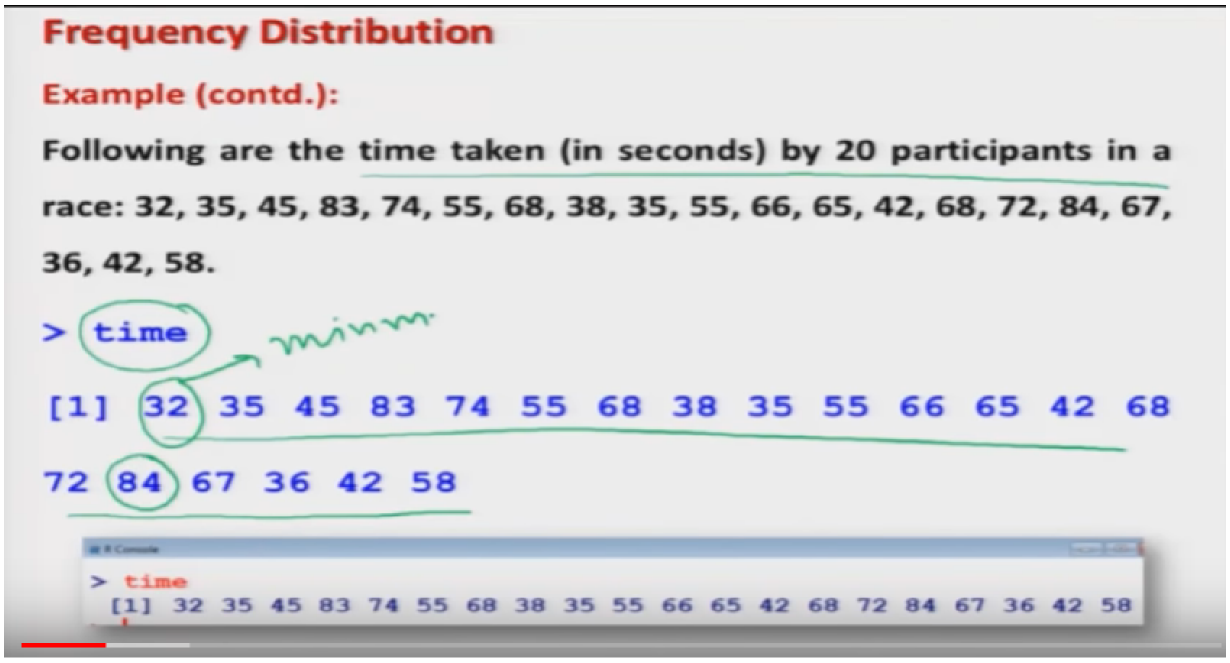
Lecture 09
Frequency Distribution and Cumulative Distribution Function

Welcome to the, next lecture on the course, Descriptive Statistics with R software. You may recall that in the earlier lecture, we started a discussion on the frequency distribution. We had understood the concept of absolute frequencies, relative frequencies and we had put them in our table, what we call as, 'Frequency distribution'. And once we are trying to construct the frequency distribution, we have different types of variable, we discrete variable, continuous variable. So, we had completed our discussion, on the discrete variable and we started the discussion, on how to, to construct the frequency distribution or the frequency tables, based on a relative frequency from a continuous data. So, in the earlier lecture, we had taken an example and we had seen, that how manually, you will try to create the groups and based on that, you will try to compute the frequencies and based on that you will try, to construct the frequency distribution or frequency table. So, we will continue, on the same lines and we would like to see today, that whatever we have done manually, manually means we had made all the calculation by hand, now we would like to implement it on the R software, using the same steps same concept, same methodologies and let us try to see how the things are going to work and essentially how you are going to obtain a frequency distribution table, using the R software in case of continuous data. Okay? So, let us start our discussion,

Refer slide time :(01:57)



Frequency Distribution

Example (contd.):

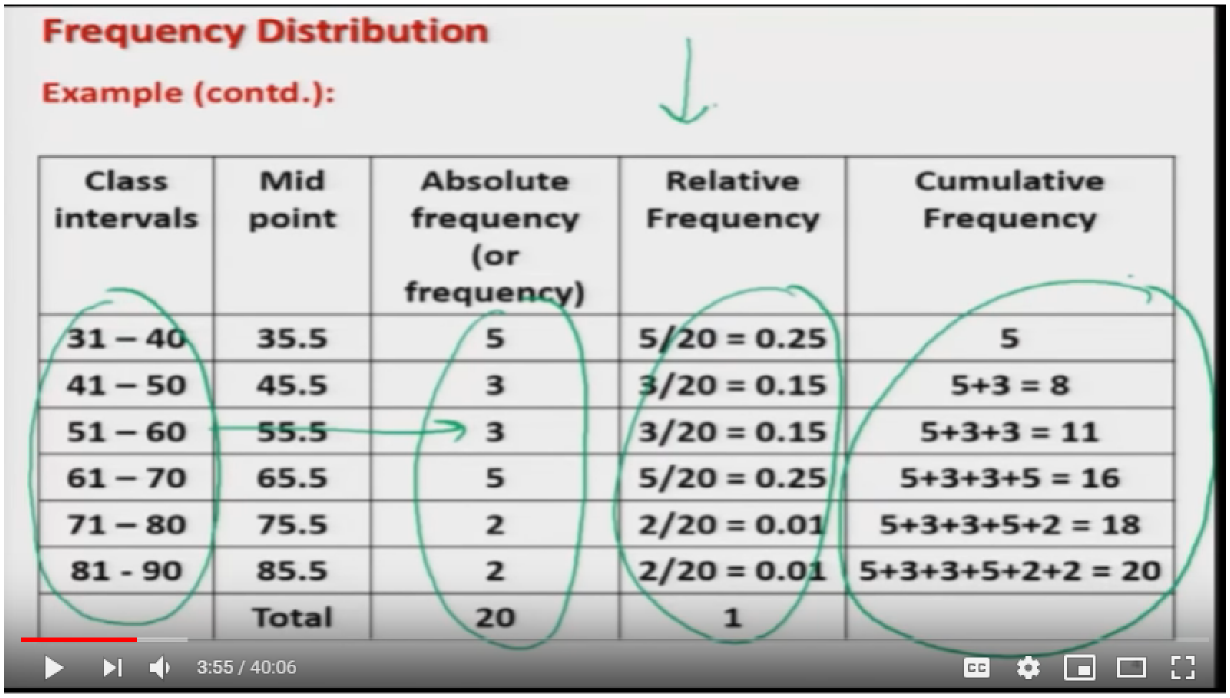
Following are the time taken (in seconds) by 20 participants in a race: 32, 35, 45, 83, 74, 55, 68, 38, 35, 55, 66, 65, 42, 68, 72, 84, 67, 36, 42, 58.

```
> time
[1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68
72 84 67 36 42 58
```

```
> time
 [1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36 42 58
```

you may recall that in the earlier lecture, I had taken an example like, this one, I had recorded the time, taken by 20participants in a race and I had created data vector, time and this data vector consists of 20 values. Now after this our objective was to create a, frequency distribution, you may recall, what was our first step? First step was that, we tried to find out, what is the minimum and maximum value, in this dataset. So, you may recall that we had identified that, 32 is the minimum value and here 84 is the maximum value and using this minimum and maximum values, we need to decide, that how many class intervals, we would like to have we had discussed, that all this data has to be grouped in some suitable number of class intervals or in simple words classes. So, first we need to, decide that, how many class intervals we can make? So, in this example we had decided to make, the class intervals of the width, 10 seconds and based on this you can,

Refer slide time :(03:32)



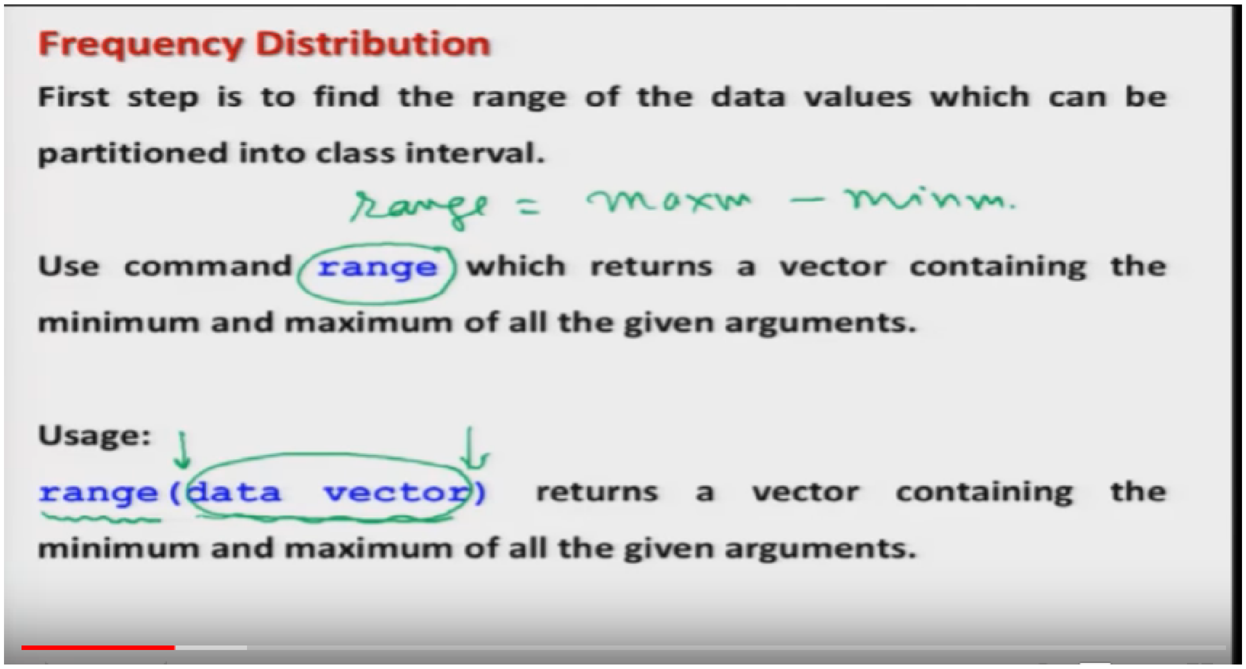
Frequency Distribution

Example (contd.):

Class intervals	Mid point	Absolute frequency (or frequency)	Relative Frequency	Cumulative Frequency
31 – 40	35.5	5	5/20 = 0.25	5
41 – 50	45.5	3	3/20 = 0.15	5+3 = 8
51 – 60	55.5	3	3/20 = 0.15	5+3+3 = 11
61 – 70	65.5	5	5/20 = 0.25	5+3+3+5 = 16
71 – 80	75.5	2	2/20 = 0.01	5+3+3+5+2 = 18
81 - 90	85.5	2	2/20 = 0.01	5+3+3+5+2+2 = 20
	Total	20	1	

have a look on this table, we had constructed this table. So, first we had constructed this class interval and based on that we had found this frequency and based on that we had found the relative frequency and finally we had found the, cumulative frequency, the same thing we are going to now do in the R software.

Refer slide time :(04:00)



Frequency Distribution

First step is to find the range of the data values which can be partitioned into class interval.

range = maxm − minm.

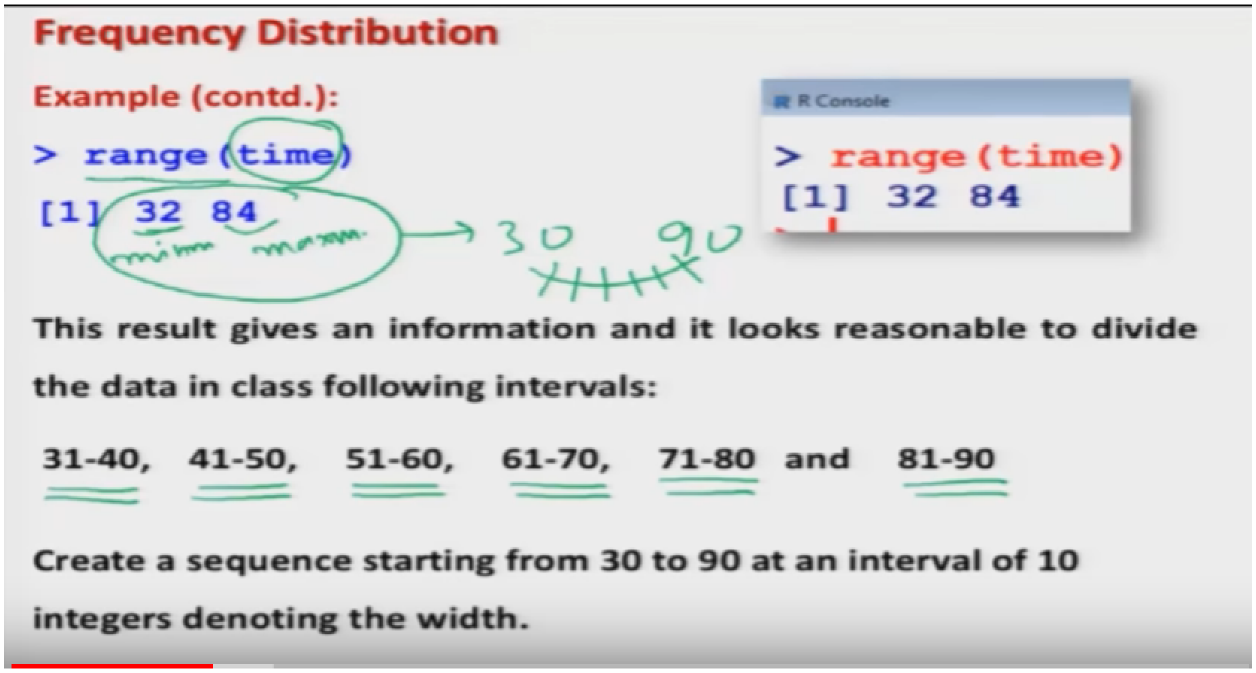
Use command `range` which returns a vector containing the minimum and maximum of all the given arguments.

Usage:

`range(data vector)` returns a vector containing the minimum and maximum of all the given arguments.

So, the first step in the construction, of a frequency distribution is to find out the, range of the data values, the range of the data values is defined as, say here maximum value minus minimum value, once you get a range then you will have an idea that this range has to be partitioned in how many class intervals. So, in order to find out the range, we have a command in R software as range and range and in order to use this, I have to give the name of the command, range and inside these arguments, these brackets, I have to give here the data vector and if I try to do so then the outcome will be, the minimum and maximum values, of the data contained in this data vector, that is the first step. So, what will be that second step? Looking at the value of the range I have to decide the number of, class interval, I have to decide the width of the interval and then I need to partition this range into different segments, what we will call as classes?

Refer slide time :(05:29)


Frequency Distribution

Example (contd.):

```
> range(time)
[1] 32 84
```

```
> range(time)
[1] 32 84
```

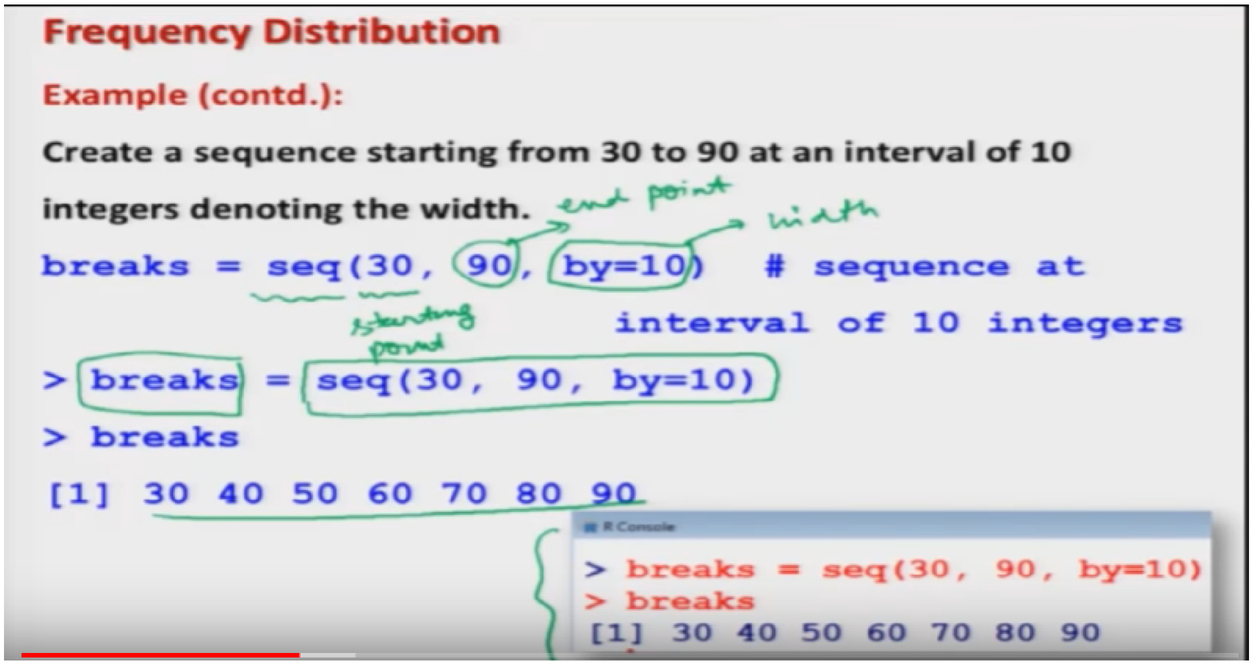
This result gives an information and it looks reasonable to divide the data in class following intervals:

31-40, 41-50, 51-60, 61-70, 71-80 and 81-90

Create a sequence starting from 30 to 90 at an interval of 10 integers denoting the width.

So, you can see here, I have operated, this functional range, over the data of time and you can see here, that this is giving us 32 as the minimum value and 84 as the maximum value and once you can obtain the range, the next step is this, how to divide this range into suitable number of classes. So, we had decided in the earlier lecture, that we are going to have the classes of width 10 seconds. So, we had the classes like a 31 to 40, 41 to 50, 51 to 60, 61 to70, 71 to 80 and 81 to 90. So, now this range, 32 and 84 will be converted into30 and 90 and this range will be partitioned in different classes. So, now I have another task, that how to create this intervals. So, you know when we are trying to, divide a range into different segments of equal length, we can use a command of sequence, we are simply trying to create a sequence or the values of the sequence as sum fixed interval. So, in order to do this thing,

Refer slide time :(07:01)



Frequency Distribution

Example (contd.):

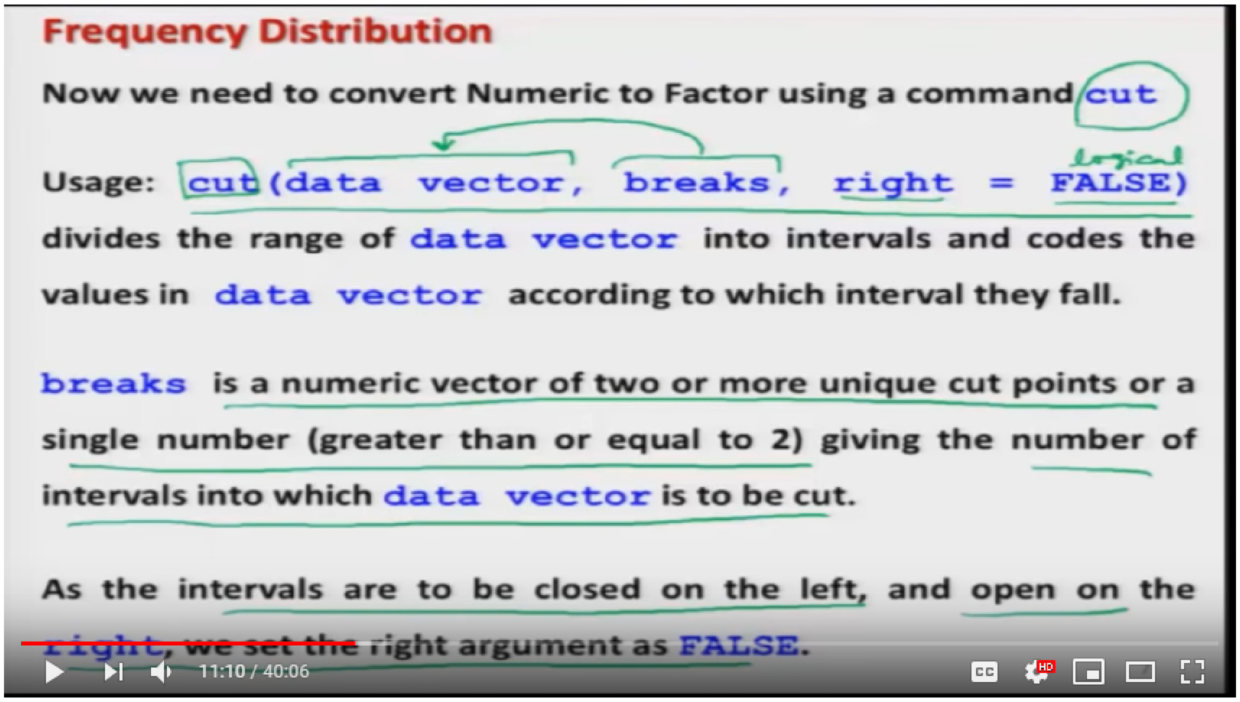
Create a sequence starting from 30 to 90 at an interval of 10 integers denoting the width.

```
breaks = seq(30, 90, by=10)   # sequence at
                                interval of 10 integers
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

```
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

I try to create here, a sequence and in order to do this thing, I'm going to use here the come on seq, I would like to address here that it is not really possible to give you here the details of all the R commands. So, for that you need to learn first the R software and once you learn the R software the basic things, then you will be able to use them or the statistical function. So, in case if you want to do it you can go to the slides available, in my another lecture, on R software that is introduction to R software, its slides and the videos are available, on the NPTEL website you can have a look. So, anyway without explaining the use and function of this operator seq, I would try to use it and the first value inside the argument is giving me the starting point and the second value in the argument is giving me the end point, that means I need to create a sequence from30 to 90 and this here y equal to 10 is giving me value or we are providing a value, that what should be the width of the interval, that means the sequence has to be broken at which point. So, once I start with the 30 then the sequence will be broken at 30 40 50 60 70 80 and 90. Right? So, in case if I try to store the outcome of this command, that trying to break the sequence into seven classes at an interval of 10 units, I would like to store all the things in, a new variable here breaks, well I'm going to use this breaks variable later on, in the construction of frequency table. So, once I try to execute, it on the R console I will get it here this value and this is here the, the screenshot. Right? Now once I get, this sequence, at an interval of 10 units is starting from,

Refer slide time :(09:14)



Frequency Distribution

Now we need to convert Numeric to Factor using a command cut

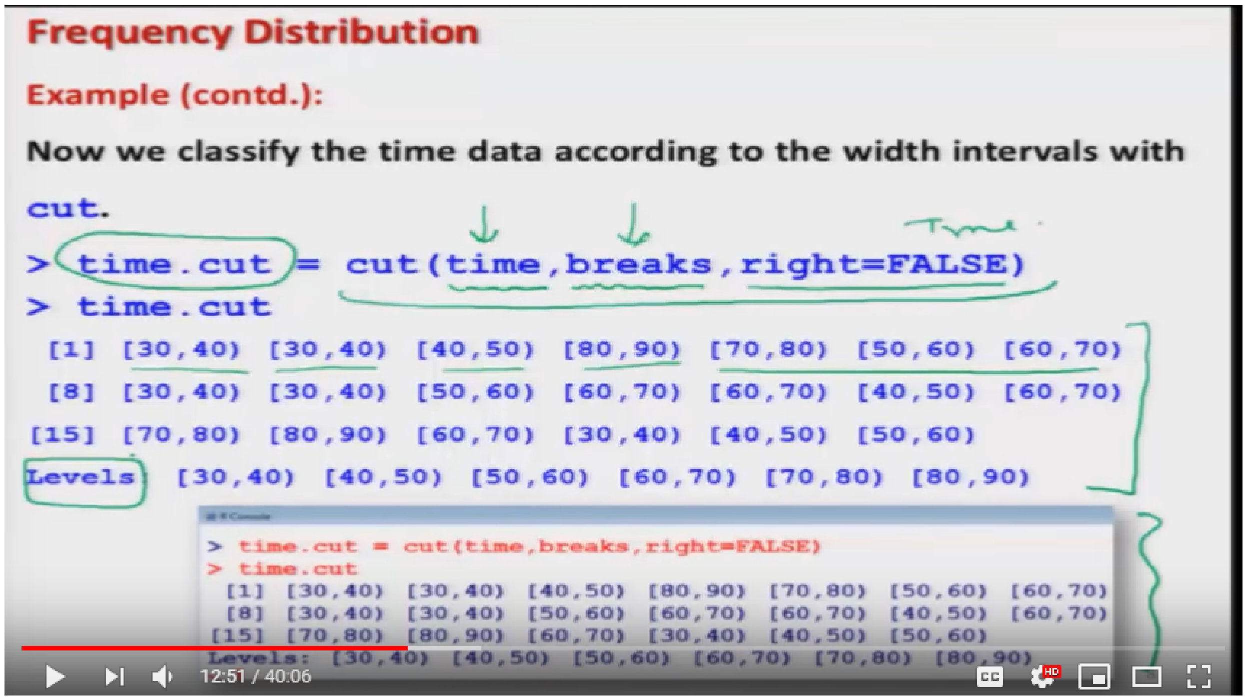
Usage: cut(data vector, breaks, right = FALSE) divides the range of data vector into intervals and codes the values in data vector according to which interval they fall.

breaks is a numeric vector of two or more unique cut points or a single number (greater than or equal to 2) giving the number of intervals into which data vector is to be cut.

As the intervals are to be closed on the left, and open on the right, we set the right argument as FALSE.

30 to 90, then I need to convert this, numeric vector into a factor, once again you need to know what are the factors in R software and for that I will again request you to have a look on the lectures on introduction to R software and in order to achieve this we have a command here cut and this command, is used in this particular way, I would try to use the command here CUT, cut and then I need to specify the data vector, for which I would like to operate the, function cut and then I would like to define here the brakes, brake is going to be a numeric vector, of two or more unique cut points, or a single number, which is greater than equal to two and this will give us the number of intervals in which this data vector has to be cut. So, this break is going to control the number of partitions in this data vector and here I have here, our command, small letters right, right and this is equal to here false and you can see here this is the logical operator,   discussed capital letters true and capital letters fault they are the logical operators and this, write equal to false is denoting, that the intervals are to be closed on the left and open on the right and this statement, if we want to set as false. So, in case if you want to make it true then, then instead of here Falls you have to give it true. So, let us try to execute this command, over the data vector.

Refer slide time :(11:14)



Frequency Distribution

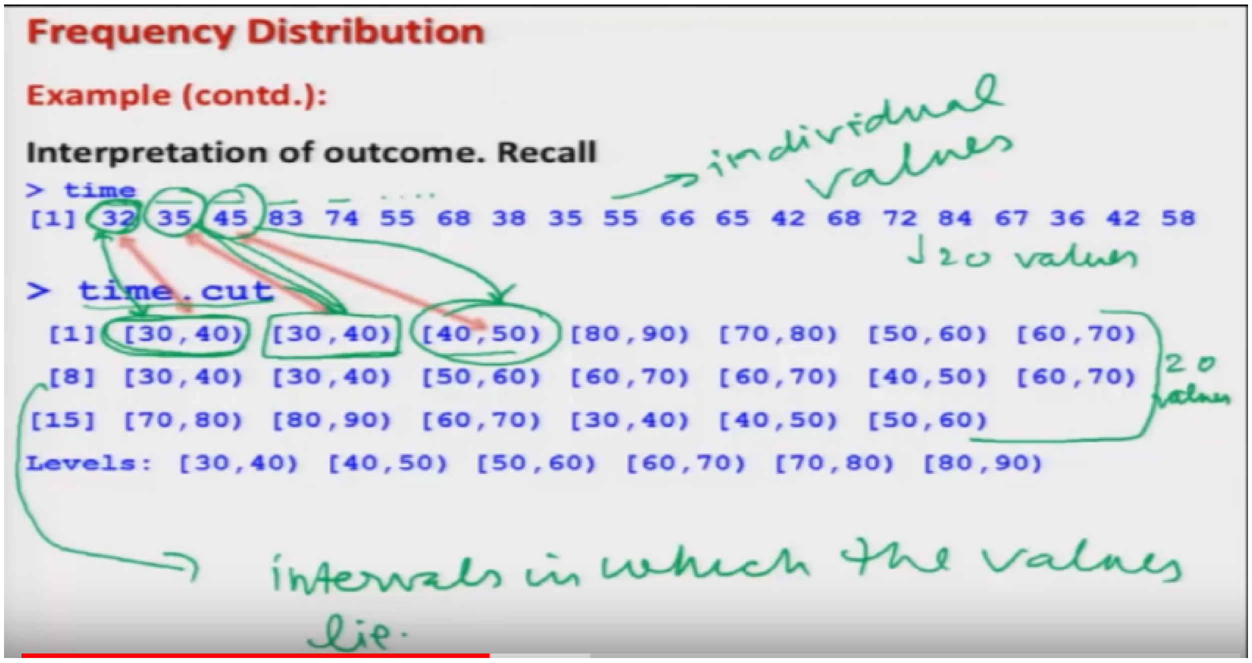
Example (contd.):

Now we classify the time data according to the width intervals with cut.

```
> time.cut = cut(time,breaks,right=FALSE)
> time.cut
 [1]  [30,40)  [30,40)  [40,50)  [80,90)  [70,80)  [50,60)  [60,70)
 [8]  [30,40)  [30,40)  [50,60)  [60,70)  [60,70)  [40,50)  [60,70)
[15]  [70,80)  [80,90)  [60,70)  [30,40)  [40,50)  [50,60)
Levels  [30,40)  [40,50)  [50,60)  [60,70)  [70,80)  [80,90)
```

and then we try to see, what happens? And what is the interpretation? And once I complete this thing, then I will take you to the R console and I will try to show you, what is really happening. Okay? So, if you try to see here, I try to take the same data vector here time and I am trying to use here the data ,which I have already generated as brakes, remember we have generated the brake where here, 30, 40, 50, 60, 70, 80 and 90. So, now I am trying to tell this function, that please use the data of time, vector and create the bricks, using the data in the brick and write is going to be false, that means all the intervals are going to be closed on the left hand side and they will be open on the right hand side, in case if you want that the vectors have to be closed on right hand side also, then you need to use here the command true, but anyway I have not used it here. And whatever is the outcome of this thing I am trying to store it in a new variable here time dot cut. So, this is simply indicating, that the time has been operated with the, command cut and the outcome of this time dot cut will lookalike this and here is the screenshot, you can see here there are values here 30 to40, 30 to 40, 40 to 50, 80 to 90. And so, on and here there are some hair levels, what are these values indicating you see, this is very important in any software, that you need to understand what the software is doing because unless and until you ,you understand it you will not be able to execute the, the correct command on, the given outcome. So, first we try to understand,

Refer slide time :(13:14)



Frequency Distribution

Example (contd.):

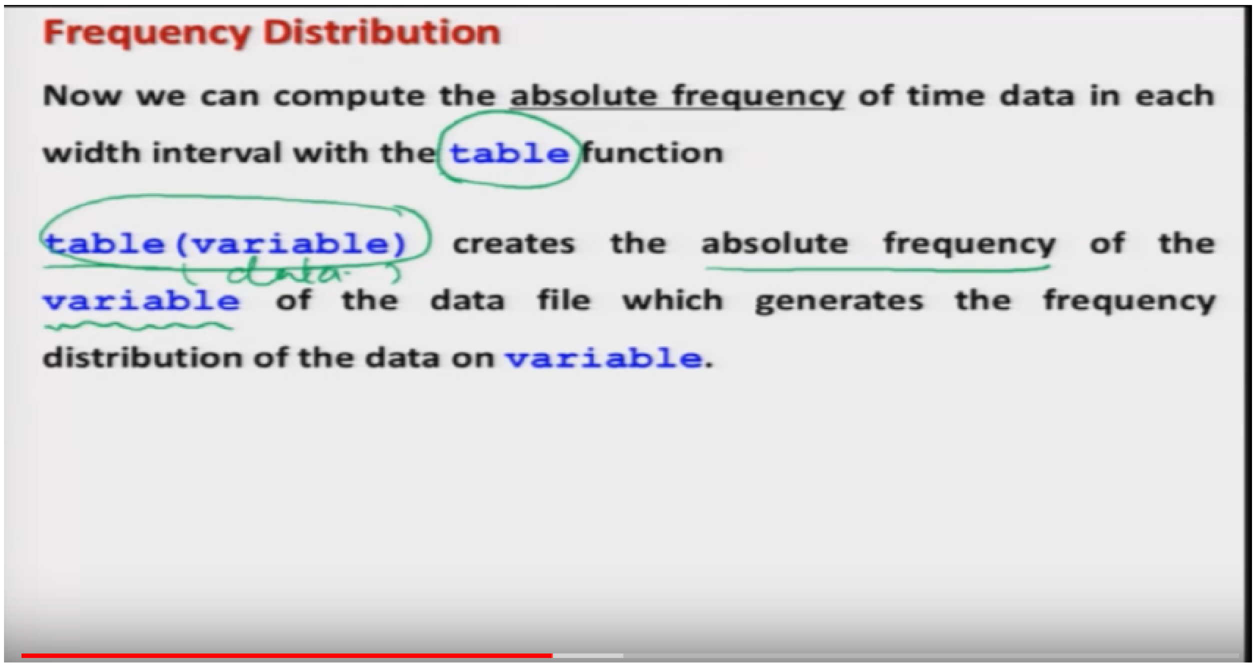
Interpretation of outcome. Recall

```
> time
[1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36 42 58

> time.cut
 [1] [30,40) [30,40) [40,50) [80,90) [70,80) [50,60) [60,70)
 [8] [30,40) [30,40) [50,60) [60,70) [60,70) [40,50) [60,70)
[15] [70,80) [80,90) [60,70) [30,40) [40,50) [50,60)
Levels: [30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
```

what is the meaning of this outcome? You will recall that the data in your time vector was, 32, 35, 45, 83, 74 and so on. And if you try to see here what is the outcome of this ,variable time dot cut I have simply copied and pasted these two data vectors over here, this is 30 to 40 to 40, 40 to 50 and so on. And you can see here from here, to here there are only here 20 values. So, what is really happening? That these are my individual data's, these are my individual values in the time vector. And now I have created the intervals, in which these values are going to lie. So, these are the intervals in which the values lie, for example suppose I take this value here 32, this is my first value and what is the first value in that variable time dot cut, this is 30 to 40 so you can see here this is indicating that this value 32 is lying in the interval, 30 to 40similarly, thus second value here, the second interval that is indicating, that where the second value of the vector time is lying. So, you can see here 35 is lying, between 30 and 40. So, the second interval is indicating the interval of the second value and similarly if you move forward the third interval here is indicating the interval in which this third value 45 is lying and so on. So, you can see here, there are 20 values here and there are 20 values in the data vector time dot cut. So, every value is corresponding, to the interval in which it is lying or the 20 values or the 20intervals in the time dot cut vector they are indicating or they are providing, the interval in which the corresponding values are lying. So, this is how the interpretation of this time dot cut goes. Right? Now what we have to do? We have got this there and now I have to create, a frequency table. So, we had learned in the earlier lecture that in order to do. So, we have a function here table, using that table command, we had constructed the frequency table in the case of discrete data also or in the case of qualitative data that was finally converted into, some numerical values, indicating the, the variables values. Okay? So, the same command I will use here, but now this command is going to be used over this new variable time dot cut, because now we have got this interval and then we have got the data also time, now I need to create the frequency distribution using, this data vector.

Refer slide time :(16:42)



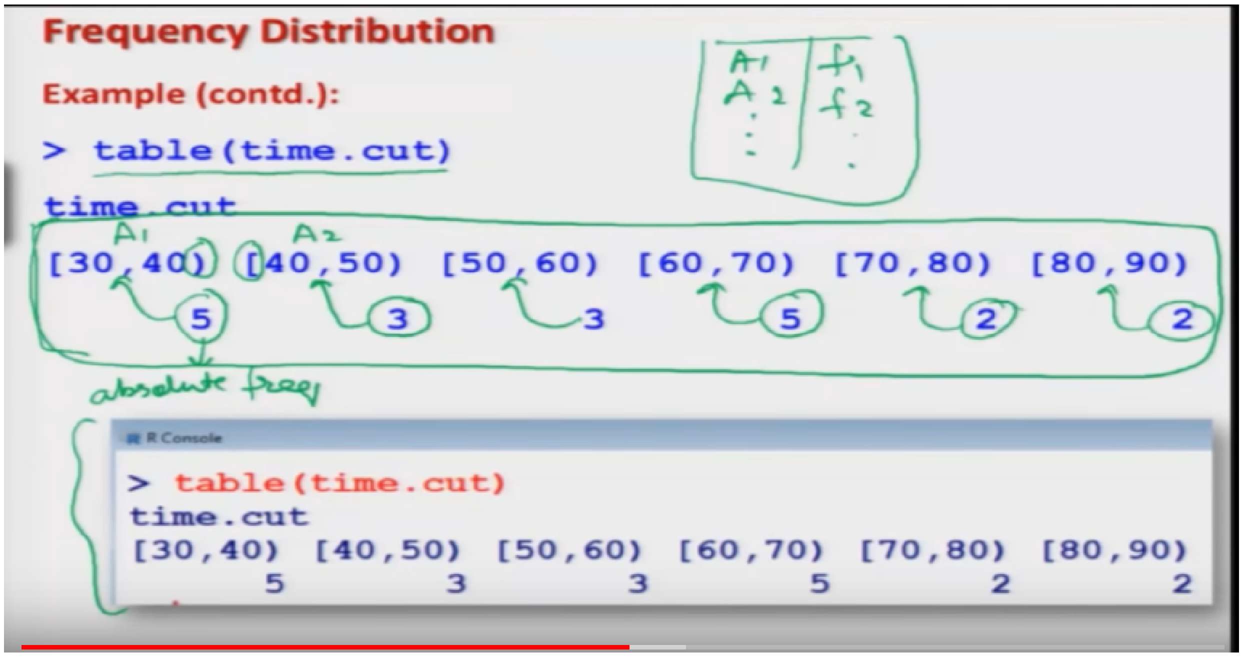
Frequency Distribution

Now we can compute the absolute frequency of time data in each width interval with the table function

table(variable) creates the absolute frequency of the variable of the data file which generates the frequency distribution of the data on variable.

So, as we had discussed earlier, now we are going to use the absolute frequencies of this data vector, using the table function. So, the users will be table, you have to type table all in a small letter and inside the arguments you have to write the data on the variable. So, here you try to write the data. So, now once you try to execute table dot variable then it will create the absolute frequencies, corresponding to the data, which is contained in the arguments under the variable name variable. So, table variable inside the argument will simply try to create or simply try to inform you the values of the absolute frequencies, with respect to different intervals. So, now when I try to execute it, over the R software,

Refer slide time :(17:43)


Frequency Distribution

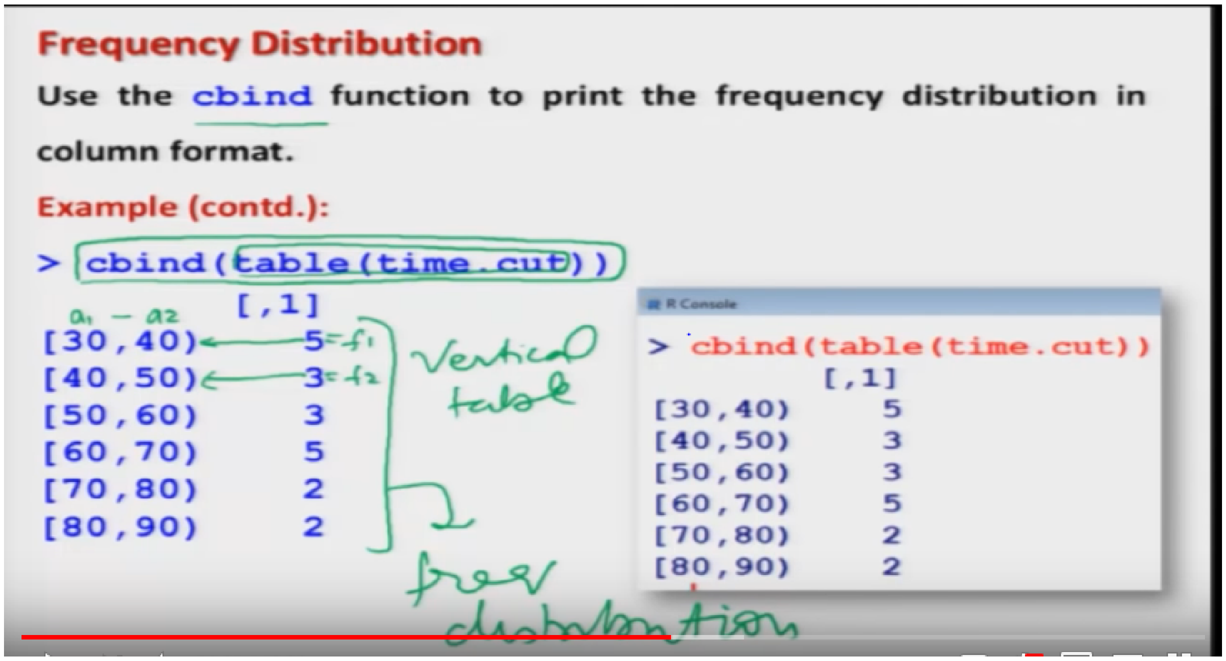
Example (contd.):

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       3       3       5       2       2
```

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       3       3       5       2       2
```

you can see here, once I try to rotate table and inside the, bracket I say time dot cut. So, you can see here I get this type of outcome, what is this telling you? This is trying to say that there are five values, in the interval 30 to 40. So,5 is essentially the, absolute frequency, of class a1 which is equal to 30 to 40.And similarly here this here is 3 this, 3is trying to indicate that there are 3values in the interval 40 to 50 which is our class a 2 and similarly there are 3values here which are between, 50 to 60there are 5 values which are lying in the interval, 60 to 70 there are 2 values which are lying in the interval 70 to 80and there are 2 values which are lying in the interval 80 to 90, well you can see here one thing, that here the intervals are continuous, because I have used here, open interval here and close interval here and that is the way we try to create the frequency distribution in a continuous data, well in this case because all the values are going to be some integers. So, that is why you have to make sure that if there is a value 40 on the boundary then where it is going to be added, in the earlier interval or in the next interval. So, you have to be careful. Okay? And now this is the screenshot, but you can see here one thing usually in that textbooks, whenever we try to write down the frequency table, they are not written horizontally, but they are written vertically like this, means here you will have here class intervals a 1 a 2 and so on and here you will have frequency f1, f2 and so on. So, now but this is the outcome, you can see here, this is actually horizontal this frequency table, is coming like cookies here this is horizontal and we would like to make it vertical.

Refer slide time :(20:19)



Frequency Distribution

Use the cbind function to print the frequency distribution in column format.

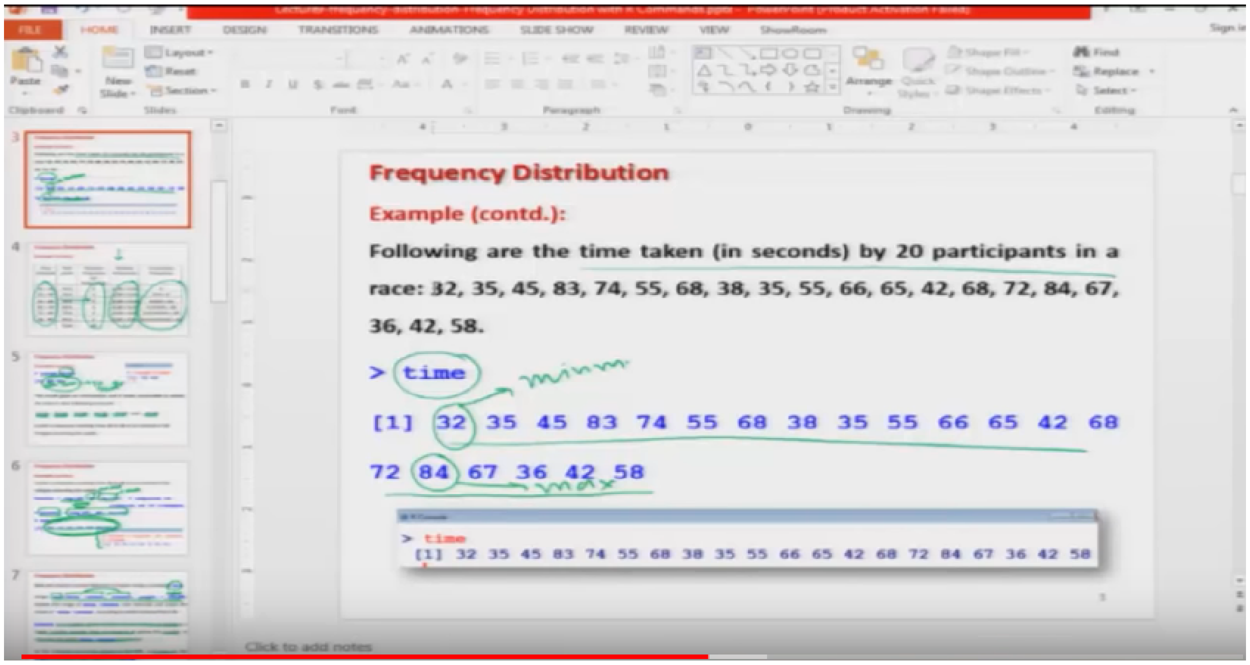
Example (contd.):

```
> cbind(table(time.cut))
        [,1]
[30,40)    5
[40,50)    3
[50,60)    3
[60,70)    5
[70,80)    2
[80,90)    2
```

```
> cbind(table(time.cut))
        [,1]
[30,40)    5
[40,50)    3
[50,60)    3
[60,70)    5
[70,80)    2
[80,90)    2
```

So, in order to make it vertical, we have a command here, cbind this function is used to print the frequency distribution in the column format, up to now, the intervals and the irrespective frequencies they are coming in rows, now I want to make it column wise. So, I have to do nothing, the same outcome whatever we have obtained here, I simply have to operate the, cbind function over that. So, you can see here I try to obtain here the cbind and inside the argument I am simply trying to write down the table and inside the argument I'm dot cut and this is the same command, if you try to see here what I have used here where I am circling. So, whatever is the outcome of this, command this is being used herewith the function cbind and you can see here you get here a vertical table. And this is you're here, frequency distribution. So, it's the same thing, there are five values in the interval, 30to 40 there are three values in the interval 40 to 50 and so, on. So, this is the lower limit, say a1 a2 and this is my interval a1 to a2 and this is a frequency, f1 this is the frequency f2and so on. So, that is the same table that we had obtained earlier there is a difference of say between 31 and to 40and 41 to 50 but that can also be adjusted by using the appropriate intervals. So, that is not going to make much difference, we had taken manually, 32 41 and 31 to 40and 41 to 50 because, we were doing it manually and all our data values were an integer. So, in case if you have fraction the data, the same concept will continue.

Refer slide time :(22:24)



Frequency Distribution

Example (contd.):

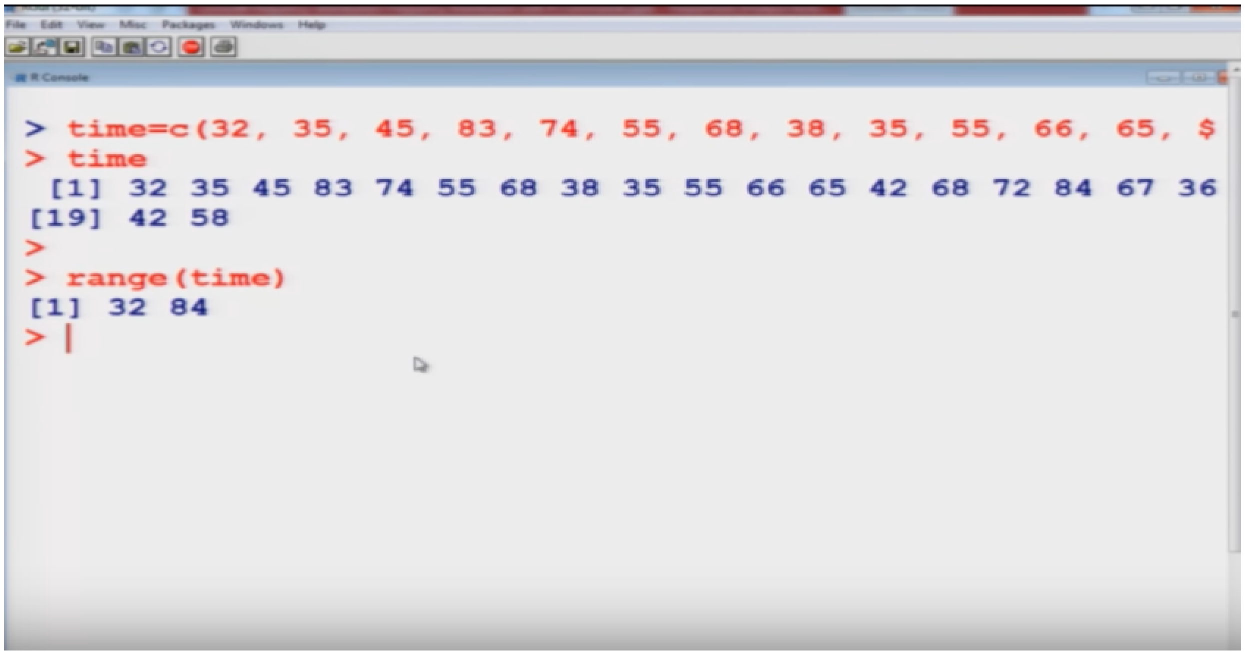
Following are the time taken (in seconds) by 20 participants in a race: 32, 35, 45, 83, 74, 55, 68, 38, 35, 55, 66, 65, 42, 68, 72, 84, 67, 36, 42, 58.

```
> time
[1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68
72 84 67 36 42 58
```

```
> time
 [1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36 42 58
```

But now, before going further let me try to show you these things over the R console. So, what we try to star there, with the data. So, you can see here your data was here like this, say here time. So, I would try to copy it here.

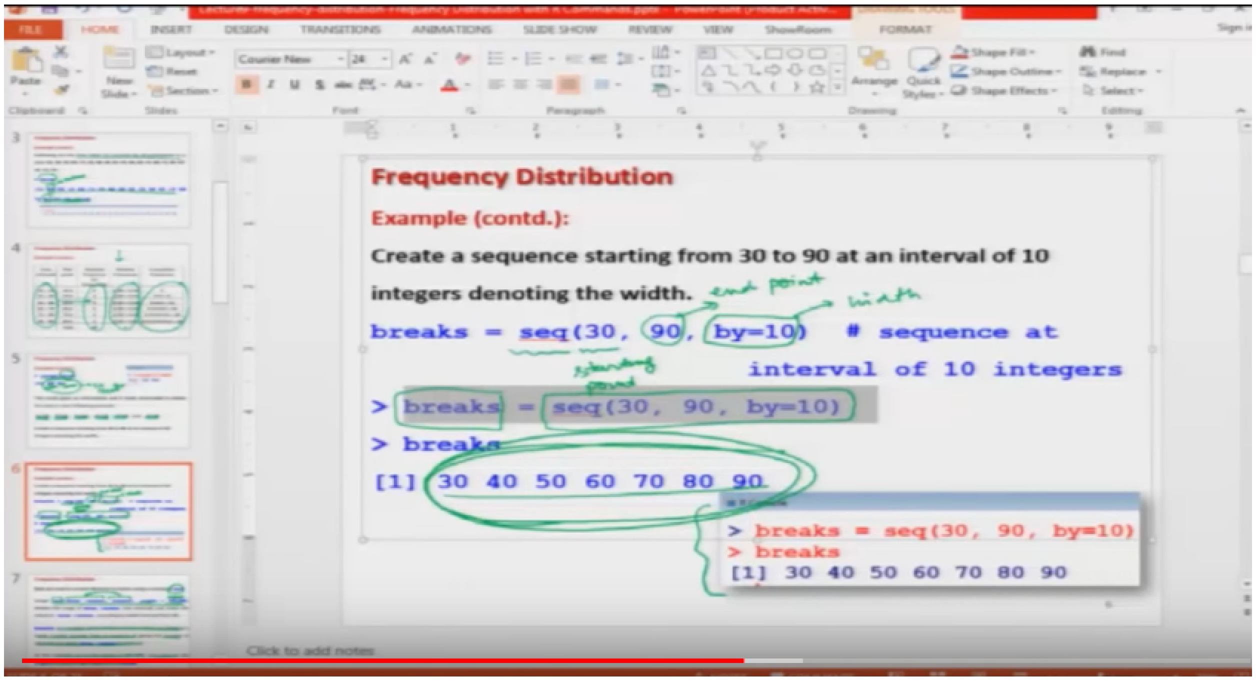
Refer slide time :(22:42)



```
> time=c(32, 35, 45, 83, 74, 55, 68, 38, 35, 55, 66, 65, $
> time
 [1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36
[19] 42 58
>
> range(time)
[1] 32 84
> |
```

 I will come to here R console and I will say her time, time is equal to C and inside the bracket I have to give that data. So, you can see here now this is my here data. Right? On which I would like to create my, frequency distribution. Now the first step is this I want to find out the range of this data. So, I have to operate range of say time and this comes out to be here 30 to 34. So, now this is giving us an idea, that we can have an interval of, 10 units starting from 30till 90.

Refer slide time :(23:26)



Frequency Distribution

Example (contd.):

Create a sequence starting from 30 to 90 at an interval of 10 integers denoting the width.

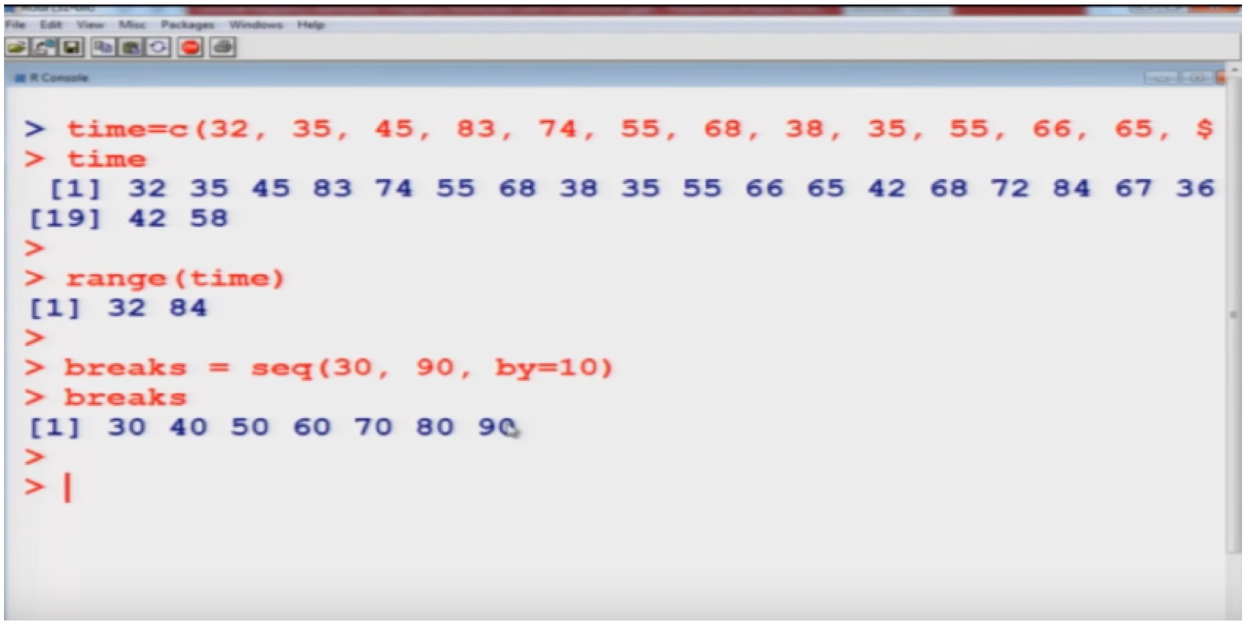
```
breaks = seq(30, 90, by=10)   # sequence at
                 interval of 10 integers

> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

```
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

So, now after this, what I have to do? I need to first create the sequence, at an interval of 10.

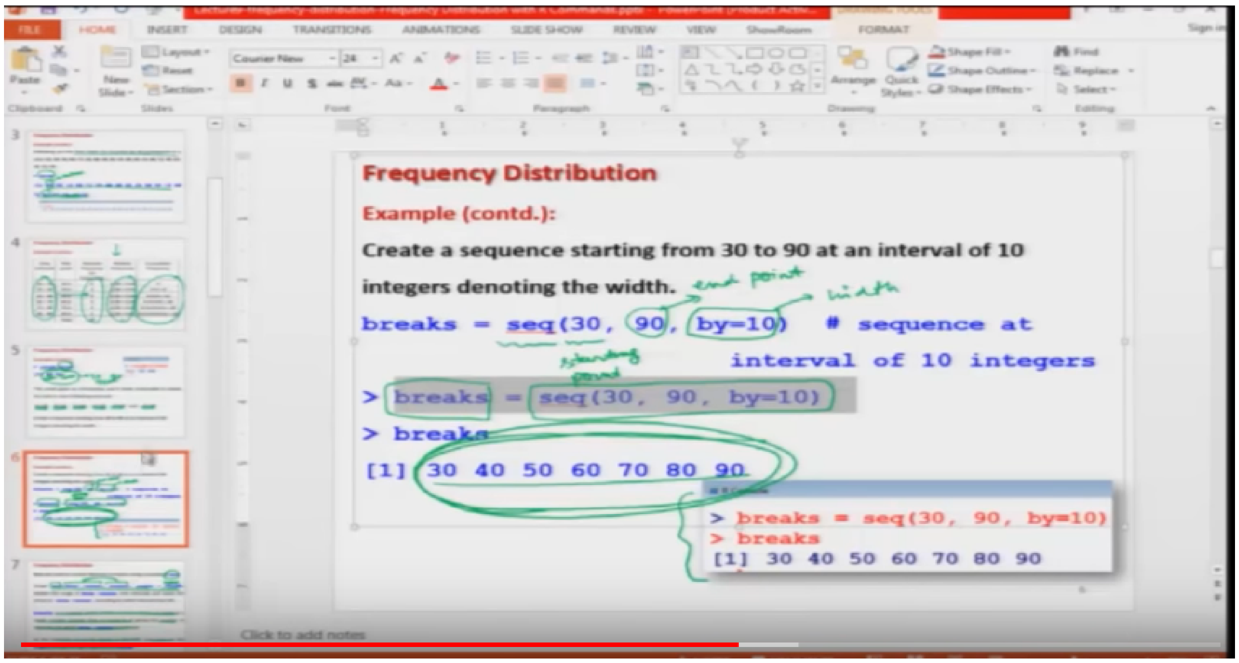
Refer slide time :(23:41)



```
> time=c(32, 35, 45, 83, 74, 55, 68, 38, 35, 55, 66, 65, $
> time
 [1] 32 35 45 83 74 55 68 38 35 55 66 65 42 68 72 84 67 36
[19] 42 58
>
> range(time)
[1] 32 84
>
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
>
> |
```

So, I would try to create here a sequence, using the break and if you try to see here, this breaks comes out to be the same outcome that we had done, earlier the use of the values30 40 50 60 70 80 and 90 here. Right?

Refer slide time :(23:58)



Frequency Distribution

Example (contd.):

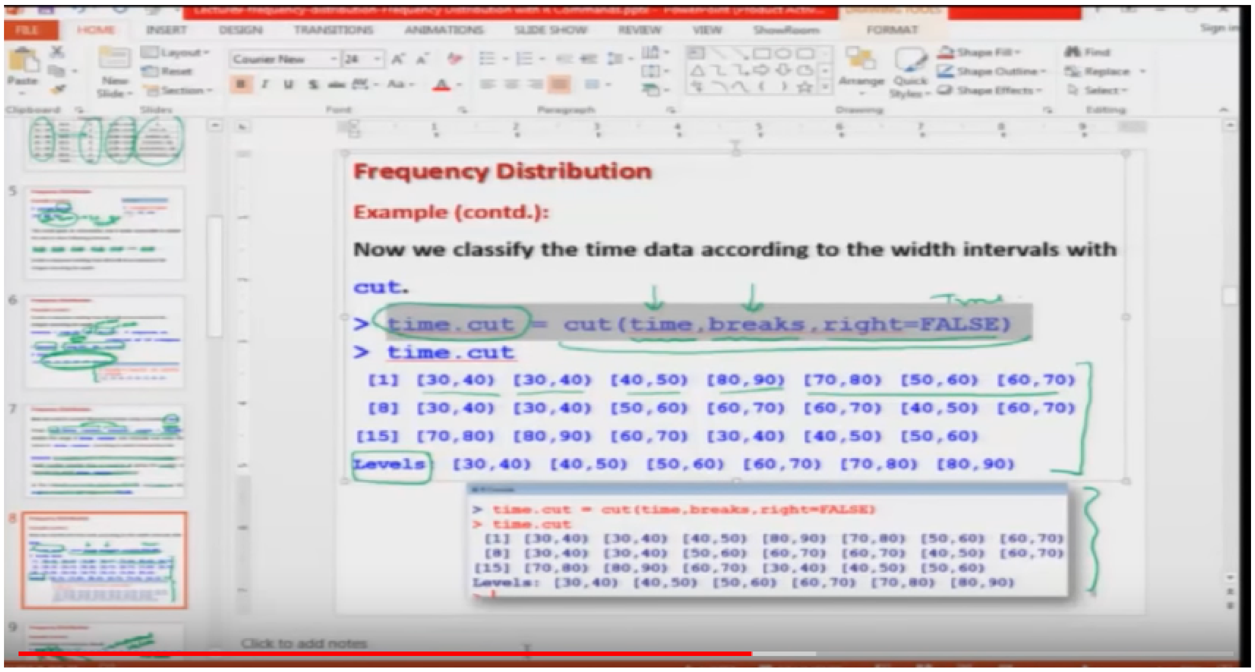
Create a sequence starting from 30 to 90 at an interval of 10 integers denoting the width.

```
breaks = seq(30, 90, by=10)   # sequence at
                                interval of 10 integers
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

```
> breaks = seq(30, 90, by=10)
> breaks
[1] 30 40 50 60 70 80 90
```

Now after this, what we have to do?

Refer slide time :(24:03)



Frequency Distribution

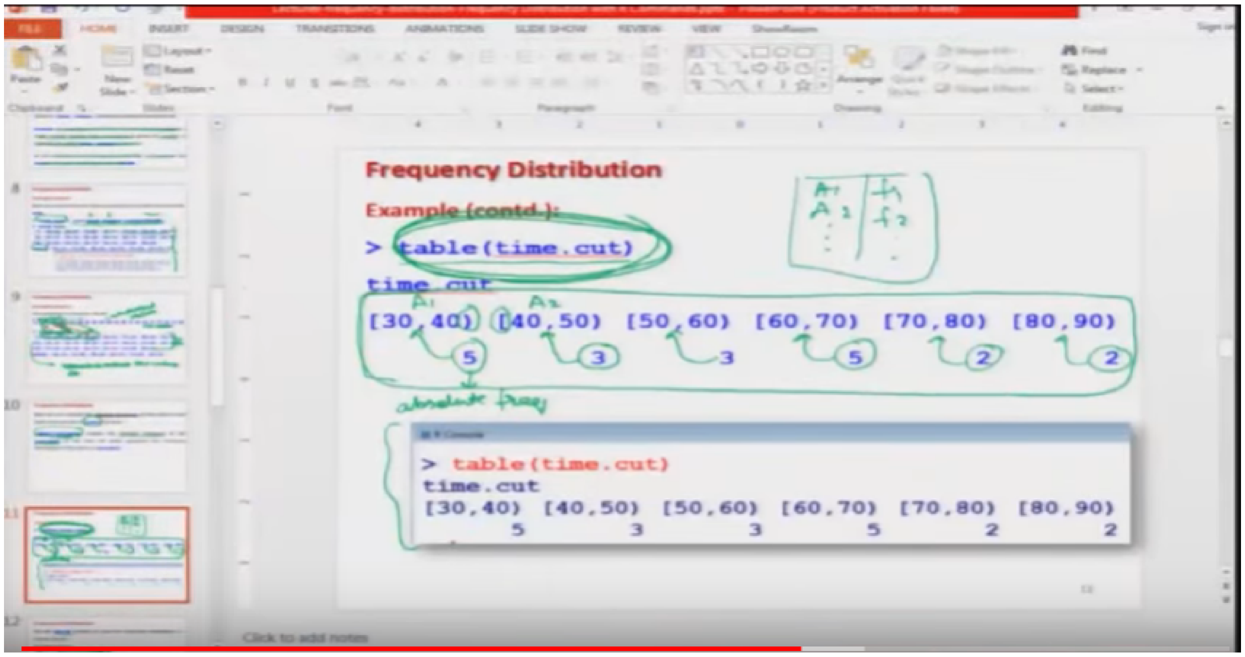
Example (contd.):

Now we classify the time data according to the width intervals with cut.

```
> time.cut = cut(time,breaks,right=FALSE)
> time.cut
 [1] [30,40) [30,40) [40,50) [80,90) [70,80) [50,60) [60,70)
 [8] [30,40) [30,40) [50,60) [60,70) [60,70) [40,50) [60,70)
[15] [70,80) [80,90) [60,70) [30,40) [40,50) [50,60)
Levels: [30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
```

I simply have to create here, I have to use here the command cut. So, I try to use the data, on time and breaks.

Refer slide time :(24:03)



Frequency Distribution

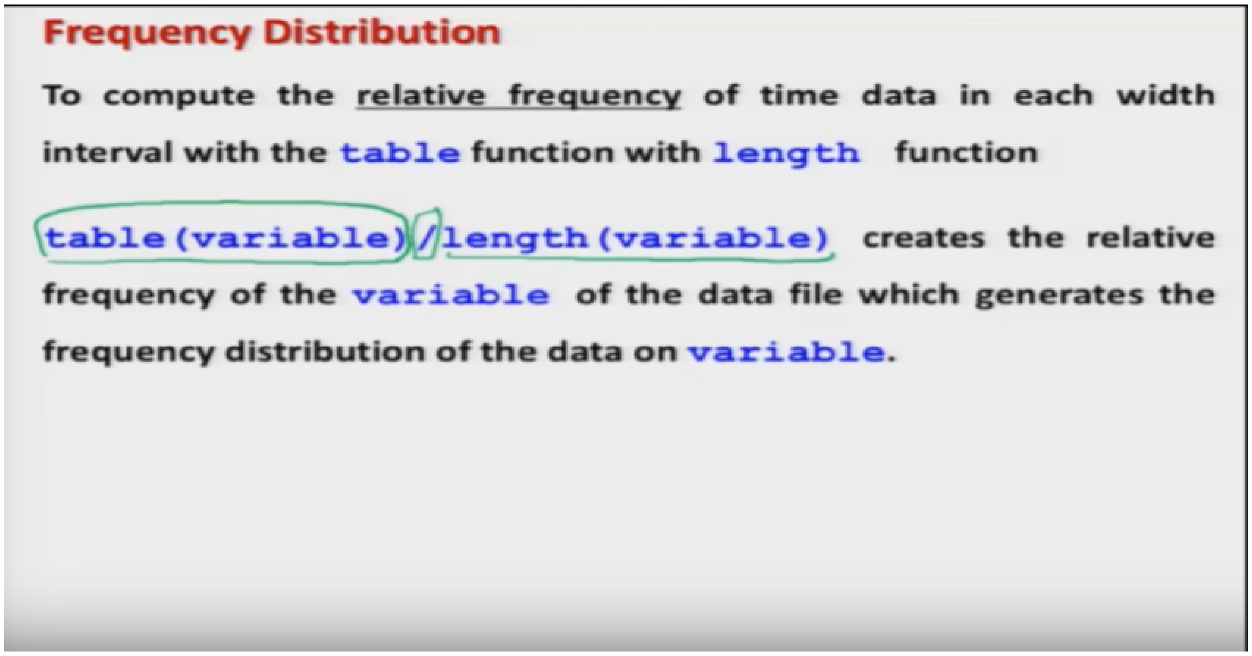
Example (contd.):

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
```

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       3       3       5       2       2
```

Time is here, break is here and I have to use this data, to get the values of time cut and you can see here time dot cut comes out to be like this. Right? And after this, what I have to do? I simply have to operate the table function over this as, you can see in, this light I simply have to use here the table function. So, if I try to use here the table function. So, you can see here table time dot cut and you can see here, this is, the same outcome here, same outcome here this is the frequency table that we have obtained, but you can see here that this frequency distribution table is in horizontal. So, I want to make it vertical. So, I can use the command here, cbind and then I will try to type here the same command or the same data that we want to convert into vertical, which is time dot cut and the outcome of the function table, time dot cut has to be converted into a vertical table. So, you can see here I am getting this thing here. So, now you can see here this is the same frequency distribution that you used to obtain by the manual calculations. Right?

Refer slide time :(24:03)



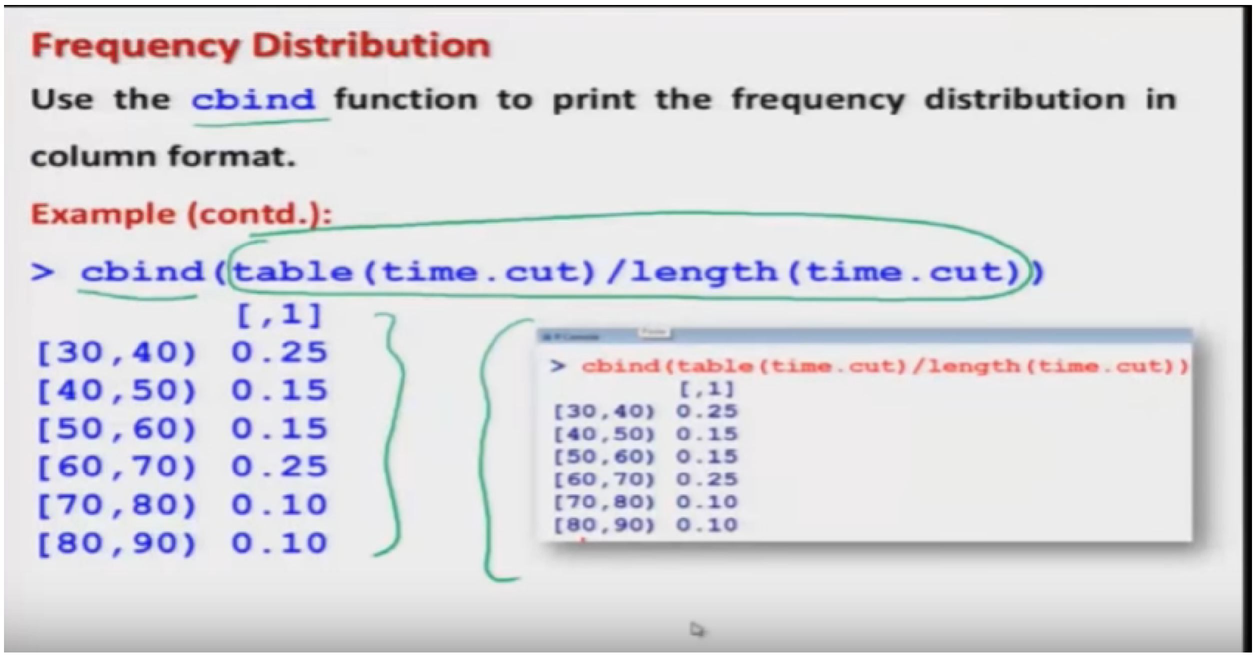
Frequency Distribution

To compute the relative frequency of time data in each width interval with the table function with length function

table(variable)/length(variable) creates the relative frequency of the variable of the data file which generates the frequency distribution of the data on variable.

Now there is another issue, they issues this in this case, you have obtained the frequency distribution with respect to absolute frequencies. Now suppose alternatively, you want to find out this frequency distribution with respect to relative frequencies, then how to do it? So, as we had discussed in the earlier lecture, that there is a very close connection between absolute frequencies and relative frequencies, the relative frequencies are simply obtained, by dividing the absolute frequency by total number of observation, for which we had used the command length. So, in order to find out the frequency distribution with respect to the relative frequency in the same data set, I simply have to divide the, variables at appropriate places, by length of the data vector. So, yes so, in the last lecture, we had obtained the frequency distribution, using absolute frequency and then I had divided it by length of n that has given us the relative frequency. So, the same concept, I am going to use it here once again. So, what have we do here in order to obtain the frequency distribution with absolute frequency, I have this function table and inside the argument I have to give the data on the variable, for which I want to create the frequency distribution. Now I will try to divide it, by the length of the variable. So, the length of the variable is simply going to be the number of observation present in the data vector and once you try to do it you will get the frequency distribution with respect to,

Refer slide time :(27:43)



Frequency Distribution

Use the cbind function to print the frequency distribution in column format.

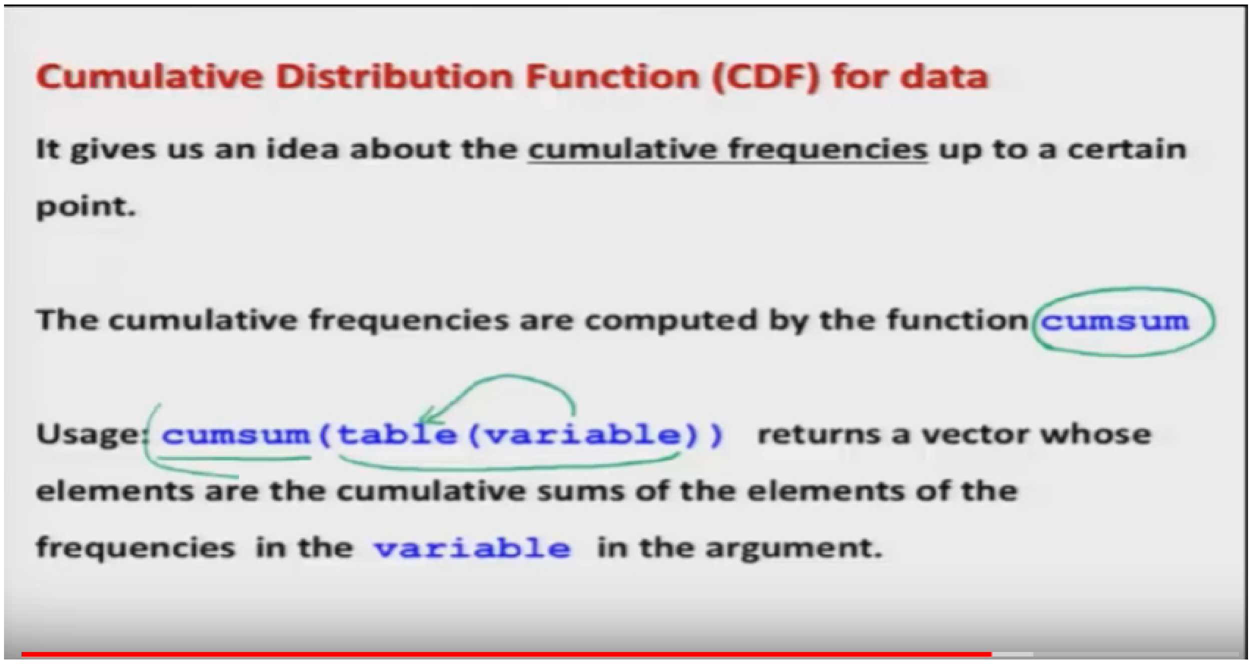
Example (contd.):

```
> cbind(table(time.cut)/length(time.cut))
        [,1]
[30,40) 0.25
[40,50) 0.15
[50,60) 0.15
[60,70) 0.25
[70,80) 0.10
[80,90) 0.10
```

```
> cbind(table(time.cut)/length(time.cut))
        [,1]
[30,40) 0.25
[40,50) 0.15
[50,60) 0.15
[60,70) 0.25
[70,80) 0.10
[80,90) 0.10
```

relative frequency. So, if you try to see here I have obtained the first third frequency distribution with respect to relative frequency. So, you can see here I have divided by here the vector time dot cut length and I am now getting here this value here 0.25 0.15, 0.15 and so, on what are these values, if you try to see what was your outcome in the earlier case the outcome was your here5, 3, 3, 5, 2, 2. So, I will try to copy this thing here 5, 3, 3, 5, 2, 2 and then now I am going to divide it by here 20 because there are 20 observations, 20 and here 20and now you can see here whatever is this outcome 0.25 this is nothing but 5divided by 20 and this is here the that screenshot and once again if you want to put it in the vertical columns, then the vertical format, then you simply have to use the, same command here cbind, but now cbind will be operated over this variable, which was used to obtain the frequency distribution with respect to the relate frequency's and here this is the same outcome and the vertical columns and this is here the screenshot. So, I would like to show you the same outcome on the R console also. So, you can see here. Right? So, we had obtained the data here time dot cut, time dot cut was this thing and now we are going to obtain the frequency distribution. So, this will be table time dot cut divided by length of this time dot cut. So, you can see here you are getting the same outcome and if you want to make it vertical, then I would say here you simply have to operate here the command C bind, write this and you can see here, this is once again the frequency distribution where the frequencies are in terms of relative frequency, here. Right? So, this is how we try to create our frequency distributions, now after this, the next thing comes the last column.

Refer slide time :(30:26)



Cumulative Distribution Function (CDF) for data

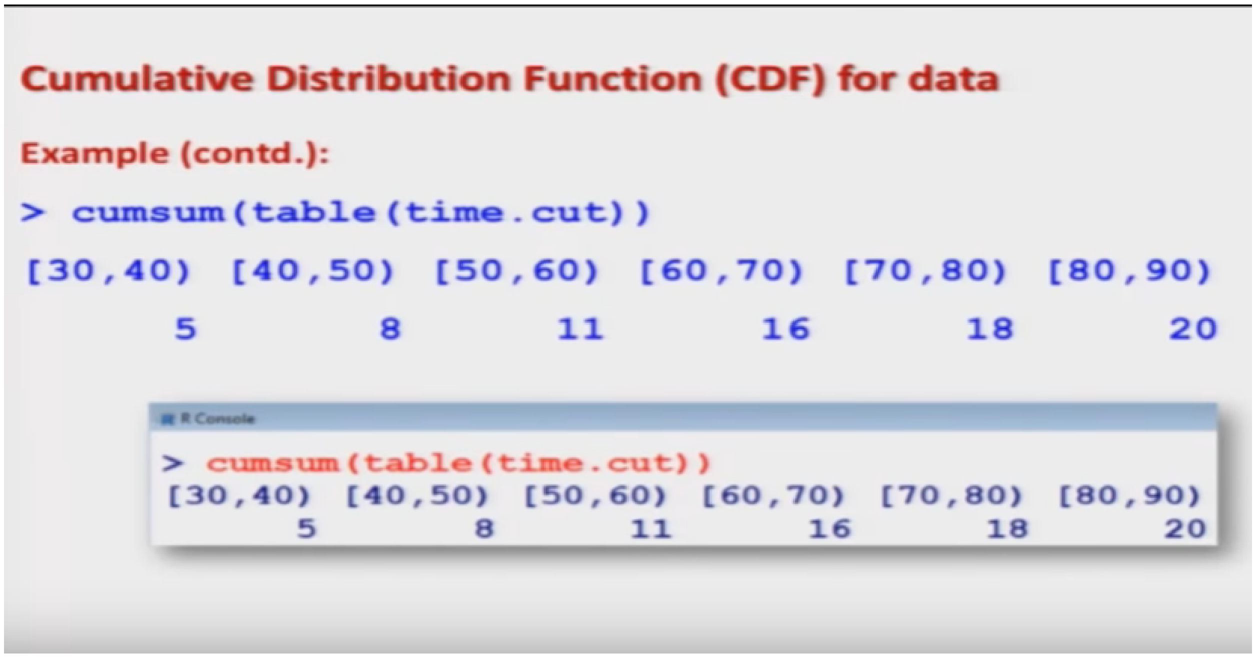
It gives us an idea about the cumulative frequencies up to a certain point.

The cumulative frequencies are computed by the function `cumsum`

Usage: `cumsum(table(variable))` returns a vector whose elements are the cumulative sums of the elements of the frequencies in the `variable` in the argument.

which was the cumulative distribution function or the data or the cumulative frequencies we would like to compute. So, as we had discussed, that the cumulative frequencies gives us an idea up to a certain point, the frequencies up to that particular point, that how many values are less than or equal to this particular value. So, in order to compute the cumulative frequencies, we can use here a function here, Cumsum. So, that is trying to abbreviate, cumulative sum and in order to use this thing, we simply have to use this function Cumsum on the variable for, which we want to create the cumulative totals, but the variable has to be first operated with the table function, because once we have a data in some variable then first it needs to be converted into a frequency table and then based on that frequency table, the sum of those frequencies, can be obtained. So, that is why the complete command will be the Cumsum function, on the table variable. Right? And in case if you try to do it? This will produce the cumulative frequencies.

Refer slide time :(32:01)



Cumulative Distribution Function (CDF) for data

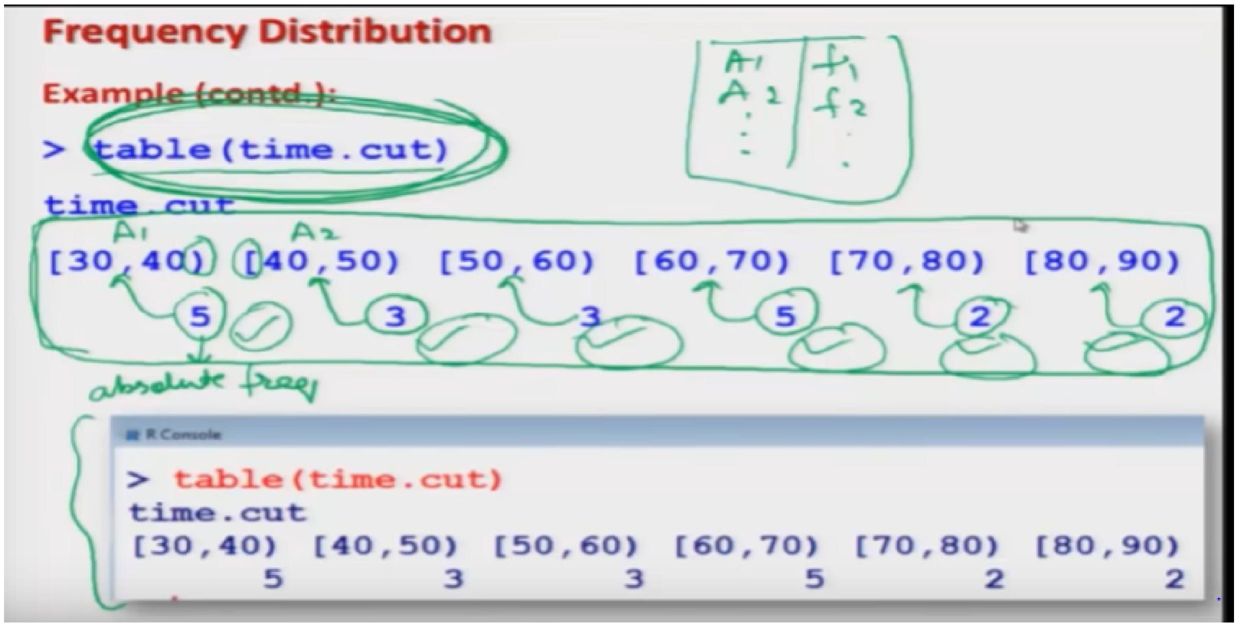
Example (contd.):

```
> cumsum(table(time.cut))
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       8      11      16      18      20
```

```
> cumsum(table(time.cut))
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       8      11      16      18      20
```

So, let me try to show it on the, data set that we are considering. So, now if you try to see, we had this dataset.

Refer slide time :(32:07)



Frequency Distribution

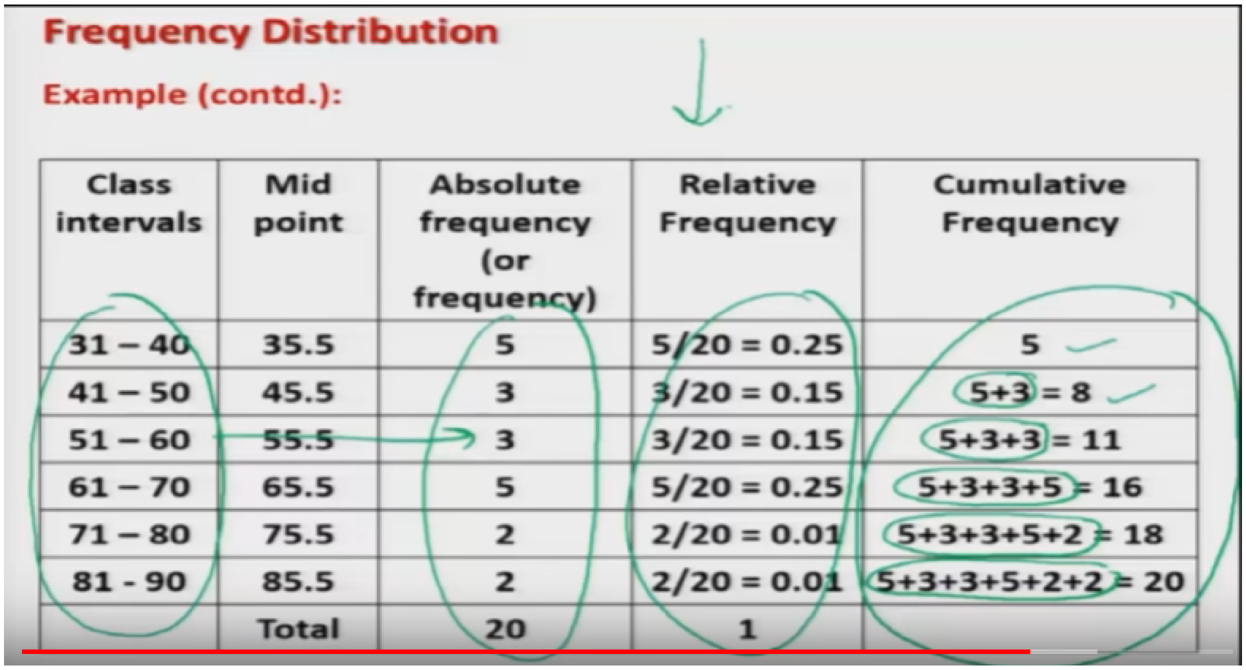
Example (contd.):

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       3       3       5       2       2
```

```
> table(time.cut)
time.cut
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
      5       3       3       5       2       2
```

That was obtained for the absolute frequencies means, I had here six intervals and then six absolute frequencies, like is 5, 3, 3, 5, 2 and here too and now, if I try to obtain the cumulative frequencies here, I had shown you in the slide.

Refer slide time :(32:36)



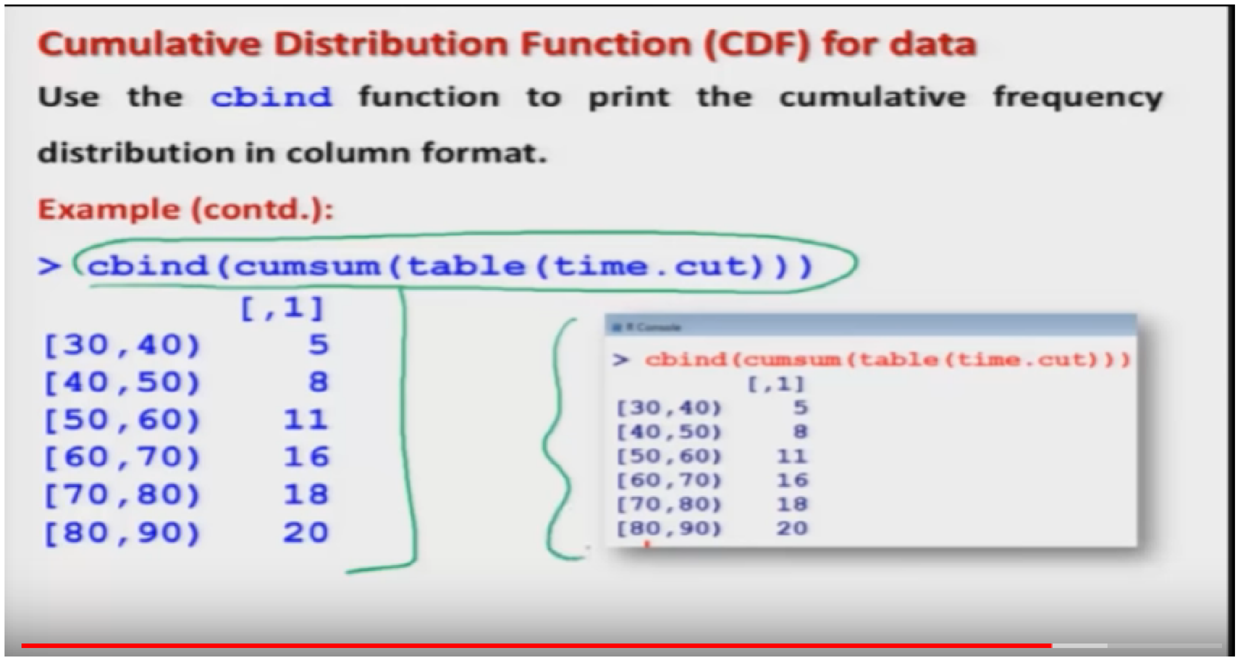
Frequency Distribution

Example (contd.):

Class intervals	Mid point	Absolute frequency (or frequency)	Relative Frequency	Cumulative Frequency
31 – 40	35.5	5	5/20 = 0.25	5
41 – 50	45.5	3	3/20 = 0.15	5+3 = 8
51 – 60	55.5	3	3/20 = 0.15	5+3+3 = 11
61 – 70	65.5	5	5/20 = 0.25	5+3+3+5 = 16
71 – 80	75.5	2	2/20 = 0.01	5+3+3+5+2 = 18
81 - 90	85.5	2	2/20 = 0.01	5+3+3+5+2+2 = 20
	Total	20	1	

In the earlier slide, that this is how we are going to find out the cumulative frequencies, first frequency that will be the sum of itself, second cumulative frequency will be the sum of the first and second frequencies third cumulative frequency, is going to be the sum of first second and third frequencies, fourth cumulative sum is going to be the sum of first four frequencies and the, fifth cumulative sum is going to be the sum of first five frequencies and the last one, which is the sixth cumulative frequency here, that is going to be the sum of all the six frequencies which is equal to the total values of the observation and you can see it that you are getting here the value 5 8 11 16 18 20 now once you are trying to obtain the outcome of Cumsum you can verify here.

Refer slide time :(33:29)



Cumulative Distribution Function (CDF) for data

Use the cbind function to print the cumulative frequency distribution in column format.

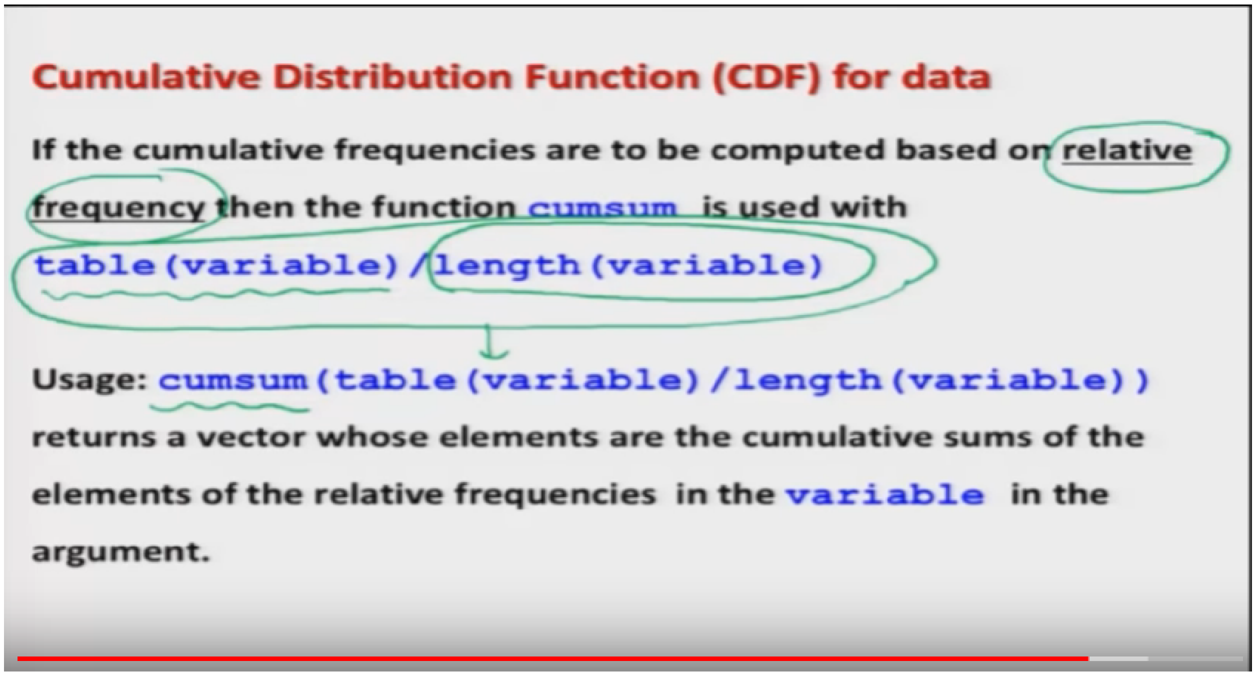
Example (contd.):

```
> cbind(cumsum(table(time.cut)))
        [,1]
[30,40)    5
[40,50)    8
[50,60)   11
[60,70)   16
[70,80)   18
[80,90)   20
```

```
> cbind(cumsum(table(time.cut)))
        [,1]
[30,40)    5
[40,50)    8
[50,60)   11
[60,70)   16
[70,80)   18
[80,90)   20
```

you are getting the same outcome here 5 8 11 16 18 20. So, this is how this cumulative frequencies are obtained, now in case if you want to find out these cumulative frequencies, with respect to relative frequencies and if you want to represent, this outcome in a vertical way you have to use the same command, in order to make this horizontal outcome into a vertical outcome just use C bind command and incase if you want to present it with respect to the relative frequencies, just try to divide this command by length of the data vector. So, let me try to show you here. So, you can see here that in this light I am simply trying to express the outcome of this light in a vertical columns by using the function C bind and here is the screen shot.

Refer slide time :(34:40)



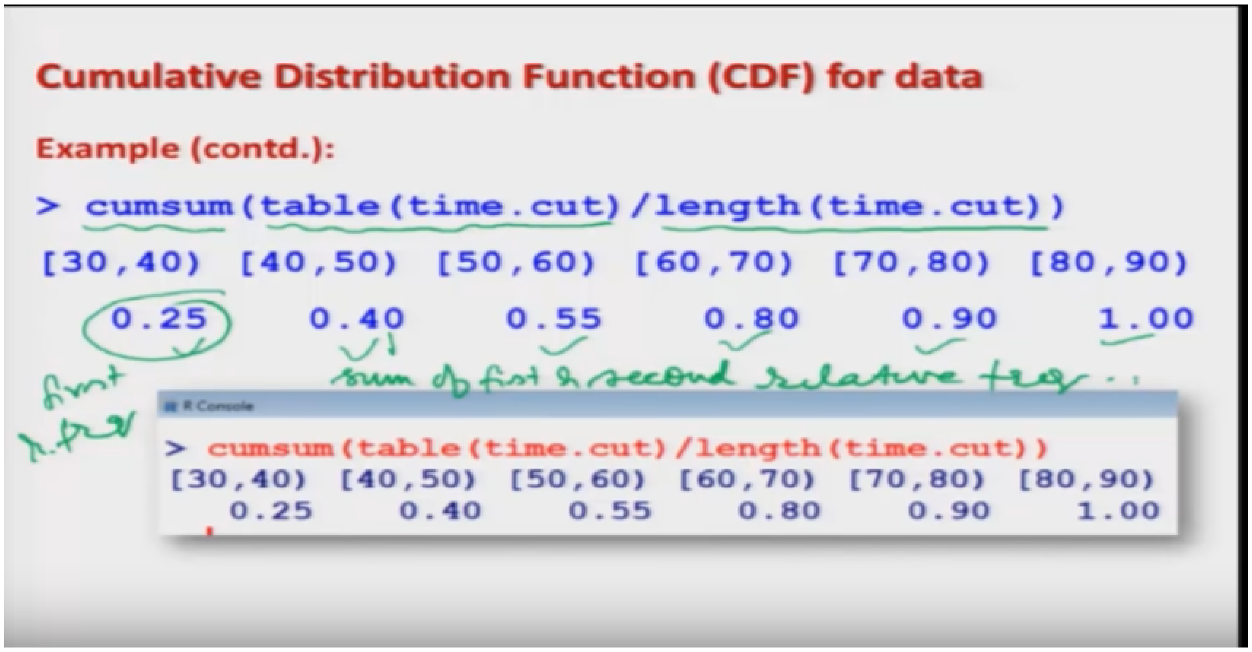
Cumulative Distribution Function (CDF) for data

If the cumulative frequencies are to be computed based on relative frequency then the function `cumsum` is used with `table(variable)/length(variable)`

Usage: `cumsum(table(variable)/length(variable))` returns a vector whose elements are the cumulative sums of the elements of the relative frequencies in the `variable` in the argument.

And similarly incase if I want to produce the same cumulative frequencies, with respect to the relative frequencies, then I simply have to use the same command, table, variable and then I have to divide it by here the function length of the variable and then use the Cumsum command over this new variable. So, using this command over the function Cumsum will produce the cumulative frequencies or the cumulative relative frequencies, of the data contained in the variable. So, again I would try to show you here.

Refer slide time :(35:12)



Cumulative Distribution Function (CDF) for data

Example (contd.):

```
> cumsum(table(time.cut)/length(time.cut))
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
   0.25    0.40    0.55    0.80    0.90    1.00
```

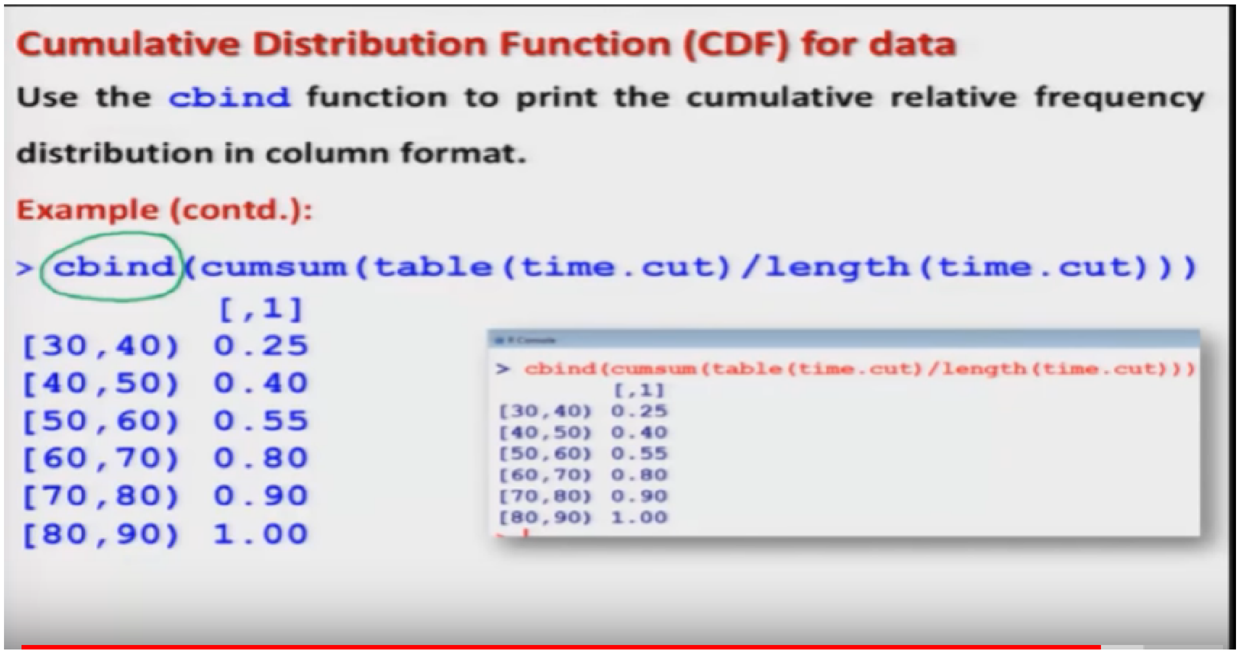
first r.freq

sum of first & second relative freq. . :

```
> cumsum(table(time.cut)/length(time.cut))
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90)
   0.25    0.40    0.55    0.80    0.90    1.00
```

That now I have operated the Cumsum on the earlier data, table time, dot cut, but now it is divided by length of the time cut. So, now this is giving me the sum of the relative frequencies. So, this is my here the first frequency, first relative frequency rather, this is the sum of first and second relative frequencies and so, on. And so on means other things are also fine and if I want to present it, in a vertical way.

Refer slide time :(35:50)



Cumulative Distribution Function (CDF) for data

Use the cbind function to print the cumulative relative frequency distribution in column format.

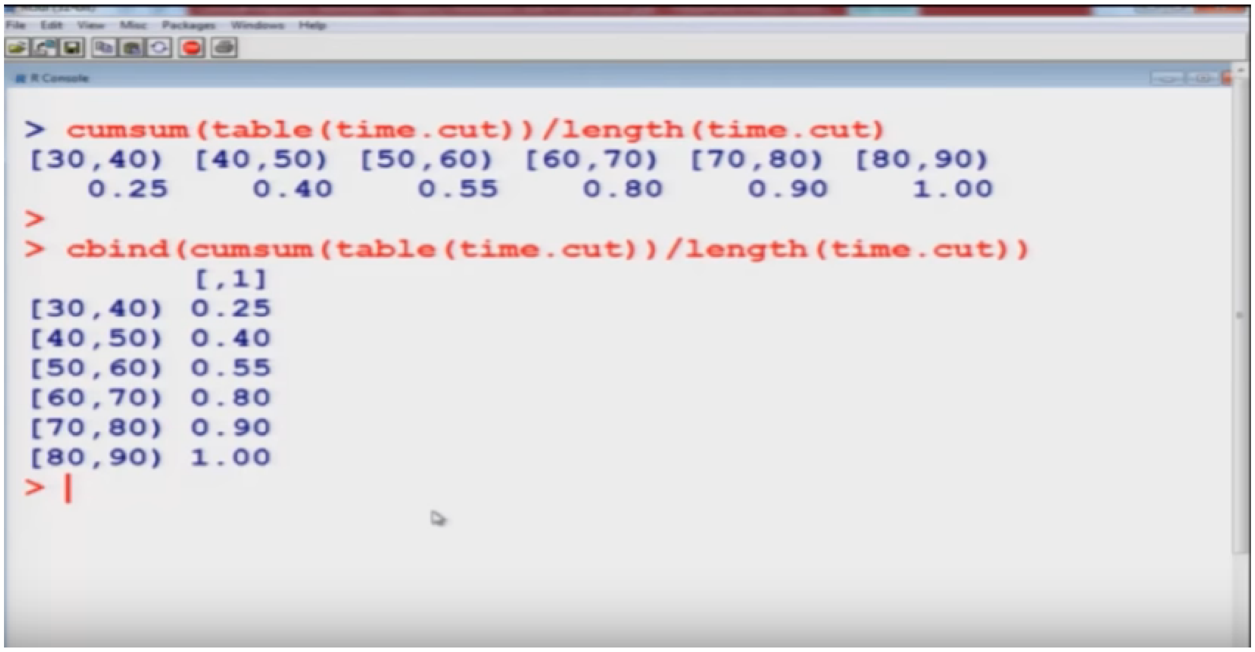
Example (contd.):

```
> cbind(cumsum(table(time.cut)/length(time.cut)))
         [,1]
[30,40) 0.25
[40,50) 0.40
[50,60) 0.55
[60,70) 0.80
[70,80) 0.90
[80,90) 1.00
```

```
> cbind(cumsum(table(time.cut)/length(time.cut)))
         [,1]
[30,40) 0.25
[40,50) 0.40
[50,60) 0.55
[60,70) 0.80
[70,80) 0.90
[80,90) 1.00
```

Then means I simply have to use, the C bind function over this. Right? So, now I would try to show you on, the R console also.

Refer slide time :(36:02)



```
> cumsum(table(time.cut))/length(time.cut)
[30,40) [40,50) [50,60) [60,70) [70,80) [80,90) 
   0.25    0.40    0.55    0.80    0.90    1.00 
> 
> cbind(cumsum(table(time.cut))/length(time.cut))
        [,1]
[30,40) 0.25
[40,50) 0.40
[50,60) 0.55
[60,70) 0.80
[70,80) 0.90
[80,90) 1.00
> |
```

So, you can see here that I have here the function or I already have obtained the data here time dot cut. Right? Now I would try to find out the, Cumsum of the table, of time dot cut and you can see here, this is the outcome and in case if you want to make it vertical, then you simply have to use here the C bind function, you can see here like this and in case if you want to obtain these cumulative sums, with respect to the relative, frequencies, means I can show you here I simply have to write down the Cumsum table with the variable time dot cut and I have to divide it by length of time dot cut, the same variable in which I destroyed that data and you can see here now this is the outcome, with respect to the, the relative frequencies, but again this outcome is in the horizontal direction and suppose if I want to, convert it into a vertical direction then I have to use the C bind function, on the same variable and executing it you can see here I'm getting the same data which was here in the horizontal way, now this is coming in a vertical columns. So, now we come to an end to this lecture and I have given you the basic idea, that how to create frequency distributions using, using the R and I would like to emphasize here one thing here more, you cannot assume this lecture or this course, which is trying to teach you pure statistics. But my objective is that that, many people are using these things. So, I want to give them a basic idea, related to the use and interpretations and I want to show them that how to do the same thing on the R software. So, my motive is very simple, definitely my this lecture cannot substitute or will not provide you the escape from saying that that without reading the books or without reading the chapters of frequency distribution from a proper book will help you. So, I would request you please try to have a look on the chapters of frequency distribution data tabulation and any good book, try to see the concept, try to learn the concept and then this lecture will help you in brushing those concept and will teach you that how to use them on the R software. So, you try to take different example from the books, from the assignment, practice it and try to see how the interpretations are being made and we will see you in the next lecture till then, Good bye.