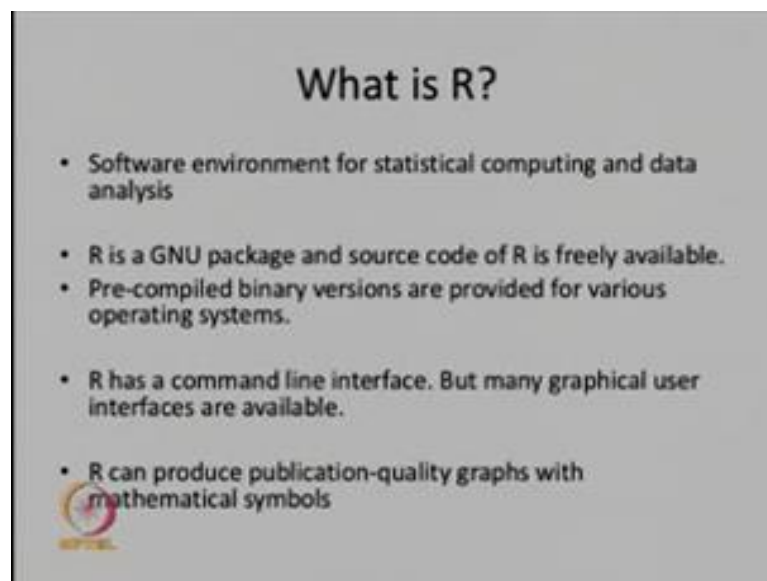


**Introduction to Biostatistics**  
**Prof. Shamik Sen**  
**Department of Bioscience and Bioengineering**  
**Indian Institute of Technology, Bombay**

**Lecture - 09**  
**R Programming**

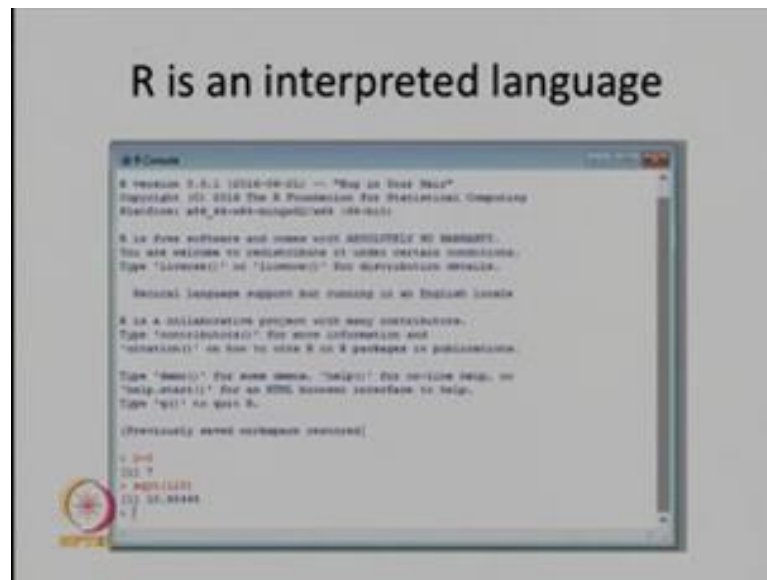
Hello and welcome to today's lecture on biostatistics. We will start with a brief recap of the of R language and then solve few examples in RStudio.

(Refer Slide Time: 00:31)

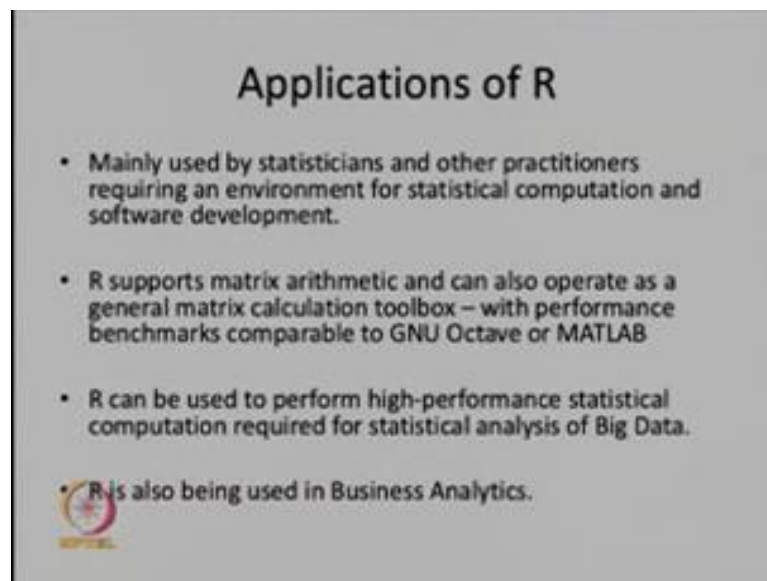


Just to revisit our discussion of what is R; it is a software environment for statistical computing and data analysis. So, it is an open source package, it is freely available, it has a command line interface, but since for many users using a command line interface may not be beneficial or easy to handle there are you know open source software switches which provide a graphical user interface. So, that it is widely used to and the best part of R is it can produce publication quality graphs with mathematical symbols.

(Refer Slide Time: 01:00)



(Refer Slide Time: 01:10)




So, it is an interpreted language this is just the you know a print screen of the R console you see it is just a you know; you can just enter things here, but and you have an r has been widely used by statistics it was actually developed at the University of Auckland and it is now widely used it has a support base where people contribute to its further development and it has met the performance benchmarks comparable to that of GNU Octave or MATLAB. So, it is that is you know reason why it is used for statistical analysis of big data as well as in business analytics.

(Refer Slide Time: 01:36)

## Getting R - 1

- R is an open source programming language. Due to its popularity pre-compiled R binaries are also available for different platforms.
- Binaries for windows, Unix or MacOS can be downloaded from R project website <https://www.r-project.org>.
- These binaries can directly be used to install the R programming of a computer.




So, you can download R from this `r project dot o r g`, `R projectors o r g` and you can; depending on the software interface whether windows UNIX or MacOS, you can download the appropriate file and install it.

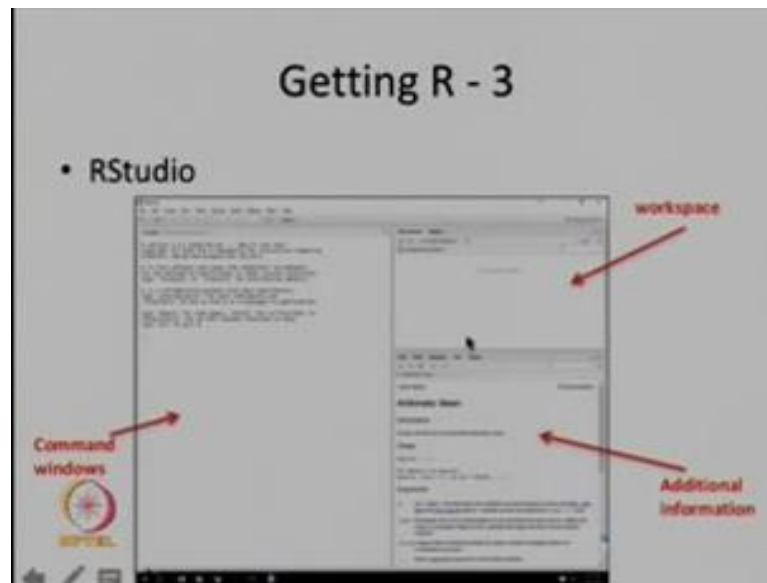
(Refer Slide Time: 01:51)

## Getting R - 2

- However, R is command line so may not be suitable for learners.
- For this, many graphical user interfaces (GUIs) software are available for R.
- These GUIs-based software provide an user friendly interface to write, correct and run R code.
- Rstudio is one such widely used GUI interface for R.

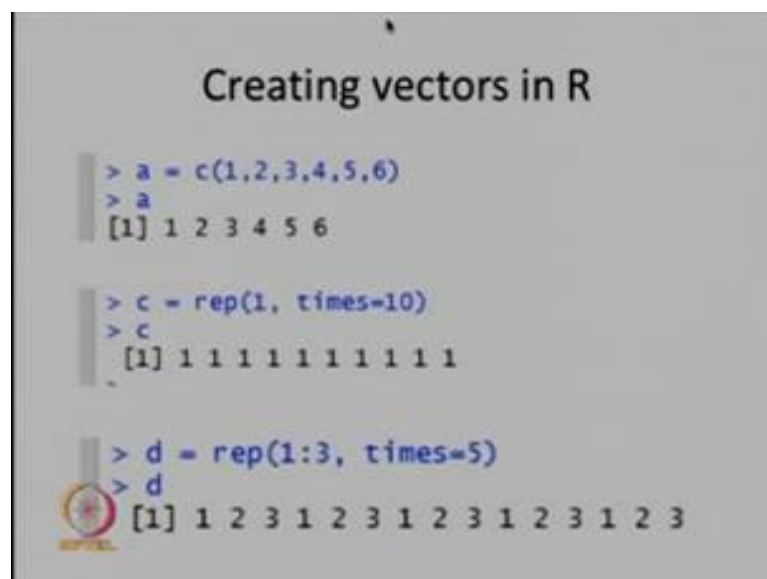


(Refer Slide Time: 02:01)



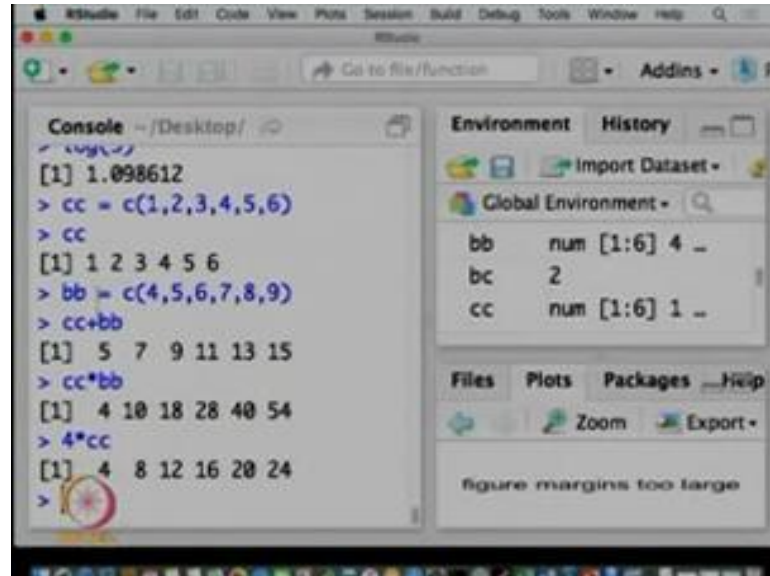
And so because it is a command line interface so people have developed software based on R which have a GUI interface and RStudio is one such them on such of them and in R, this is just a print screen of how RStudio looks, this is the command window where we type in numbers or whatever we need to do, this is the workspace which stores all the data that we generate and here you have you have to provided additional information on how to use it or what something means.

(Refer Slide Time: 02:21)



So, this is just an example of how I can create a vector in R and. So, let us directly go to RStudio and see rerun all these examples so that your idea is in christened.

(Refer Slide Time: 02:34)



```

> [1] 1.098612
> cc = c(1,2,3,4,5,6)
> cc
[1] 1 2 3 4 5 6
> bb = c(4,5,6,7,8,9)
> cc+bb
[1] 5 7 9 11 13 15
> cc*bb
[1] 4 10 18 28 40 54
> 4*cc
[1] 4 8 12 16 20 24
>

```

Variable	Class	Value
bb	num [1:6]	4
bc		2
cc	num [1:6]	1

So, this was an example I had done earlier, but let me see. So, in order to generate a vector so, you can do all basic computation I can write a equal to 1, b b equal to 2 I can write a power b b. So, I can get these answers b b power b b exponential of b b so on and so forth. And what you see is if I drag this down all these numbers that we are putting in are we getting stored here in terms of value it has stored the value of a a stored the value of b b; b c whatever. So, you can do the basic arithmetic you can have. So, sin of 30 as I pointed out yesterday you have to write by pi by 180. So, in order to come you know convert it into radians and only then will you get the value.

So, this is how you get 0.5 if you do sin of 30, you will get some other value. So, remember that we have to convert every degree into a radian in order to use the sin cos tan functions you can do a sin or sin inverse let us say we can do a sin of half and you get the value of 0.5 to 3 which is nothing but this particular value. So, arc sin arc chords you can do log; log 10 of 10 is 1, you can also do log of 10 comma 10 which is also going to give you the value. So, you can put log 10 base 2 you can do just log of 3 let us say it is the natural log and you will get slightly value because e is 2.7 something.

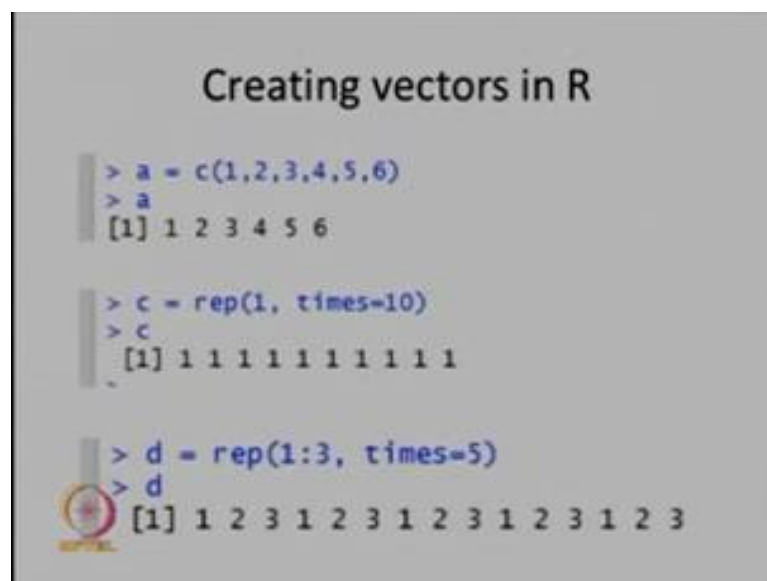
So, these are all the basic calculations in order to generate a victor let us say. So, I can write c c is equal to c 1 comma 2 comma 3 comma. So, c and. So, the as you see the

format the syntax is you get the variable name which is a vector and c is for concatenate and you give them item one item 2 separated by commas. So, if I type c c and put enter, it will give you the exact value. So, this is how you create vectors, you can do elementary vector operations like let us say you can do. So, let us say c c is this.

Let us say you define b b as c of you can do c c and b b as another vector I can do c c plus b b and I get the answer I can do c c star b b. So, you see that these are element wise operations. So, that is why when I am doing c c star b b 1 is multiplied by 4 so on and so forth. So, if I want to multiply something throughout then I should do let us say 4 start c c then all the individual entries are multiplied by this factor.

So, for a scalar multiplication you put a pre factor outside the vector and for these are all vector additions vector multiplications what you see is these operate at the at the single element level.

(Refer Slide Time: 05:45)



```
> a = c(1,2,3,4,5,6)
> a
[1] 1 2 3 4 5 6

> c = rep(1, times=10)
> c
[1] 1 1 1 1 1 1 1 1 1 1

> d = rep(1:3, times=5)
> d
[1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```


So, if I go back to this power point. So, this you can even create at same number repeated whatever number of times. So, the syntax is very you know understandable you repeat whatever is the number is 1 or a or b and times how many times you want to repeat and you see you create a vector of the same number which is 10 times you can repeat a sequence. So, let us say you have 1 to 3 and so 1 colon 3 means between 1 to 3 and this has to be repeated 5 times and you can accordingly get this particular you know vector.

(Refer Slide Time: 06:18)

### Creating vectors in R

```
> b = seq(from=1,to=7,by=1)
> b
[1] 1 2 3 4 5 6 7

> e = rep(seq(from=1,to=7,by=1), times=5)
> e
[1] 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7 1 2 3 4 5 6 7
```



You can also create a sequence. So, this is another example of a sequence where you go from 1 to 7 in steps of 1 and you will get this particular t, you can create a repeat of a sequence. So, every time you put a bracket and you put an operator, it operates on this whole thing and that is how you have a repetition of a sequence which is number of time 5 times. So, f 7, 7, 7, 7, 7, so on and so forth.


(Refer Slide Time: 06:43)

### Basic operations on vectors - 1

```
> a = c(1,2,3,4,5,6)
> a
[1] 1 2 3 4 5 6

> b = a+1
> b
[1] 2 3 4 5 6 7

> d = a*5
> d
[1] 5 10 15 20 25 30
```



(Refer Slide Time: 06:48)

```
Basic operations on vectors - 1

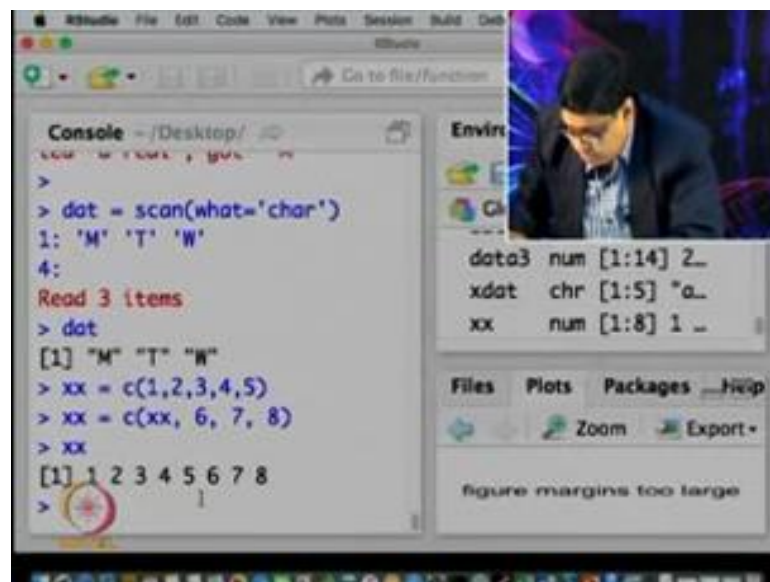
> g = a*b
> g
[1] 2 6 12 20 30 42

> h = a^2
> h
[1] 1 4 9 16 25 36

> i = exp(a)
> i
[1] 2.718282 7.389056 20.085537 54
```

So, this is just what I did which was element wise operations on a vector you can; you know do all the calculations here let us go back to RStudio if I go back to RStudio.

(Refer Slide Time: 07:06)



Now, let us say I have my vectors and what I want to add a vector online as opposed to typing them inside like this if the vector is long then it is difficult to enter. So, what I can do is I can add the vector online. So, in case I can write that is equal to scan till open brackets. So, if I do enter then it gives the it puts the command prompt here which means I can enter anything I put spaces if I put enter it again gives me the option of adding you



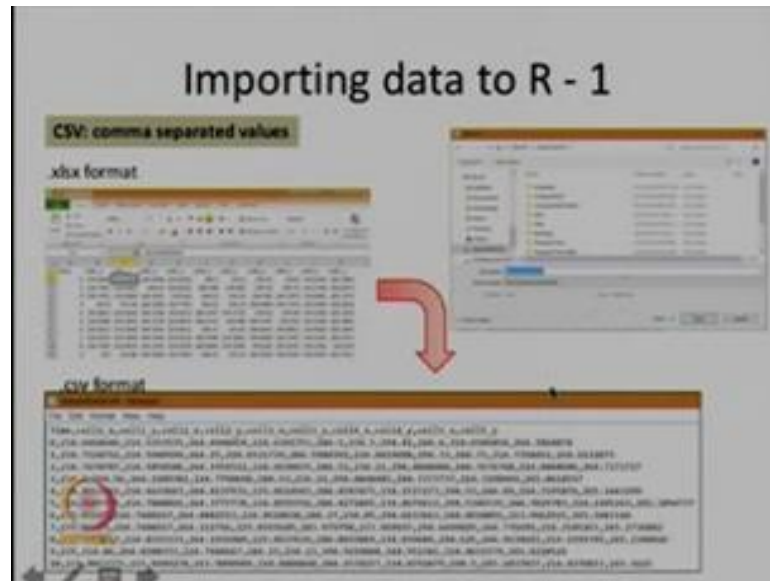
know many more numbers. So, you see the number 13 is written here because I have already made 12 entries. So, this is the 13th entry.

So, it begins with one which is this entry. So, I can keep on putting random numbers and every time I put enter it gives me the option of putting in any entries, but if I press one more entry then essentially it will understand that that is the end of it and it has created this particular vector which has 17 items only. So, I can now type that and see what is the value of and you see that this is this is the way it is. So, you see that depending on how you know you write it and how you know what is your font size? So, when it is coming you can only reach till 13th entry here that is why from the 14th entry it is coming here. So, in order to know the length of the vector I can use this length of that it tells me that there are 17 column entries in this particular vector.

I can also. So, the jargon for entering a character entry is slightly different. So, let us say I can do this extract that is equal to scan if I do this if I want to enter characters then what I have to do is let us say Monday, Tuesday, Wednesday. So, I put 2 times that. So, in this case I have to let us see I think I have to write that is equal to scan what equal to char. So now, by default this function scan always expects to get real numbers so that is why you could see expected a real got m which is a character. So, if I put this statement inside that scan and what it is expecting is a character then there is no problem. So, if I now write that here you will see Monday, Tuesday, Wednesday, as the numbers being entered.

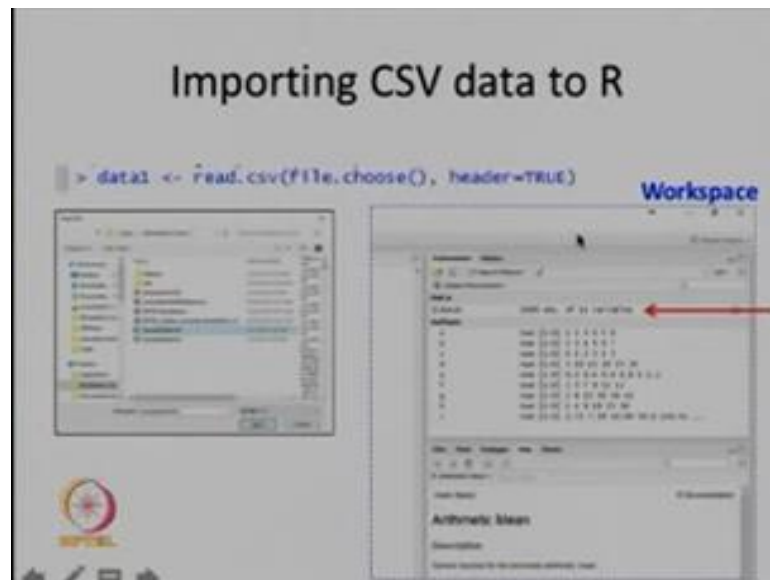
So, I can add, let us say if I have  $x \times x$  equal to  $c$  of I can write I can add elements to  $x \times x$  by writing  $c$  of  $x \times x$  comma 6, if I write  $x \times x$ , now you see it has added. So, it is possible to add numbers into a particular vector and as before as I had shown you before, we can you know even puts to the order of the  $x \times x$  does not matter I could have well written  $c$  of 6, 7, 8 comma  $x \times x$  in that the vector range would have been changed. So, if I go back to the presentation. So, it the next things of course, these are still numbers which I can enter and I can keep on entering on screen.

(Refer Slide Time: 11:09)

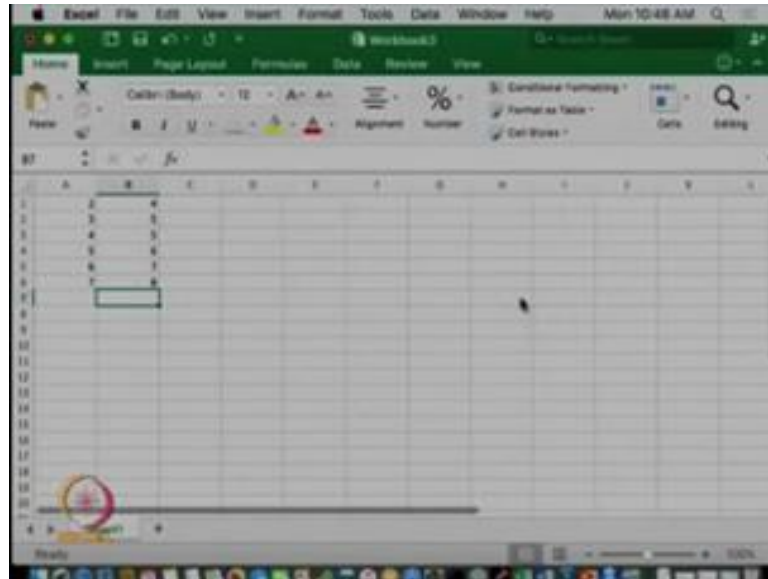


But if you have a big file then this function does not work you have to use what is called you know you can import data using various ways.

(Refer Slide Time: 11:17)

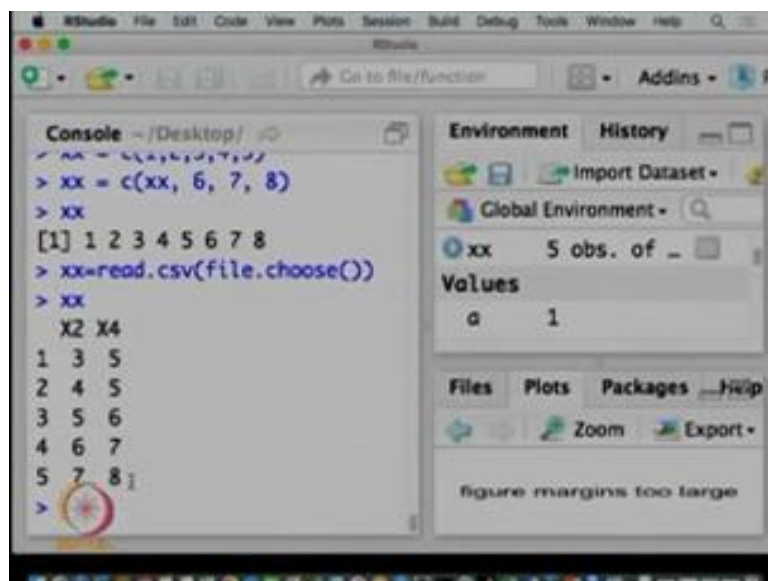


(Refer Slide Time: 11:30)



So, this is an example where you can import data using CSV format, this is called comma separated variable. So, let us do this particular example what I will do is I will create and I will open an excel sheet and let us. So, I have entered some numbers and I can save it in CSV format. So, when you do save as by default it is always in excel workbook, but what you can go down and you can choose comma separated values this is CSV and you can do the save it give it prompts you this particular warning, but you can say continue and this you know this is stored as a CSV file.

(Refer Slide Time: 12:16)



Now, what I can do is I can do this I can go to RStudio and I can write x x is equal to read dot CSV and I can write file dot choose what this means. So, when you do this it allows you it allows you to choose a particular file. So, if I do enter it will give it will prompts me this and this is the latest workbook that we have had this is a CSV file, I can open it and it gets chosen you can select it if I write x x here and you see that these are the values which it chose.

So, even though we did not explicitly enter the tags of what are the column names in CSV these things are already chosen and even the left row numbers they were stored in the CSV format generated. So, if I go back to the CSV file. So, you can you know this is the easiest way of this is just another way of example of how you can choose this particular values. So, this is the most widely used system to import data, but particularly when you have big data.

(Refer Slide Time: 13:28)

**Calculating descriptive statistics in R-1**

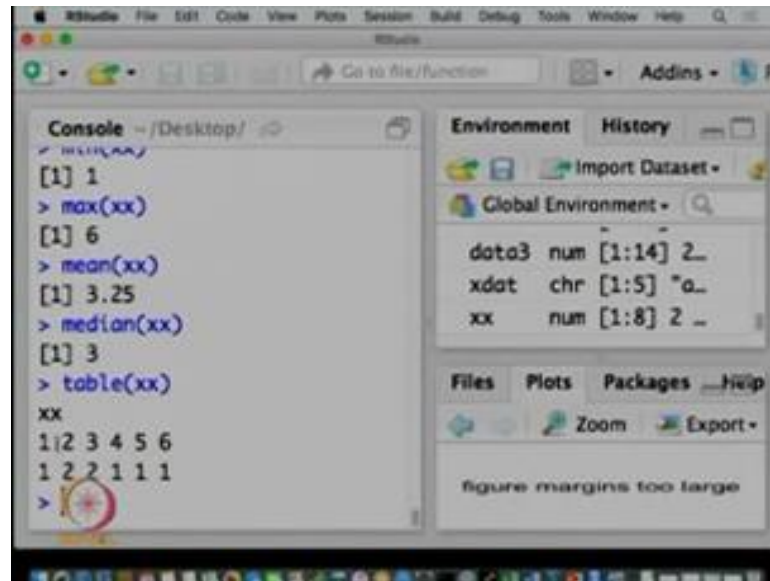
**Finding frequency in categorical data**

```
> celltype = c("stemCells", "differentiatedCells",  
> table(celltype)  
celltype  
differentiatedCells      stemCells  
                2                6
```

**Mean, median, min & max**

The slide includes a logo in the bottom left corner and navigation icons at the bottom.

(Refer Slide Time: 13:39)



```
Console ~/Desktop/
> [1] 1
> max(xx)
[1] 6
> mean(xx)
[1] 3.25
> median(xx)
[1] 3
> table(xx)
xx
1 2 3 4 5 6
1 2 2 1 1 1
>
```

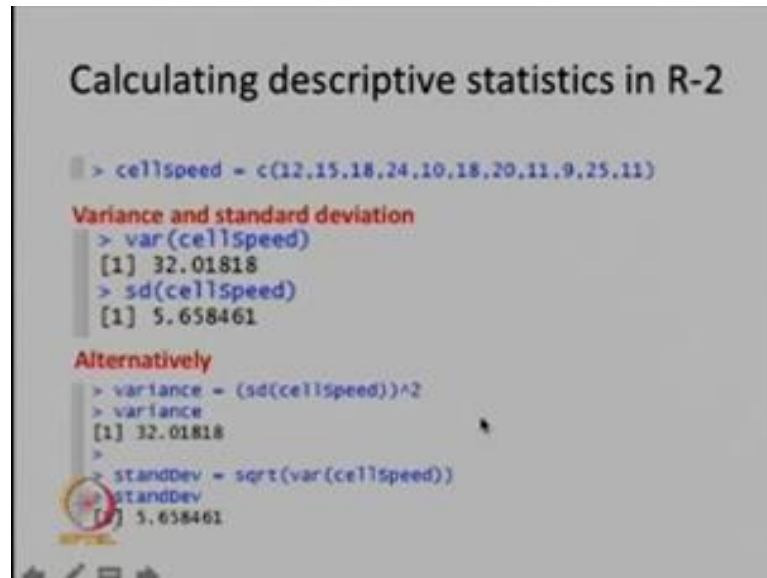
The screenshot shows the RStudio interface. The console on the left displays the following R commands and their outputs: `[1] 1`, `> max(xx)` returns `[1] 6`, `> mean(xx)` returns `[1] 3.25`, `> median(xx)` returns `[1] 3`, and `> table(xx)` returns a frequency table for the vector `xx` with values 1, 2, 3, 4, 5, 6 and their respective counts 1, 2, 2, 1, 1, 1. The Environment pane on the right shows the global environment with variables `data3` (numeric, length 14), `xdat` (character, length 5), and `xx` (numeric, length 8).

Now, let us get down to some examples of finding frequency mean medians one and. So, forth if I go back to RStudio. So, let us say I will again entered another set of parameters let us say x x, x x, in the screen is equal to c of I have entered this random array I can know the length of the vector by writing length of x x, it has 8 entries, I can find minimum of x x which is one max of x x which is six I can have mean of x x which is 3.25 I can have median of x x it gives me a value of 3.

So, these are widely useful you know ways of doing of getting the statistics descriptive statistics from your vector and 1 more thing I wanted to show is if you do table of x x then you will get the distribution. So, how many values have value of one you have only one entry of one you have it says that there are 2 entries of 2 let us see. So, these are the numbers I have entered there are 2 twos and that is why when I do the frequency count there are 2 2s here similarly 2 3 is 1, 1, 1.

So, this is a very useful way of getting the; you know the statistics from these particular examples I have shown you how to calculate the mean median minimum and maximum.

(Refer Slide Time: 15:10)



```
Calculating descriptive statistics in R-2

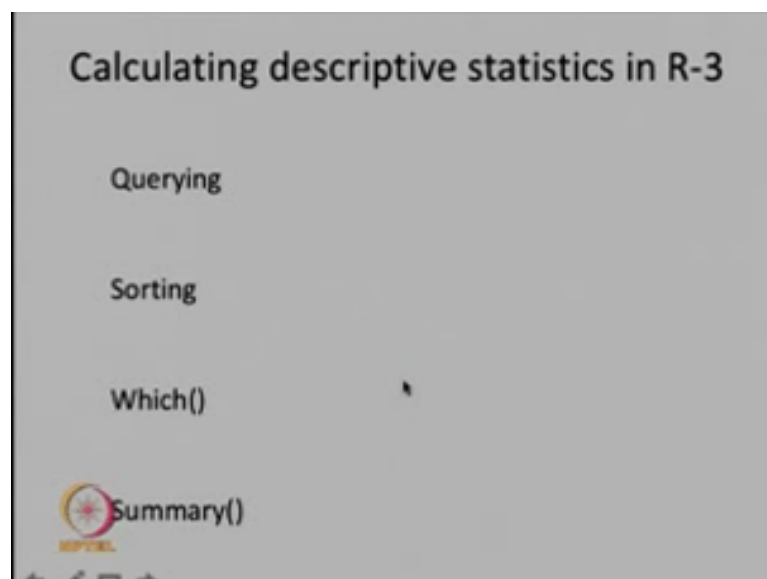
> cellspeed = c(12,15,18,24,10,18,20,11,9,25,11)

Variance and standard deviation
> var(cellspeed)
[1] 32.01818
> sd(cellspeed)
[1] 5.658461

Alternatively
> variance = (sd(cellspeed))^2
> variance
[1] 32.01818
>
> standDev = sqrt(var(cellspeed))
standDev
[1] 5.658461
```

You can also do the variance and standard deviation as. So, you can use this particular function of where to find out the variance of this distribution SD to find the standard deviation of this particular vector. So, in this particular example where you have how many entries 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 entries your variance is coming out to be a value of 32 and standard deviation is around 5.6. So, you can also find out the standard deviation and square it to find out the value of variance which will give you the same information or you can calculate the square root of the variance to find out the standard deviation.

(Refer Slide Time: 15:48)



```
Calculating descriptive statistics in R-3

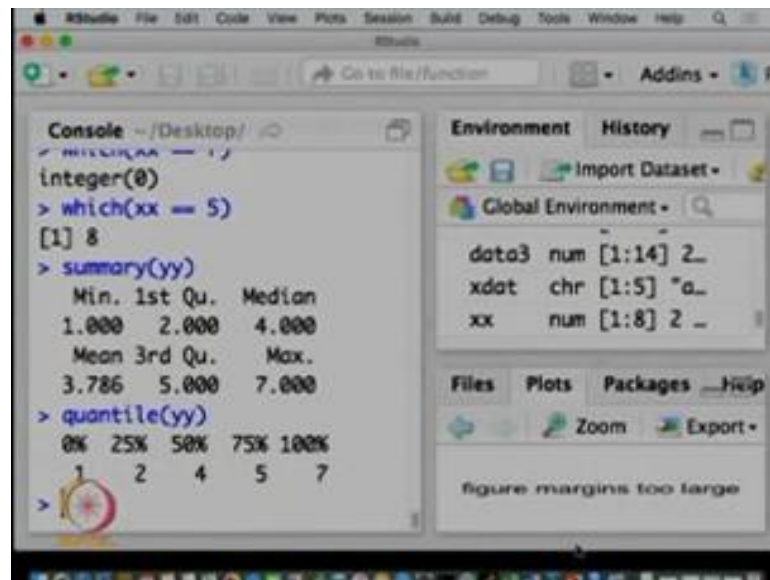
Querying

Sorting

Which()

Summary()
```

(Refer Slide Time: 15:27)



```
integer(0)
> which(xx == 5)
[1] 8
> summary(yy)
  Min. 1st Qu.  Median 
1.000  2.000  4.000 
  Mean 3rd Qu.  Max. 
3.786  5.000  7.000 
> quantile(yy)
  0%  25%  50%  75% 100% 
  1   2   4   5   7
```

The screenshot shows the RStudio interface. The console window displays the following R code and its output:

```
integer(0)
> which(xx == 5)
[1] 8
> summary(yy)
  Min. 1st Qu.  Median 
1.000  2.000  4.000 
  Mean 3rd Qu.  Max. 
3.786  5.000  7.000 
> quantile(yy)
  0%  25%  50%  75% 100% 
  1   2   4   5   7
```

The Environment pane on the right shows the following objects:

Object	Class	Length	Value
data3	num	[1:14]	2...
xdat	chr	[1:5]	"a..."
xx	num	[1:8]	2...

Some more examples, so you can use; I will show you how you can use this particular functions again let us go back to RStudio. So, let us say I want to sort x x. So, let us write you know a longer value longer vector. So, y y has 14 entries length should give me a value of 14. So, let us I already know the information I can use these earlier values to find out this, but I what I can also do as you can do sort of y y and then you see that it is being sorted in ascending order if you wanted to sort the same you know same vector in reducing order the other way round. So, you can write short y y comma decreasing equal to true. So, in this case you have the reverse order from the topmost number to the lowest number you can also have let us say you want to some statistics how many there are how many numbers in this vector which are greater than 2 for example; so I can write x x. So, there is 3 6 3 4 5 with the 5 numbers which are greater than x x.

I can also write, I can ask at which position is x x equal to 5. It is returning me a value of 8. Let us see where I have the first value of 5 1 2 3 4 5 6 7, no 1, 2, 3, 4, 5. So, x x is equal to 5, it is returning via value of 8. I can do another thing which is the summary of y y. So, what it gives me are all the minimum the first quartile the median the third quartile and the maximum. So, instead of summary you can also use this function called quantile and this is the same thing, but given in terms of exact percentage again you see the minimum is one your first quartile is quantile is 2 as is showing up here your median which is the 50th percentile has a value of 4 you are 70. So, your 75th percentile is this and the maximum is 7. So, clearly, in this case you have median and mean both reported

you mean you write quantile you will only get the plot of these actual percentiles. So, these are the things that I wanted to you know these functions I wanted to these are useful.

Let us take another example. So, imagine you are doing a measurement where you are tracking or your aim is to correlate the cell. So, a cell which is moving and you want to see whether when it is moving it is elongated or it is round.

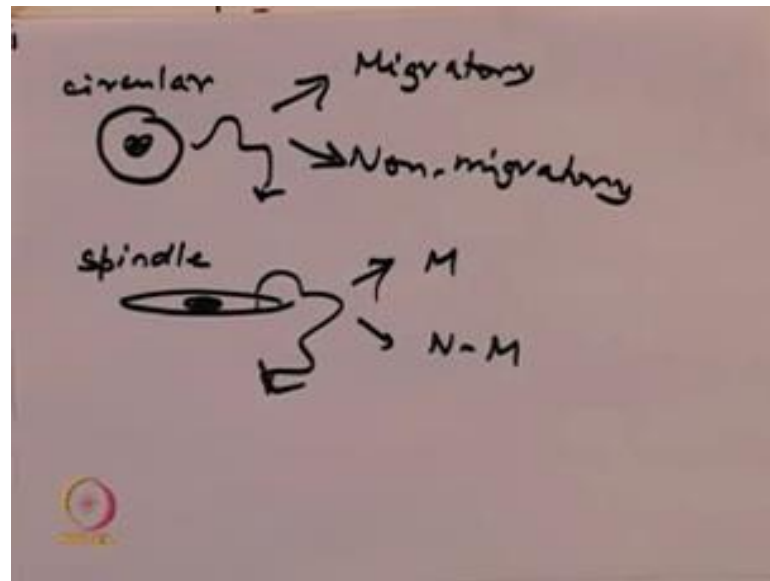
(Refer Slide Time: 19:17)

CellShape	CellPhenotype
Circular	non-migratory
spindle	migratory
spindle	non-migratory
spindle	migratory
Circular	non-migratory
Circular	non-migratory
Circular	non-migratory
spindle	migratory
spindle	migratory
Circular	non-migratory
Circular	non-migratory
Circular	migratory
Circular	non-migratory
spindle	migratory
Circular	migratory
spindle	migratory
Circular	non-migratory
spindle	migratory
spindle	non-migratory
spindle	migratory
spindle	non-migratory

And what you do is you do this particular you know this is the data where you have 2 matrix for characterance the cell phenotype one is whether it is circular or spindle.



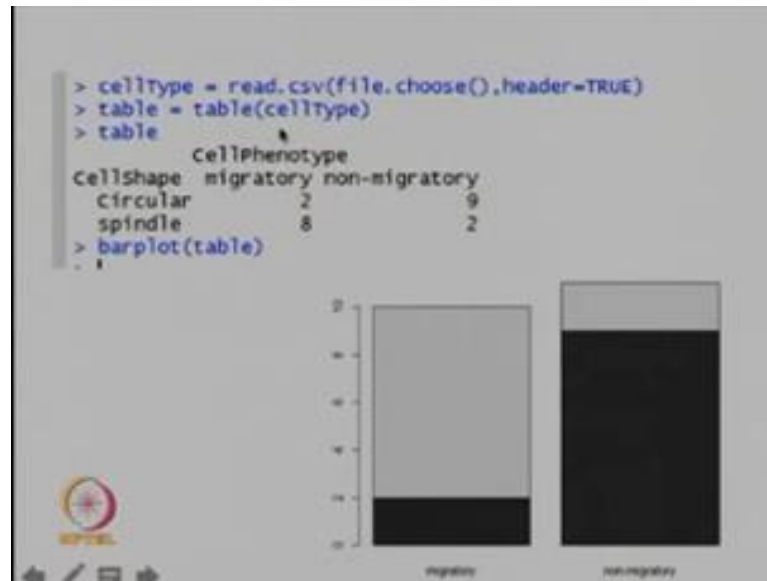
(Refer Slide Time: 19:33)



So, circular cell would look something like this; this is your circular cell and this is your mode spindle shaped. So, this is spindle and this is circular and you are. So, these are of course. So, you have a nucleus at the center and you want to see; what are the trajectories of these cells in 2 d plane and you want to find out? So, based on the distance, it is moving each of these have 2 population migratory and non migratory same here you have m and non migratory. So, this is just an example. So, in this particular example what we have is the distribution for let us say 22 such cells for each cell you have the shape which is either a circular or spindle and the phenotype which you characterize as non migratory or migratory.

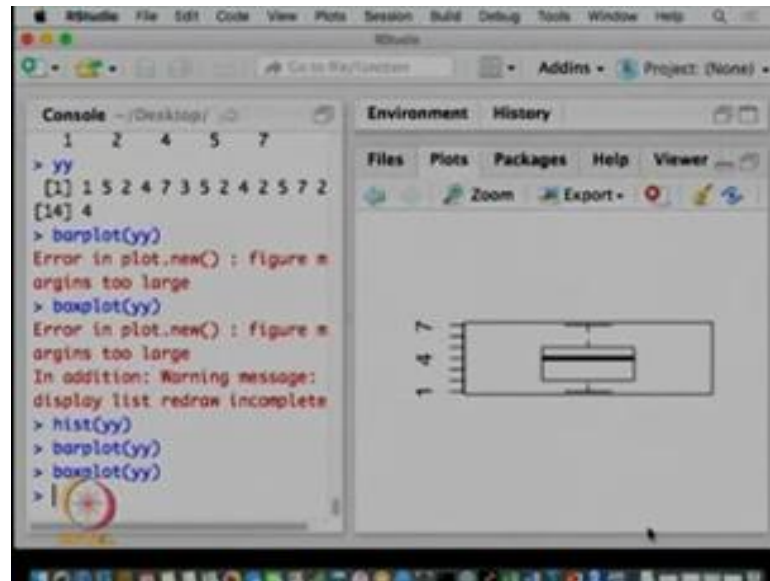
So, essentially both these metrics are categorical data and you want to find out. So, from this table you want to find out what is the distribution of this data.

(Refer Slide Time: 20:42)



So, I can have I can first generate the table of the cell type. So, this is just the way to you know import the data into the r framework you can do this command of table to get what are the different distributions what you see is they are circular in shape 2 of the such cells are migratory and there are 9 which are non migratory. And in case of spindle shapes 8 of them are migratory 2 of them are non migratory. So, this data kind of conveys the point that a greater proportions of cells which migrate are spindle in shape and a greater proportion of cells which are non migratory are circular in shape. So, this is a way of plotting this data you can do bar plot of this table and you have this particular distribution.

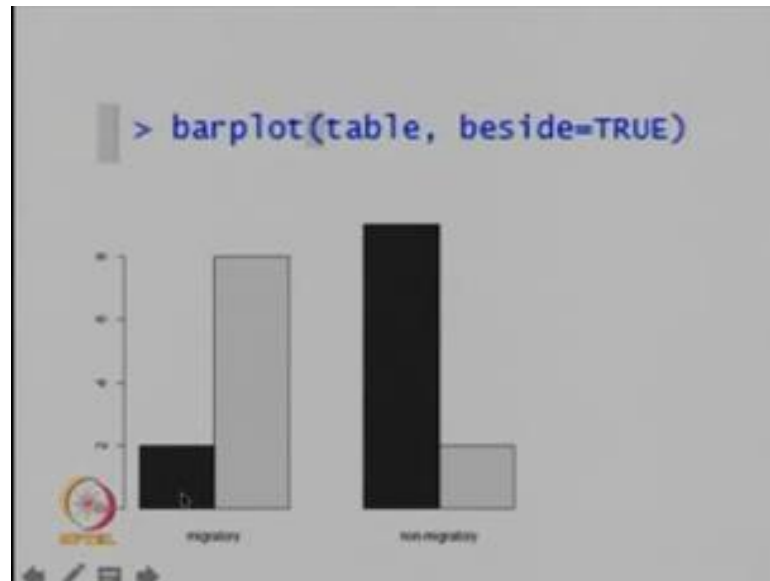
(Refer Slide Time: 21:35)



So, let us go back to R. So, let us use the same data which is y y, I can write let us say bar plot of y y and figure margins are too large. So, this is perhaps not coming here I do not understand, but you can also go histogram of y y, this problem is coming we will see, but if we if you use these particular functions box plot bar plot let us reduce the file size will now you see. So, it was coming like this because this required a certain amount of space you see histogram of y y will give you this particular distribution it conveys a message you have a peak here and another peak here and then one convert data here let us see again if I could you know plot the bar plot here.

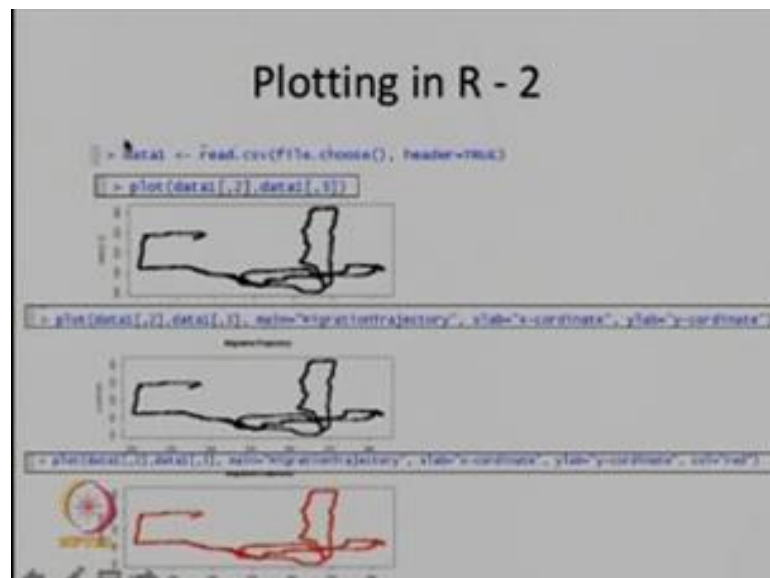
So, this is your same plot plotted in bar plot manner we can also do the same thing for box plot. So, this is the box plot with buds what we can also do is we can. So, if I go back to the presentation now. So, this is what we had plotted which is the bar plot of the table and you see that in case of these cells which are circular you have a greater portion which are non migratory which are circular in shape and lesser portion which are migratory, but circular in shape.

(Refer Slide Time: 23:32)



So, I can also represent this data by plotting this side by side. So, this would be how it would look like. So, if I write this particular framework which is beside equal to true then I would generate this particular plot. So, in terms of migratory and these are circular which is very few and you know and spindle shape which are very high. So, I can use a similar thing to generate the plotting for r let us say you have a trajectory of a cell which is moving in x y plane I can any how imported the data.

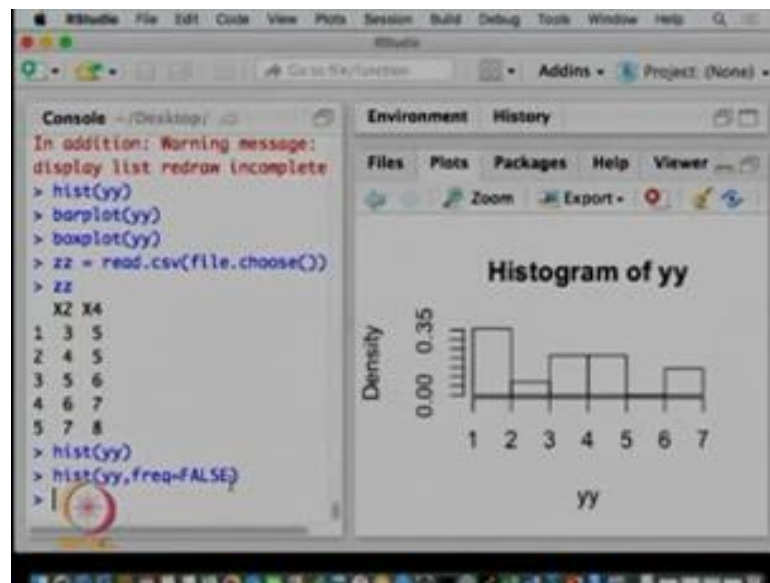
(Refer Slide Time: 24:02)



So, you can clearly see that by plotting the data as; so you have read this particular data and the data has become you know data is as represented as x and y and so, what you see here is you have first imported the data and in this case, we are plotting data colon comma 2 data colon comma 3.

Why are you choosing the second and the third column the reason is because when you import your you know you have a row or a column which gets inserted which is the row number.

(Refer Slide Time: 24:39)



So, let us go back to R. So, which is the vector that we had you know let us again have z z is equal to read dot CSV file dot choose. So, if I again read my workbook 3 and I plot z z. So, you see that there are 3 columns which are generated this one was the default entry in excel which we do not have no control over. So, this is why you have to neglect this particular column and you plot x 2 and x 4 and this is what we have done in this power point file this is what we have done in this power point file whereby we have plotted data colon comma 2 colon comma 3, this is high and then this is the corresponding you know plot in x y plane, I can also what I can do is I can use I can give a title to this particular plot which is migration trajectory which appears here I can set my limits. So, x lab, I can set my labels. So, x lab is x coordinate y lab is y coordinate you can enter here, you can what you can also do is you can change with the play with the color of this particular

trajectory what you see here is color equal to red means essentially we are resetting the trajectory color to red.

And the last is you can also put `x lim`; `x lim` is for limits. So, if you want to you know probe it within a certain domain you want to plot it within a certain range you can have control over `x range` and `y range` with that I think you have gotten a good enough handle of how to do this. So, we would stop here I would just go back to you know let me go back to this particular slide let me just go back to RStudio once more. So, you can have these particular plots that is; we had plotted histogram of `y y` you can plot it in histogram version.

So, if you write `histogram of y y comma frequency equal to false`, what it does? It actually converts this data as density or relative frequency and you instead of absolute values, you will get a range. So, here itself for example, you know; in terms of `y y`, I can write other values I can change the color of this and so on and so forth.

With that you will I hope you have; you have been convinced of the power of this R software which is open source. You can freely download it, install it and use it for your own purposes particularly when you are handling you know you should get into the habit of calculating standard deviation and mean and all these descriptive strategies even of generating these plots in R.

With that I thank you for your attention and we will meet again in next class.