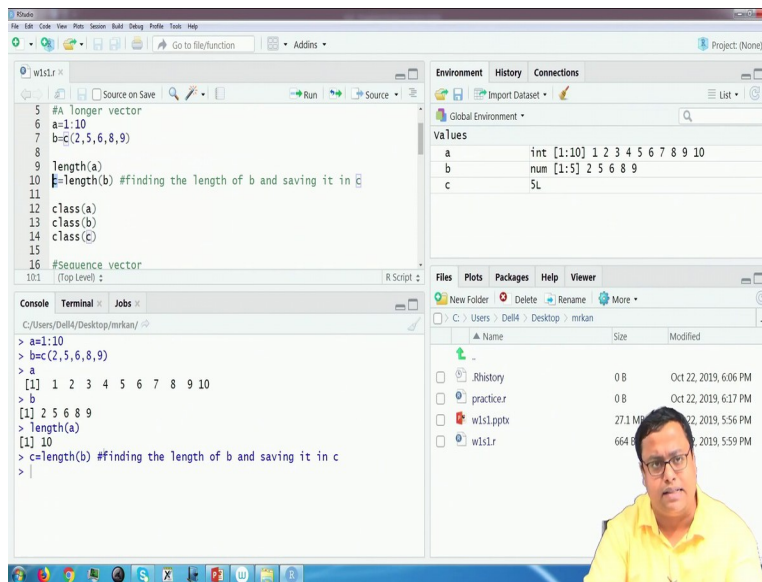
```
> a=1:10
> b=c(2,5,6,8,9)
>
```

A man in a yellow shirt is visible in the bottom right corner of the screenshot, gesturing with his hand.

So if I just run this line see here it is written 1 to 5 because there are 1 to 5 cells from first cell to fifth cell. And the contents are these. The difference between the first one and second one is these are integer and this is a numeric. The first one is integer because the moment I write a `1:10`, it knows I am asking from 1 to 10 only the integer values. So that is why it is putting 'int' there.

But when I am writing `b` is equal `c(2, 5, 6, 8, 9)`, that R does not know whether I will write in the next time 9, 9.5, 10.3, 11.6 or whatever. It does not know whether the next entry of this particular series will be non integers or not. That is why it is playing safe, it is putting numeric there.

(Refer Slide Time: 25:04)



The screenshot shows the RStudio interface. The editor window contains the following R code:

```
5 #A longer vector
6 a=1:10
7 b=c(2,5,6,8,9)
8
9 length(a)
10 c=length(b) #finding the length of b and saving it in c
11
12 class(a)
13 class(b)
14 class(c)
15
16 #sequence_vector
101 (top Level)
```

The console window shows the following output:

```
> a=1:10
> b=c(2,5,6,8,9)
> a
[1] 1 2 3 4 5 6 7 8 9 10
> b
[1] 2 5 6 8 9
> length(a)
[1] 10
> c=length(b) #finding the length of b and saving it in c
>
```

The Environment pane on the right shows the following values:

Variable	Class	Value
a	int [1:10]	1 2 3 4 5 6 7 8 9 10
b	num [1:5]	2 5 6 8 9
c	5L	5

The Files pane on the right shows a directory listing for the Desktop folder:

Name	Size	Modified
..		
..history	0 B	Oct 22, 2019, 6:06 PM
practice.r	0 B	Oct 22, 2019, 6:17 PM
wis1.pptx	27.1 MB	Oct 22, 2019, 5:56 PM
wis1.r	664 B	Oct 22, 2019, 5:59 PM

So how to get them; how to print them? Here the allocation happens, nothing else. So how to print them? If I print 'a' and the press an enter, I get 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. If I press 'b' and press an enter, then I will get 2, 5, 6, 8, 9 and so on. Similarly, I will talk about the third thing in a few minutes.

So then the next thing is length of 'a', how to find out how long a particular vector is. Often to run a loop, we have to know at what point it is ending. So what is the length of 'a'. length is a function, and then if I just select it and press run, it will give the length of 'a' is 10 because 'a' has 10 inputs. Easy, nothing fancy till now. So I will save these.

`c=length(b)` is finding the length of b; see I have written; so I have told whatever written after a hash is a comment. That will have no meaning. Whatever written before the hash is not a comment, so that will run. So `c=length(b)`. so what happens? Length of b gives me 5 because 'b' has 5 entries. That 5 value gets saved in 'c'.

Now why I have written this, you have to identify that this 'c', the one that I highlighted and this 'c' is different. This c is a name of a particular vector, and this c is a function. R actually more or less understands that but oftentimes you have to be careful that what you are writing.

(Refer Slide Time: 26:46)

The screenshot shows the RStudio interface. The script editor contains the following code:

```
8 length(a)
9
10 c=length(b) #finding the length of b and saving it in c
11
12 class(a)
13 class(b)
14 class(c)
15
16 #sequence vector
17 a=seq(1,30,by=2)
18 help(seq)
19
```

The console shows the execution of these commands:

```
> b=c(2,5,6,8,9)
> a
[1] 1 2 3 4 5 6 7 8 9 10
> b
[1] 2 5 6 8 9
> length(a)
[1] 10
> c=length(b) #finding the length of b and saving it in c
> class(a)
[1] "integer"
> class(b)
[1] "numeric"
> class(c)
[1] "integer"
>
```

The Environment pane shows the following values:

Variable	Class	Value
a	int [1:10]	1 2 3 4 5 6 7 8 9 10
b	num [1:5]	2 5 6 8 9
c	5L	

The Files pane shows a directory structure with files like 'Rhistory', 'practice.r', 'wisl.pptx', and 'wisl.r'.

Class is another function which gives you what class it is, means what type of object it is. So class of 'a' is an integer. Class of 'b' is a numeric and Class of 'c' is an integer again because 'c' is also an integer. So at any point of time you can write class to find out.

(Refer Slide Time: 27:04)

The screenshot shows the RStudio interface. The script editor contains the following code:

```
12 class(a)
13 class(b)
14 class(c)
15
16 #Sequence vector
17 a=seq(1,30,by=2)
18 help(seq)
19
20 #Repeat vector
21 b=rep(2,20)
22 ?rep
23
```

The console shows the execution of these commands:

```
> help(seq)
>
```

The Environment pane shows the following values:

Variable	Class	Value
a	int [1:10]	1 2 3 4 5 6 7 8 9 10
b	num [1:5]	2 5 6 8 9
c	5L	

The Files pane shows a directory structure with files like 'Rhistory', 'practice.r', 'wisl.pptx', and 'wisl.r'.

The Help pane shows the documentation for the `seq` function:

Sequence Generation

Description

Generate regular sequences. `seq` is a standard default method. `seq.int` is a primitive which can be used with a few restrictions. `seq_along` and `seq_len` are very common cases.

Usage

```

12 class(a)
13 class(b)
14 class(c)
15
16 #Sequence vector
17 a=seq(1,30,by=2)
18 help(seq)
19
20 #Repeat vector
21 b=rep(2,20)
22 ?rep
23
201 (Top Level)

```

Environment History Connections

Global Environment

Values

a	int [1:10]	1 2 3 4 5 6 7 8 9 10
b	num [1:5]	2 5 6 8 9
c		5L

Files Plots Packages Help Viewer

R: Sequence Generation Find in Topic

Usage

```

seq(...)

## Default S3 method:
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)),
    length.out = NULL, along.with = NULL, ...)

seq.int(from, to, by, length.out, along.with, ...)

seq_along(along.with)
seq_len(length.out)

Arguments

```

```

12 class(a)
13 class(b)
14 class(c)
15
16 #Sequence vector
17 a=seq(1,30,by=2)
18 help(seq)
19
20 #Repeat vector
21 b=rep(2,20)
22 ?rep
23
171 (Top Level)

```

Environment History Connections

Global Environment

Values

a	num [1:15]	1 3 5 7 9 11 13 15 17 19 ...
b	num [1:5]	2 5 6 8 9
c		5L

Files Plots Packages Help Viewer

R: Sequence Generation Find in Topic

The methods [seq.Date](#) and [seq.POSIXt](#).

[seq](#), [seq.Date](#) and [seq.POSIXt](#).

[seq](#), [seq.Date](#) and [seq.POSIXt](#).

Examples

```

seq(0, 1, length.out = 11)
seq(stats::rnorm(20)) # effectively
seq(1, 9, by = 2) # matches 'end'
seq(1, 9, by = pi) # stays below 'end'
seq(1, 6, by = 3)
seq(1.575, 5.125, by = 0.05)
seq(17) # same as 1:17, or even

```

Now I will show you another interesting thing. Let us say; I wanted you to find out something called a sequence and sequence starts from 1, ends at 30 with a jump of 2. So 1, 3, 5, 7, 9, 11 and so on till it reaches 30. Now I know that this particular function is called sequence function 'seq' function. So probably let us say, if you know that function's name is 'seq', for example previously I knew class and length.

So if I know the name of the function, I can ask the help like this help(seq) to know that what are the various aspects. And on the right side you will see how the help documentation is coming. So you can read it a little bit. So that is how I am showing in

the fourth quadrant, one tab I am showing, which is called the help tab. There are other tabs. We will come to that when it is required.

So in the help tab, there is a description; there is usage. So in the usage, you will see that here it is written help from is equal to this, to is equal to this, by is equal to something. And as you go down it says from and to is the starting and end values of the sequence, by is the number increment of the sequence, and if you further come down, you have to read it carefully. And if you further come down, the usages are also given. There is one usage called sequence 1, 9, by is equal to 2. So I can probably copy this and paste it here and try. What is it giving? It is giving 1,3,5,7,9. That means it starts at 1, ends at 9, each jump is 2. That is why if I save this, if I run this now and then try to print A, I got 1,3,5,7...because it starts at 1, ends at 30 by 2.

Now you can do this thing as long as you know the function's name is 'seq'. That is how I found out. I run the help, and it gave me all the helps. If you do not know the function; in real life situation you will not know. There are lots of functions. There are probably billions; if not millions, lakhs of functions at least. For a single human being, it is almost impossible to remember all the functions, their syntax and its use. So, you do not have to remember the names. So what to do? We will actually try to see in the next video.

Thank you. We will continue from this particular line only from the next video. Thank you for being in my class. I hope you have a wonderful learning journey. Thank you.