Introduction to R Software

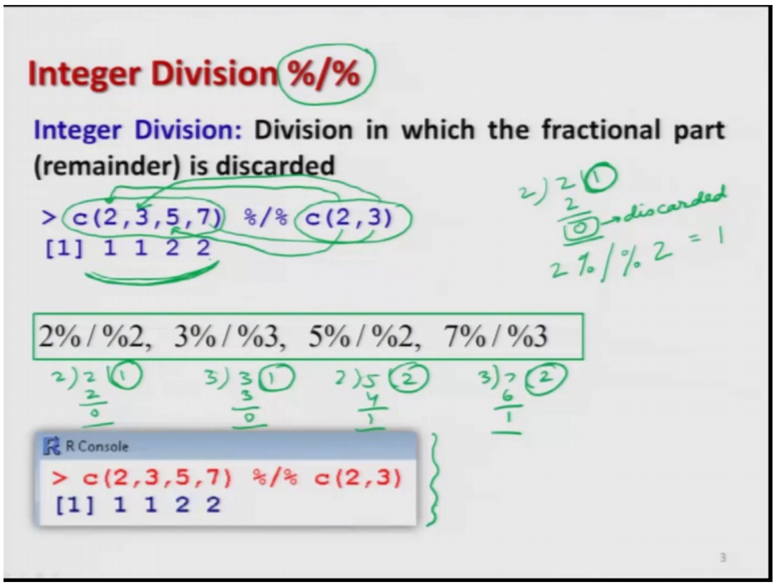
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture - 05
R as Calculator, Built in Functions and Assignment

Welcome to the next lecture on introduction to R software course. You may kindly recall that in the earlier lecture, we had discussed about some basic preliminary calculations in R and we had explained you that how one can operate the addition, subtraction as well as for the addition and subtraction for vectors; where we have more than 1 values at a time, right. We will continue with those things and we would like to explore some more aspect of basic calculation where I can use this R just like a simple calculator.

So, now, we will try to discuss about the integer division in R. The first question comes; what is an integer division. So, the division in which the fractional part which is the remainder that is discarded that is called as integer division.

(Refer Slide Time: 01:09)



Integer Division %/%

Integer Division: Division in which the fractional part (remainder) is discarded

```
> c(2,3,5,7) %/% c(2,3)
[1] 1 1 2 2
```

2%/%2, 3%/%3, 5%/%2, 7%/%3

```
> c(2,3,5,7) %/% c(2,3)
[1] 1 1 2 2
```

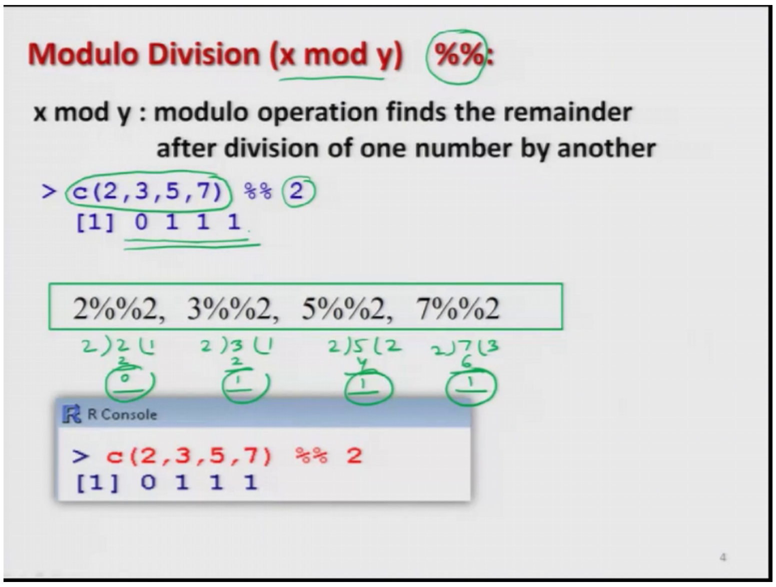
discarded
2 %/% 2 = 1

For example, in case, if I want to divide 2 by here 2, then we can see that here the answer will be divisor will be 1 and 2 1 za 2 and the remainder will be here 0, right.

So, in this case when I say that I am trying to do the integer division of 2 with 2, then the answer will come out to be 1 which is the divisor and this remainder part is discarded, right. The symbol for doing the integer division in R is percentage sign back slash percentage sign. So, in case if I want to do the integer division of 2 by 2, then I have to write here 2 percentage back slash percentage and 2 and this will give us the answer to be equal to here 1. So, now, I try to consider combined values of 2, 3, 5 and 7 in this vector and I would like to do the integer division with respect to another vector which is containing 2 values 2 and 3. So, now, there are 2 things that you have to observe first thing how this integer division has been made this; I already have explained you the second aspect; what you have to see that this integer division is organized in this way, this 2 is divided by this 2, this 3 is divided by this 3 and once again this set of values, they are operated over the values 5 and 7 and so 2 is.

So, 5 is integer divided by 2 and 7 is integer divided by 3. So, in this case, you can see now here thus 2 is being divided by 2 with respect to integer division answer comes out to be 2 1 za 2 and the remainder here is 0. Similarly here the remainder comes out to be 0 and similarly here the remainder comes out to be here 1 and similarly here the remainder comes out to be here 1. So, now, you can see here because the integer part of the division which is here 1, 1, 2 and here 2 and these 4 values are reported over here 1, 1, 2, 2 and the screen shot of this thing is given over here which I would request all of you to do it with your own hand, right.

(Refer Slide Time: 03:56)



Modulo Division (x mod y) %%:

x mod y : modulo operation finds the remainder after division of one number by another

```
> c(2,3,5,7) %% 2
[1] 0 1 1 1
```

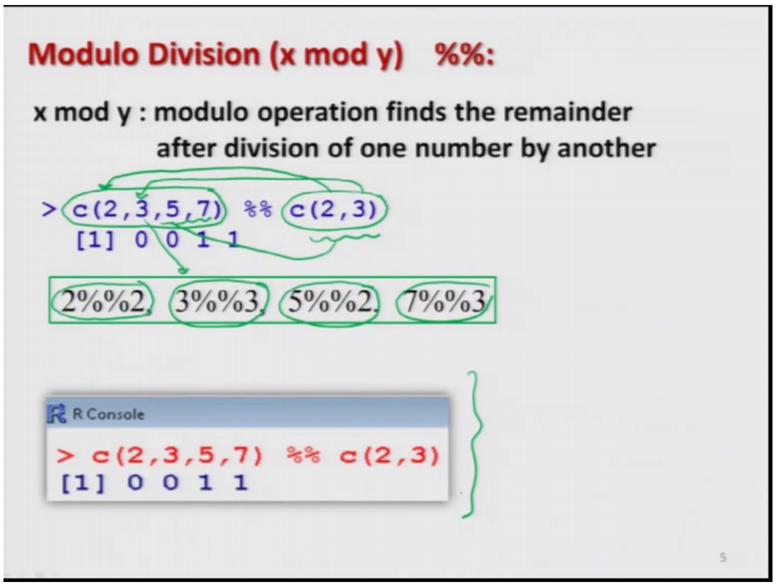
2%%2, 3%%2, 5%%2, 7%%2

```
> c(2,3,5,7) %% 2
[1] 0 1 1 1
```

Similarly, when we try to go for modulo division and in mathematics that is written that even I am trying to divide x by y with respect to modulo division, we try to write it with x mod y and notation in R is double percentage sign; what really happens in the modulo division actually modulo division finds out the remainder after division. So, now, I will try to take an example where I have taken 4 values 2, 3, 5 and 7 which are combined in a vector and I would like to operate the modulo division with respect to here two. So, in this case what you have to observe that each of this values 2, 3, 5 and 7; this is being a modulo divided by 2. So, I am trying to do it here in this way say 2 divided by here 2 and answer comes out to be 2 1 za 2 and remainder here is 0 and similarly 2 divided by 3 answer comes out to be 2 1 za 2 remainder here is one similarly if I divided by 2; 2 2 za 4, remainder comes out to be here 1 and 7 divided by 2 answer comes out to be 2 3 za 6; remainder here is 1 and now this 0; this 1, this 1 and this 1 which are the remainder which are they are reported here as 0, 1, 1, 1.

So, this is the procedure for the modulo division, right and when you try to do it over R means that I can just show you here for the sake of understanding.

(Refer Slide Time: 05:39)



Modulo Division (x mod y) %%:

x mod y : modulo operation finds the remainder after division of one number by another

```
> c(2,3,5,7) %% c(2,3)
[1] 0 0 1 1
```

2%%2 3%%3 5%%2 7%%3

R Console

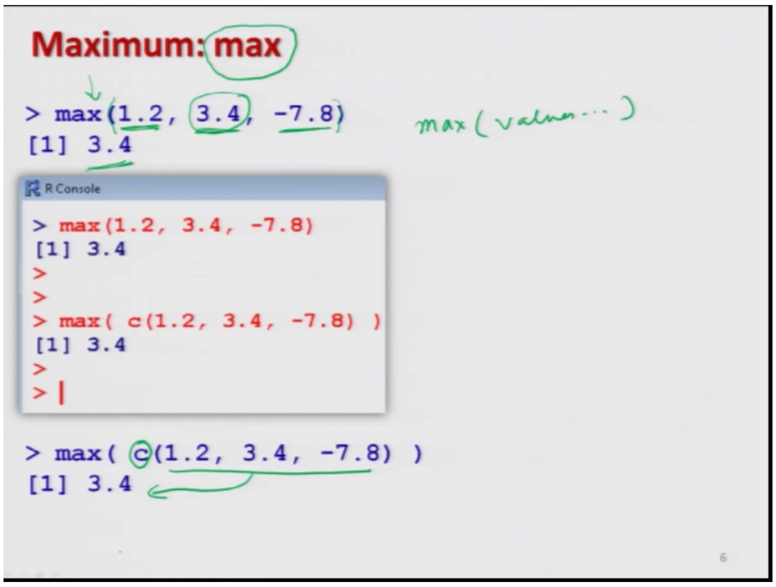
```
> c(2,3,5,7) %% c(2,3)
[1] 0 0 1 1
```

You can see here; this is the same answer and this is the slide which is being copying and pasted over here, right. So, this is how we tried to operate the modulo division and similarly now in this case in say in say another example I would try to do the modulo division of a vector with respect to a vector. So, here I am trying to take 4 possible values

2, 3, 5 and 7 which are combined by a vector and similarly, I am trying to consider another vector and there are 2 values 2 and 3 which are combined by a vector containing 2 and 3.

So, what will now happen; this 2 will divide this 2 and this 3 will divide this 3 and in the next step, the same set of values 2 and 3; they are going to operate with respect to 5 and 7. So, what will happen here that 2 is being divided by 2 with respect to the modulo division or we call it say 2 modulo 2 and similarly, this 3 is being modulo divided by here 3 and the 5 is being modulo divided by here 2 and 7 is being modulo divided by here 3 and the screen shot of this operation in R is given over here once again I would like to request you to operate it with your own hand on your computer.

(Refer Slide Time: 07:13)



Maximum: max

```
> max(1.2, 3.4, -7.8)
[1] 3.4
```

max(value ...)

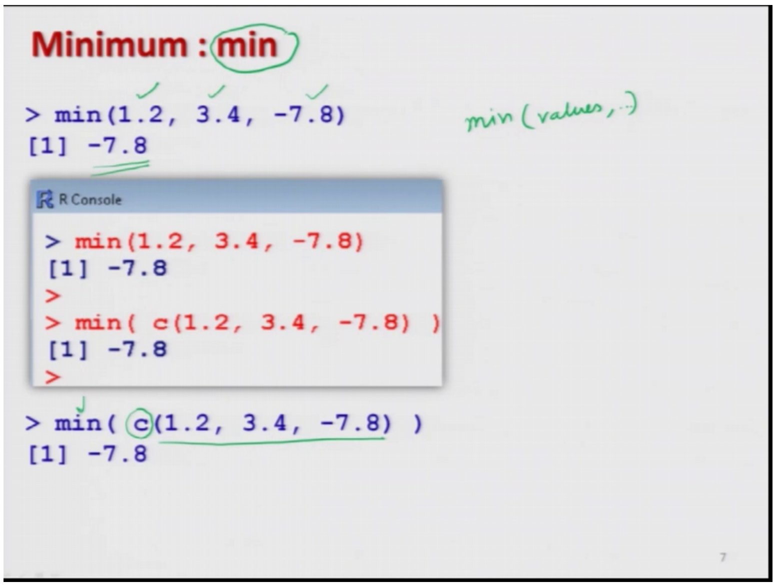
```
R Console
> max(1.2, 3.4, -7.8)
[1] 3.4
>
>
> max( c(1.2, 3.4, -7.8) )
[1] 3.4
>
> |
```

```
> max( c(1.2, 3.4, -7.8) )
[1] 3.4
```

Now, there are some built in function in R which can give us the direct answer; for example, suppose I want to find out the maximum value among certain given number of values then the function here is max. So, m a x is the function which is already built in inside the R package and as soon as you try to find out the maximum of several values using this operator m a x; it will finally, give us the maximum among those value.

So, I will try to take here one example and try to show you here first example; I am taking here 3 values 1.2, 3.4 and say minus 7.8, right and when I try to operate the function m a x here. Then I try to write all these values inside this bracket means maximum and inside the bracket all the values, they will be written separated by a comma and when I try to operate it, you can see here out of 1.2, 3.4 and minus 7.8; 3.4 value which is the maximum among all the 3 given values. So, the answer comes out to be here 3.4 and similarly, another approach to the same thing is this that I can combine all the same value 1.2, 3.4 and minus 7.8 by this combined operator and then I try to find out the maximum value this will also give us the same answer.

(Refer Slide Time: 09:01)



Minimum : min

```
> min(1.2, 3.4, -7.8)
[1] -7.8
```

min (values, ...)

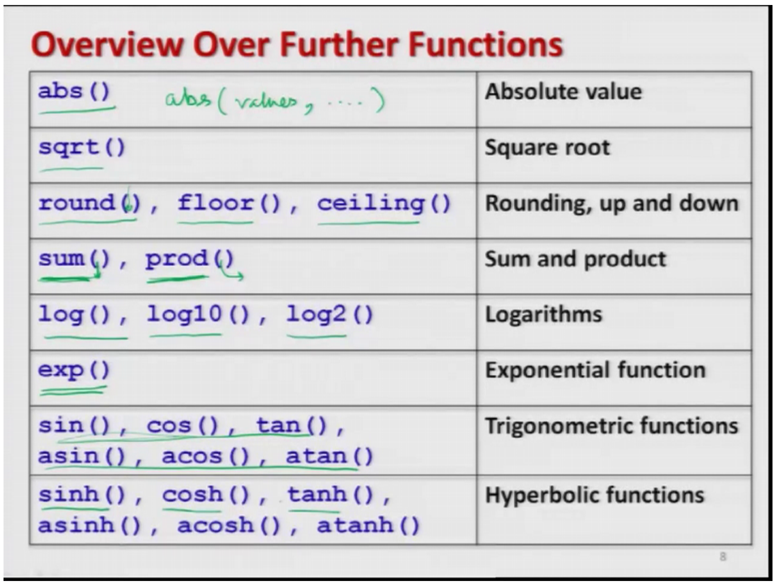
```
> min(1.2, 3.4, -7.8)
[1] -7.8
>
> min( c(1.2, 3.4, -7.8) )
[1] -7.8
>
```

```
> min( c(1.2, 3.4, -7.8) )
[1] -7.8
```

So, just for the sake of illustration, I try to show you here that how we can obtain it. So, for the sake of understanding I tried to take this command and I try to operate it here I try to paste it here and as soon as I press enter I get the value here 3.4 right. Similarly, we can also find out the minimum in this case the command is m i n; we which is the short form of minimum and the rule here is again the same just as we did in the case of maximum that I would try to write down the minimum and inside the bracket I will try to write all the values which are separated by comma.

So, here you can see I have considered the 3 values 1.2, 3.4 and minus 7.8 and when I try to do the minimum of these 3 value the answer comes out to be minus 7.8 and the same thing I can also do it in the say alternative way that I can combine all the 3 values with this combined operator c and I can write all these values inside this bracket and then I try to use the function m i n. So, minimum is again another function which is built in inside the base package of R and it can and you can directly obtain the minimum value from some the given set of values, right.

(Refer Slide Time: 10:27)



Overview Over Further Functions

abs() abs(values, ...)	Absolute value
sqrt()	Square root
round(), floor(), ceiling()	Rounding, up and down
sum(), prod()	Sum and product
log(), log10(), log2()	Logarithms
exp()	Exponential function
sin(), cos(), tan(), asin(), acos(), atan()	Trigonometric functions
sinh(), cosh(), tanh(), asinh(), acosh(), atanh()	Hyperbolic functions

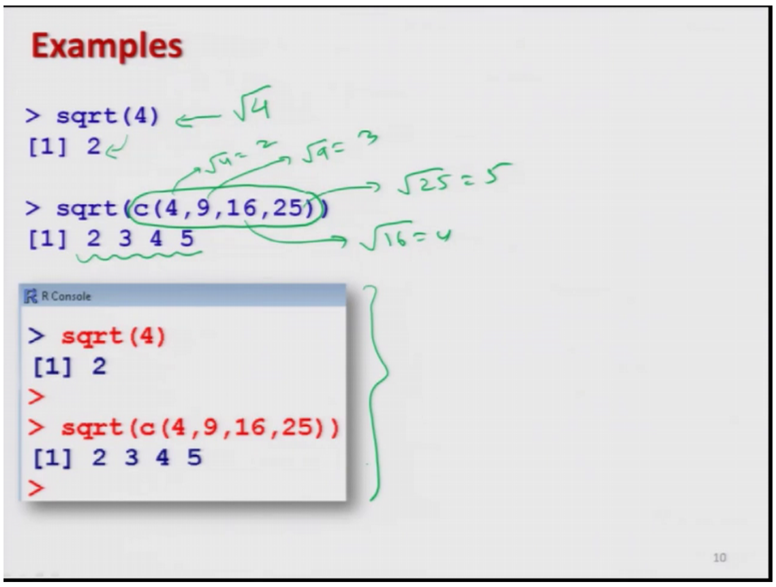
Similar to maxima maximum and minimum values there are some other values which are already given in the base package right. For example if I want to find out the absolute values then the function is a b s and inside the bracket and pattern here is the same that we try to write down the a b s function and then all the values separated by comma and similarly in case if I want to find out square root the function is s q r t and again all the values are written inside the bracket and yeah in the next slide, I will try to take some examples to show you that how these things are being operated.

And similarly, for rounding of a value, we have a function round and again, if I have more than one values I can put all the values inside this bracket separated by a comma similarly in case if I want to one value up and one value lower, then we have function here floor and ceiling. And similarly if I want to find out the sum of several values, I can use the built in function here s u m sum and again inside this bracket, I have to give all the values and the outcome will be the sum of all the values which are prescribed inside the bracket.

The next command is product; suppose I want to find out the product of several values. So, I have to write down all the values inside the bracket separated by a comma and then I have to use this operator p r o d and this will directly give us the product of those values, right, similarly log values log with base ten and log with base 2 and so on; all these values are built in inside the R package and depending on your need you can use them directly without writing an ad additional program.

Similarly, in case if you want to find out the exponential of certain function, I can use the function e x p and for some trigonometric function like they are sin function cos function tan function a sin a cos a tan function. They can be operated in a similar way and hyperbolic trigonometric functions are also they are like is sin h cos h tan h and a sin h a cos h a tan h all these function you can see here they can built in inside the base package of R and whenever you need it you can use it.

(Refer Slide Time: 12:56)



Examples

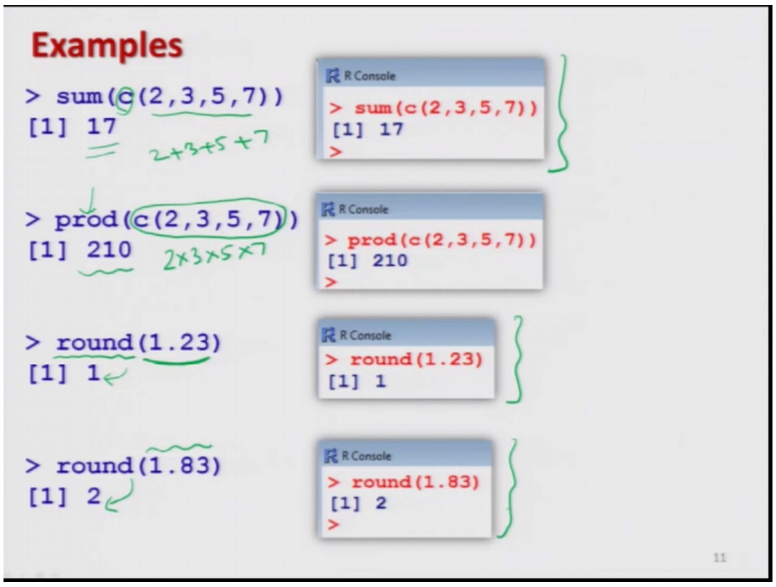
```
> sqrt(4)
[1] 2

> sqrt(c(4,9,16,25))
[1] 2 3 4 5
```

```
> sqrt(4)
[1] 2
>
> sqrt(c(4,9,16,25))
[1] 2 3 4 5
>
```

Now, let me demonstrate some of the functions that how one can use it, for example, if I want to find out the absolute value you can see here I am trying to take here a negative value minus 4 and I am trying to operate this a b s function. So, if I try to write down a b s function and inside the bracket values then the answer comes out to be four; obviously, the absolute value of minus 4 is 4 and similarly, if I have more than one values then for simplify I try to take 5 values minus 1 minus 2 minus 3, 4 and 5 and suppose 3 values here are negative values and if I try to operate the a b s function the absolute function over this vector, the answer comes out to be 1, 2, 3, 4 and 5 like this why because the absolute value of minus 2 is 2 absolute value of minus 3 is 3 and a absolute value of 4 is 4 and a absolute value of 5 is 5.

So, you can see here that the absolute function operates over the each and every element in your combined vector and this is here the screen output of the same operation. So, you can try it yourself right similarly if I want to find out the square root we already have a function s q r t; that means, the square root. So, suppose I want to find out the square root of here four. So, this can be written like here as this s q r t and here four. So, the answer comes out to be here 2 and similarly, in case if I want to find out the square root of more than one values then all those values, they have to be combined inside a vector and then this square root function is operated. So, you can see here the square root of 4 is here 2 square root of 9 is 3 square root of 16 is 4 and square root 25 is 5 and this is the answer which is obtained over here and the screen shot of the same operation when you try to do it in R and some more example.

(Refer Slide Time: 14:59)



Examples

```
> sum(c(2,3,5,7))
[1] 17
```

2+3+5+7

```
> prod(c(2,3,5,7))
[1] 210
```

2×3×5×7

```
> round(1.23)
[1] 1
```

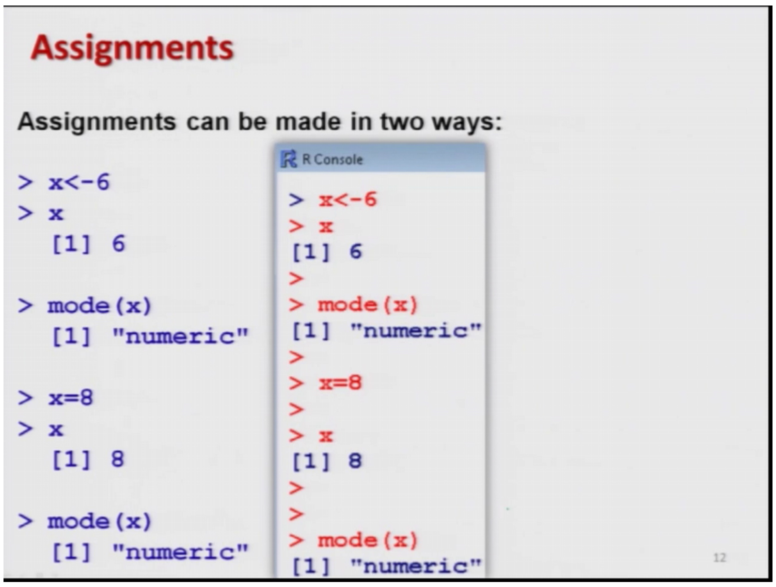
```
> round(1.83)
[1] 2
```

Suppose, I want to find out the sum of 4 values 2, 3, 5 and 7 that is I want to find out 2 plus 3 plus 5 plus 7, then in that case I can combine all these values inside a vector using the c command and then I will have to use the function sum. So, sum of all these values will come out be here seventeen and you can see here this is the screen shot, right. Similarly incase if I want to find out the product of 2, 3, 5 and 7; that means, I want to know the value of 2 into 3 into 5 into 7. So, in that case I simply try to club or combine all the values inside a vector using the c command and I try to operate the function p r o d; that means, product and 2 into 3 into 5 into 7 this comes out to be 210.

Now, suppose I want to find out the round of the value 1.23. So, I try to use here the function round and inside the bracket I will try to write down the values 1.23. So, we know that the rounded value of a 1.23 is 1. So, as soon as you write down this command in R and press enter you will get here the value one and this is here the screen shot and similarly if you want to find out the round of value of one point eight 3 then definitely it is more than 1.5. So, it is going to be 2 and if you try to do this command in R you get an answer 2 and this is the screen shot of this function.

So, you can see here all these functions sum product round of they can be directly operated over the R without any problem, right.

(Refer Slide Time: 16:45)



Assignments

Assignments can be made in two ways:

```
> x<-6
> x
  [1] 6

> mode(x)
  [1] "numeric"

> x=8
> x
  [1] 8

> mode(x)
  [1] "numeric"
```

R Console

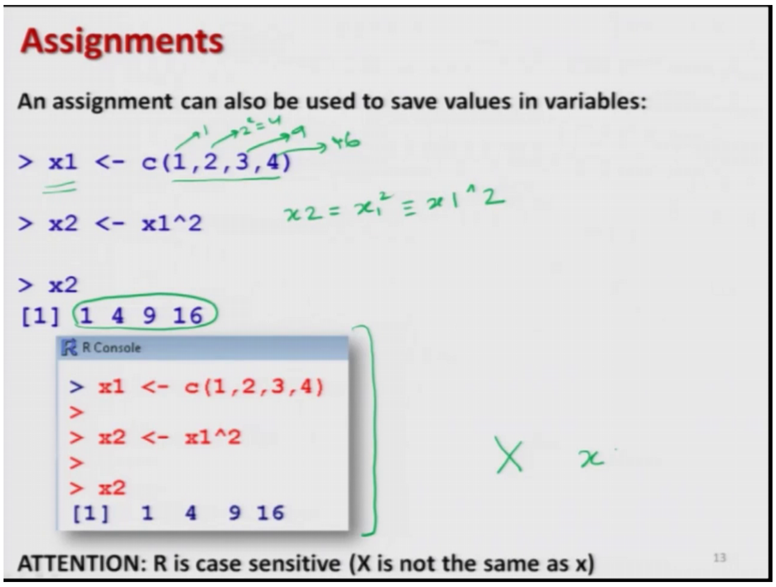
```
> x<-6
> x
[1] 6
>
> mode(x)
[1] "numeric"
>
> x=8
>
> x
[1] 8
>
>
> mode(x)
[1] "numeric"
```

12

Now I come to another aspect; what is about assignments; assignments, we already have discussed that; how do we assign a value to a variable. For example, if you preamble earlier we had discussed that if I define variable x and if I write 2; that means, I have assigned the value 2 to a random variable x. So, the same thing here try to do it here by writing x less than hyphen 6 or I can also use here directly here x is equal to 6 and then I try to check the value here this gives me here the value of x here 6. Now I would like to show you something else that suppose I want to find out what is the mode of x 1 conscious, I am not finding out the mode which is a statistical function in your in the topics of mean median and mode, but here I am trying to find out the mode of x meaning there by what is the nature of x whether it is numeric or not.

So, as soon as you try to do here mode of x you see that the answer comes out to be here is numeric for example, I will try to show you here in the R function itself suppose if I try to write down here x equal to 6 and then I press here x this is giving me a value 6 and now if I try to write down here the mode of x then this is here numeric. So, this is trying to indicate that x equal to 6 is a numerical value. So, let us now come back to our slides and the next thing similarly I can operate.

(Refer Slide Time: 18:30)



Assignments

An assignment can also be used to save values in variables:

```
> x1 <- c(1,2,3,4)

> x2 <- x1^2

> x2
[1] 1 4 9 16
```

$1 \quad 2^2=4 \quad 9 \quad 16$

$x2 = x_1^2 \equiv x1\hat{}2$

R Console

```
> x1 <- c(1,2,3,4)
>
> x2 <- x1^2
>
> x2
[1]  1  4  9 16
```

$X \quad x$

ATTENTION: R is case sensitive (X is not the same as x)

13

This thing over a vector quantity also, right, suppose I try to define here a vector x one which is combining 4 values 1, 2, 3 and 4 and then I try to assign a different value that I am going to define here another value here x 2 which is x 1 square.

So, I try to write down here as say here x 1 hat 2 so; obviously, the square of 1 is 1 the square of 2 is 4 the square of 3 is 9 and the square of here 4 is here 16. So, you can see here that now 1, 4, 9 and 16 are coming over here; that means, this x 2 has been defined over a vector. So, the moral of the story is this; I can use this assignment fu function over a scalar quantity having only one value as well as I can use it over a vector quantity having more than 1 values and here you can see here that this is the screen shot of the same operation being done in R and here yeah you have to keep in mind that as I told earlier that that R is case sensitive means capital X and small x they are different they are not the same. So, that you have to keep in mind over here, right.

So, with this slide I would like to stop here and I would like to request all of you that you please try to practice all these assignments taking different types of example and whatever example I have taken, try to take some other values, try to make them more complicated, try to take some question from the assignment; some question from say other books and try to practice it; till then good bye.