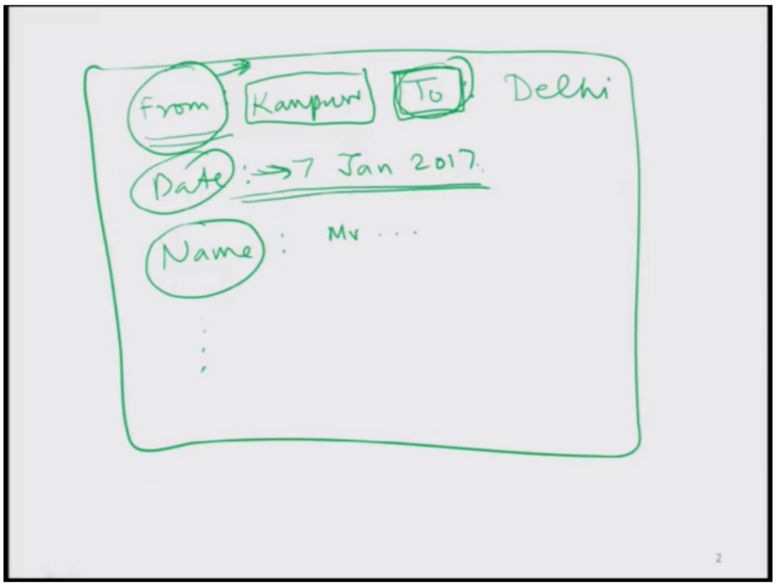
Introduction to R Software
Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology, Kanpur

Lecture – 24
Print and Format Function

Welcome to the next lecture on the course Introduction to R Software. From this lecture and in the next couple of lectures, we are going to talk about the display and formatting of the outcomes. So, the first question comes why should we do it, and what is the utility of such functions which are useful in displaying the output.

Now, I will ask you a simple question have you ever seen the train ticket, in that train ticket they already have printed something like this.

(Refer Slide Time: 00:54)



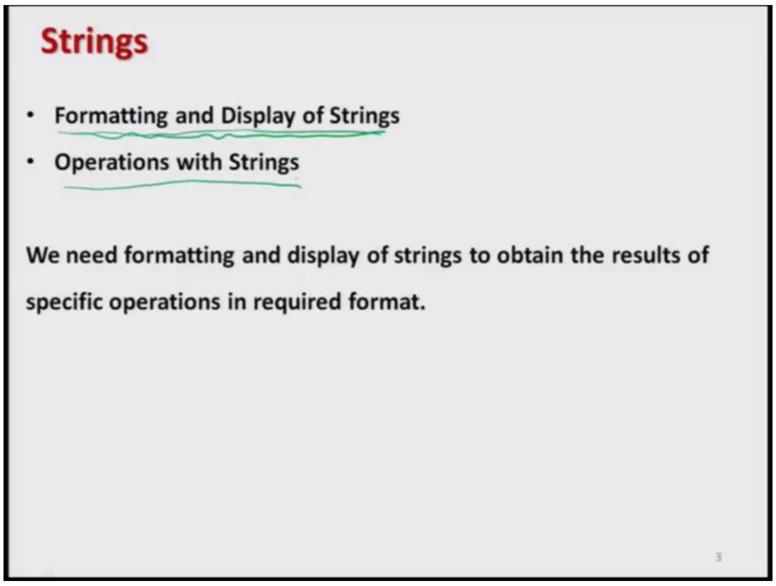
Suppose this is my train ticket, they already have printed something like from then to then here date and so on, and send name of the passenger and so on, and some other information. This is already there in that ticket, and whenever someone goes to the ticket counter, they will ask the requirement that where do you want to go on what date, what time, and the name of the passenger, and they will simply enter it in their computer and finally, they will take a printout. And when you see the printout it looks like, at the place from they are say giving the station name say here Kanpur. Suppose if you want to travel from Kanpur to say Delhi, and to they have entered the string Delhi and it is coming were there, then date whatever they enter, this comes over here say 7th of say this January 2017, and name whatever is the name of the person say mister whatever it is.

So, you can see here there are 2 types of entries 1 type of entries which are already there from to date name etcetera, and second type of entries are which other person at the ticket counter is doing, but what is the requirement; the requirement is that finally, whatever input has been given that has to be printed at a particular place. For example, wherever is the field for from, here only does the starting station has to be printed, wherever they already have printed to their the station of destination should only be printed, this cannot happen that if somebody wants to travel from Kanpur to Delhi and it is printed on the ticket from Delhi to Kanpur, that will be misleading. And then whatever the date is taken by the ticket person that has to be entered only at this place, after the date column; that means, I need to print my input at particular places or in simple words I need the outcome in a certain format.

Now, you have seen that earlier we have taken different types of output, and they come in the a special way which is predefine inside the R software, but now what we want is that we need the output in a particular format according to our wish our requirement our need.

So, here now we are going to discuss about different options available in R software to take the output in terms of display on the screen, or printing on a printer in different ways, and how we can arrange the output in a certain given format ok.

(Refer Slide Time: 04:34)



Strings

- Formatting and Display of Strings
- Operations with Strings

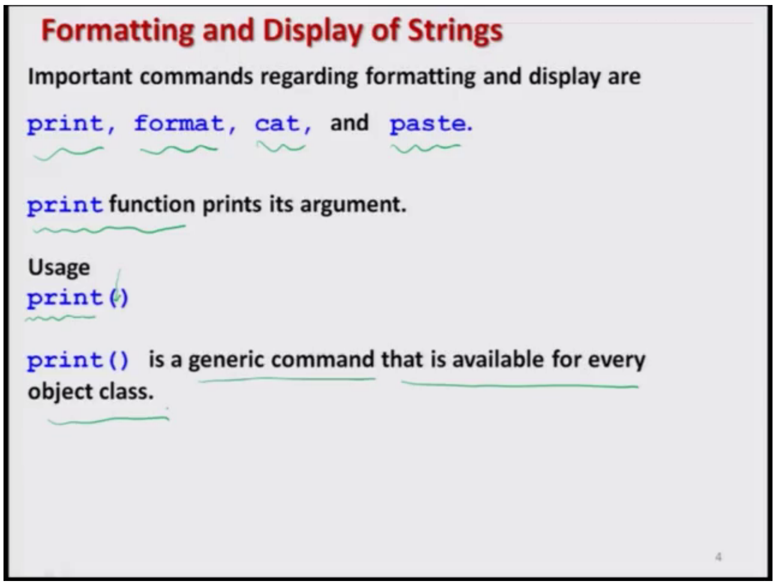
We need formatting and display of strings to obtain the results of specific operations in required format.

So, when we come to the display and formatting option in R we have 2 options. First of all we want to format and display; that means, if there is a string or a number whatever it is, because whatever we are entering that has to be taken only as a character, I just want to print whatever, I am entering; if I am entering, 10th of Jan 2017, then it should be print as 10 J a n; Jan 2017, and when I am entering 10th January; j a n u a r y 2 0 1 7, then it has to be printed as 10 January 2017. So, everything is taken as there as a say string or a character.

So, now we have 2 option that how to format this and display, these a strings what we want, and second thing is this how to do operation with different types of string for example, we can have a particular type of format where, I want to print certain strings in a particular way. So, for example, if there are 3 students the names of the students can be printed student 1, student 2, student 3 in a sequence in a row or say column wise, student 1, then student 2, and then student 3.

So, we are going to discuss all those aspects over here, and for that we have here several commands, which are used for getting the final outcome in different ways depending on the needs of the situations. 1 command is print second is format third is cat and another we are going to discuss is paste.

(Refer Slide Time: 06:12)



Formatting and Display of Strings

Important commands regarding formatting and display are

`print`, `format`, `cat`, and `paste`.

`print` function prints its argument.

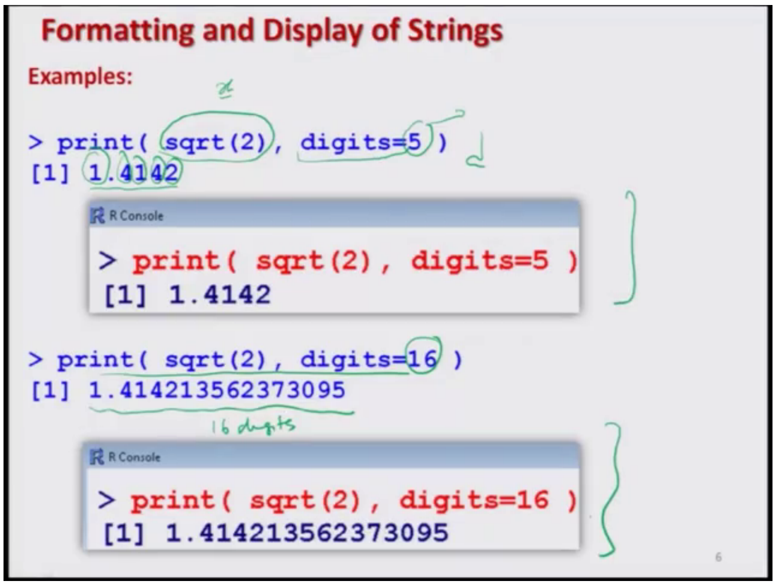
Usage

`print()`

`print()` is a generic command that is available for every object class.

So, first of all let us try to understand, how to use the function print, print as it, indicates the meaning by its words simple to print, whatever I want to print I will simply print. In order to use this print function, I have to write print that is all in small letters and whatever I want to print that is a indicated inside the arguments inside this brackets. And the advantage of using print function is this that print is a generic command and which is available for all sorts of object classes. If you remember we had discussed different types of object classes like its character number; number data frame and everything list, so print can be use in all the classes without any problem.

(Refer Slide Time: 07:18)



Formatting and Display of Strings

Examples:

```
> print( sqrt(2), digits=5 )
[1] 1.4142
```

R Console

```
> print( sqrt(2), digits=5 )
[1] 1.4142
```

```
> print( sqrt(2), digits=16 )
[1] 1.414213562373095
```

16 digits

R Console

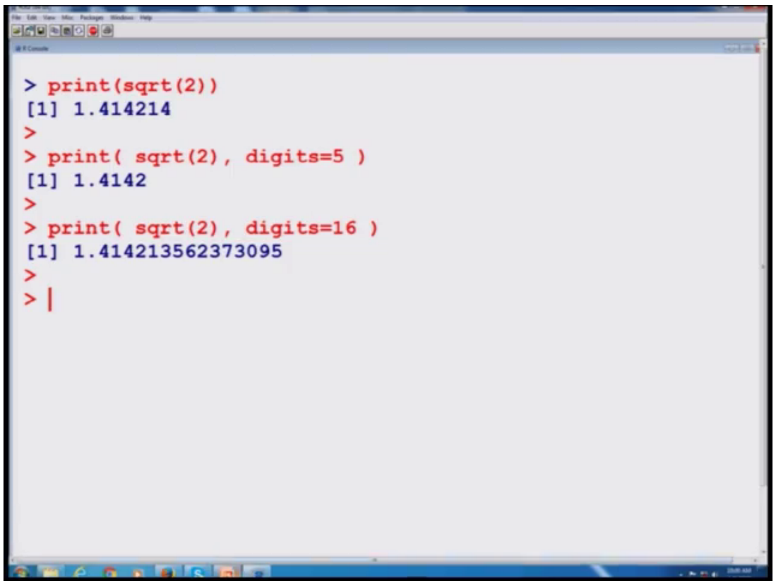
```
> print( sqrt(2), digits=16 )
[1] 1.414213562373095
```

Now let us try to take some example, and we try to understand how to get it done for example, if I want to print the square root of 2, then I can simply write down here print and inside this arguments I can say sqrt and 2 inside the bracket, which is the usual command for finding out a square root of a function, and the answer comes out here 1.414214 and so on. And you will see that this R outcome will give you certain number of digits, and these digits are automatically decided by the R.

Now there is another option that I can also control this number of digits after the decimal point. So, here this is the screenshot I will try to show you that how it comes, but before that let us try to understand that suppose I want to find out the square root of 2 in such a way such that there are total 5 digits inside the outcome. So, I can do like this print, then whatever I want to print here this vector say here x, and now I have to use this command d i g i t s digits all in small letter equal to what were the number of digits I want to have, suppose I want to have here 5 number of digits in the outcome. So, you can see here this will give you on the R console 1.4142, digit number 1, digit number 2, digit number 3 4 and digit number 5, so there are all together 5 digits.

Similarly, in case if I want to have, supposed say this 16 digits. So I can do the same function here, print a square root of 2 and here, I say digit equal to 16, and you can see here the outcome is like this. It has altogether 16 digits, and these are the say this outcome on the R console, but before that we will try to do it, on the R console and let us write to see what happens. First you look at the slide to print say this is square root of 2 you can see here this comes out to be like this.

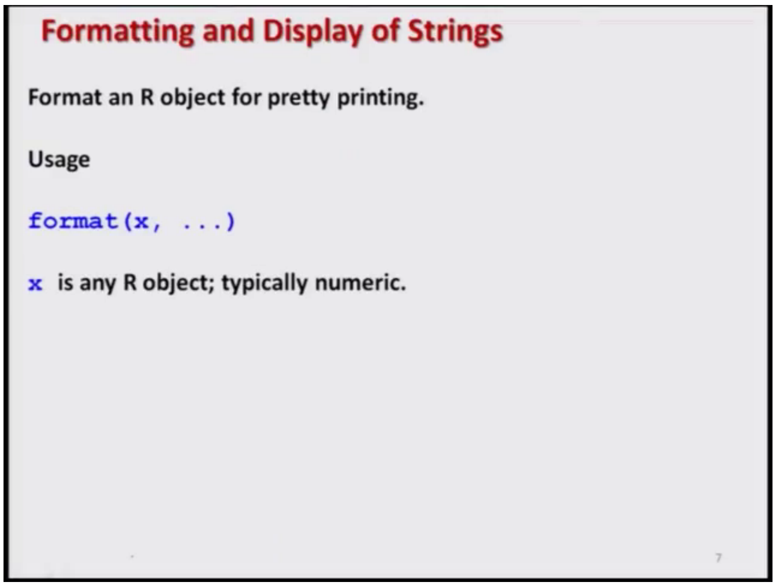
(Refer Slide Time: 09:45)



```
> print(sqrt(2))
[1] 1.414214
>
> print( sqrt(2), digits=5 )
[1] 1.4142
>
> print( sqrt(2), digits=16 )
[1] 1.414213562373095
>
> |
```

Now, we try to print the square root of 2 with 5 digits, this comes out to be you can see the same outcome. And similarly I try to print say 16 digits you can see here we get this outcome, and the same screenshot has been given here also. Now this print function is giving us an output in a format which is already defined inside the R package, but now I need to format it.

(Refer Slide Time: 10:20)



Formatting and Display of Strings

Format an R object for pretty printing.

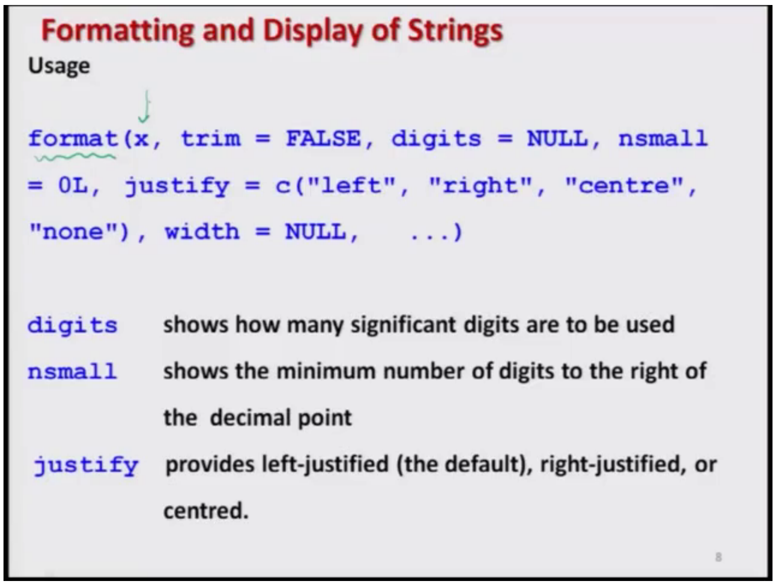
Usage

```
format(x, ...)
```

`x` is any R object; typically numeric.

So, in order to format any outcome we have a different command, what is called as f o r m a t format, and inside the argument inside the brackets we give the values which we want to format in a particular way. And this format is essentially used for a nice printing for the pretty printing or separate ear did display in the required format, and here this x can be any of the R object, but many times we are interested only in the numerical values.

(Refer Slide Time: 11:11)



Formatting and Display of Strings

Usage

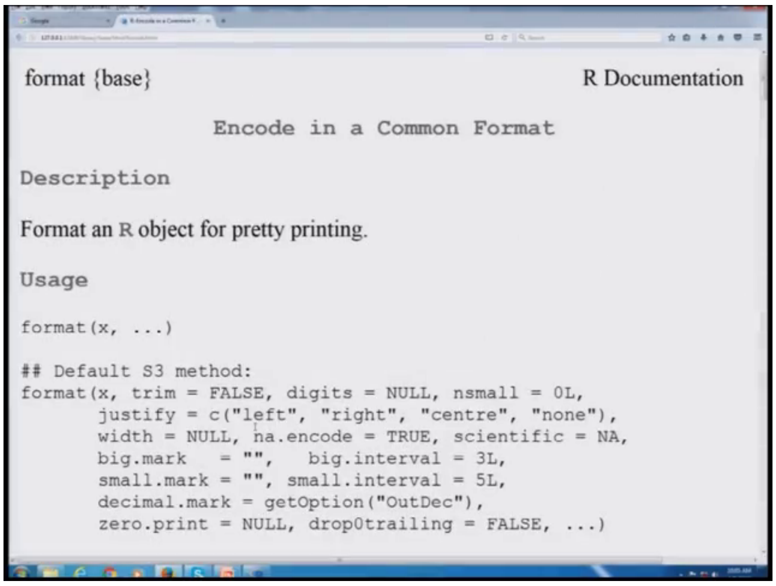
```
format(x, trim = FALSE, digits = NULL, nsmall
= 0L, justify = c("left", "right", "centre",
"none"), width = NULL,    ...)
```

`digits`	shows how many significant digits are to be used
`nsmall`	shows the minimum number of digits to the right of the decimal point
`justify`	provides left-justified (the default), right-justified, or centred.

But anyway the more detailed syntax of the format command is as follows that we try to write down here the format, inside the argument we try to give different types of options. First thing is this whatever you want to format that is contained in the variable x, then we have 1 option here trim, if you want to trim the outcome next option is here digits, and it shows that or you can control that how many significant digits asked to be used, that depends on you have to specify it the default here is null and it uses the default number whatever is available in the R package.

Then another option is here and is small, and small shows that or you can control the minimum number of digits, which are to be given on the right set of the decimal point. And similarly you have here justified whether you want the sintering to be on the left hand side right hand side or in the center, or you do not want any this centering also any position. And similarly here you have a some more arguments over here, and means as usual what I say that whenever you want more details on this format, I will always recommend that you go to the help command and try to look it over the website of this R, you can see here this comes out to be here R right.

(Refer Slide Time: 12:44)


format {base}	R Documentation

Encode in a Common Format

Description

Format an R object for pretty printing.

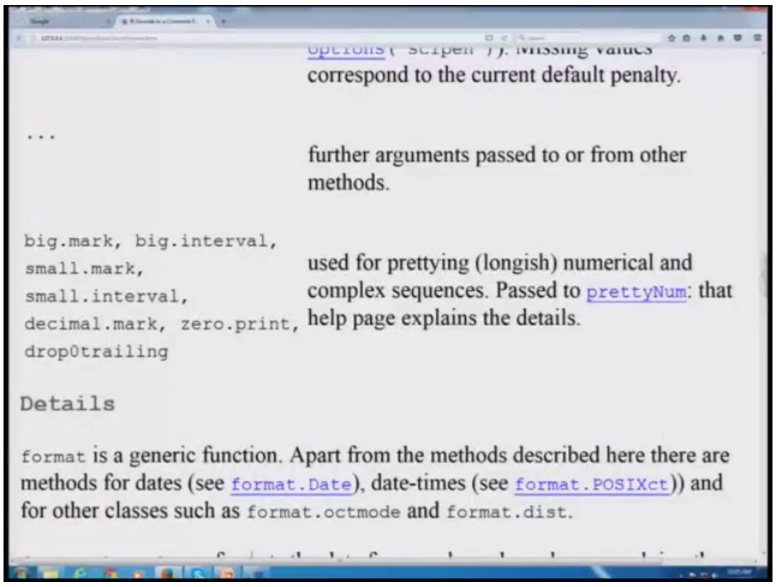
Usage

```
format(x, ...)

## Default S3 method:
format(x, trim = FALSE, digits = NULL, nsmall = 0L,
       justify = c("left", "right", "centre", "none"),
       width = NULL, na.encode = TRUE, scientific = NA,
       big.mark   = "",   big.interval = 3L,
       small.mark = "", small.interval = 5L,
       decimal.mark = getOption("OutDec"),
       zero.print = NULL, drop0trailing = FALSE, ...)
```

It is giving you here each and everything, we cannot discuss your each and everything, but definitely you can also do it on your computer right ok.

(Refer Slide Time: 13:01)


options("scipen")). Missing values correspond to the current default penalty.

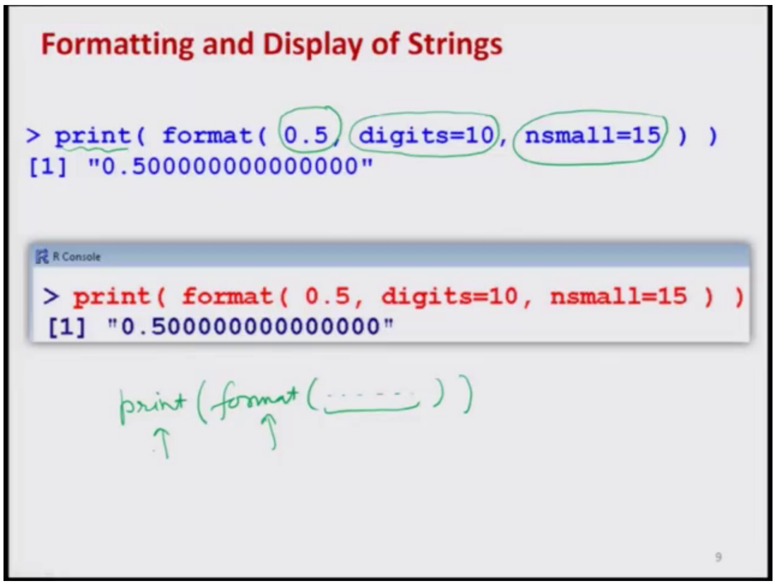
...	further arguments passed to or from other methods.
big.mark, big.interval, small.mark, small.interval, decimal.mark, zero.print, drop0trailing	used for prettying (longish) numerical and complex sequences. Passed to prettyNum: that help page explains the details.

Details

format is a generic function. Apart from the methods described here there are methods for dates (see format.Date), date-times (see format.POSIXct)) and for other classes such as format.octmode and format.dist.

So, let us now come back to the slides.

(Refer Slide Time: 13:09)



Formatting and Display of Strings

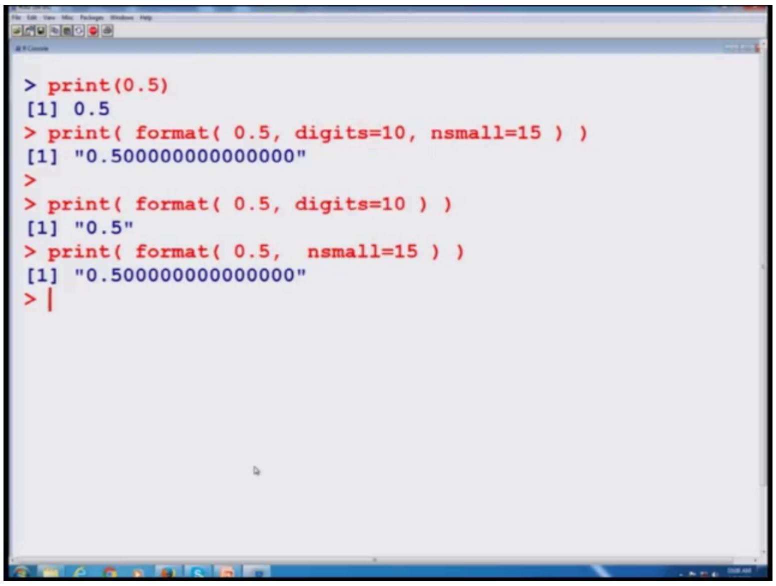
```
> print( format( 0.5, digits=10, nsmall=15 ) )
[1] "0.500000000000000"
```

```
> print( format( 0.5, digits=10, nsmall=15 ) )
[1] "0.500000000000000"
```

And now let us try to take some example; some simple example to understand that, how this print and format can be combined together to get any particular type of outcome. For example, here I am trying to say here print, what print 0.5, but now I am saying that this has to be printed in a particular format, and my format is this that I need total number of digits to be here 10, and the number of any small values to be here 15.

So, if you try to see here how I am trying to write down this function say print, and inside this argument inside this bracket, I have to a specify what I want to print in what format, so here I will try to write down here format, and then whatever I want to print in a particular format I can a specify here. So first this function is executed, and then the spring function is executed and we get the outcome in a required format.

(Refer Slide Time: 14:29)

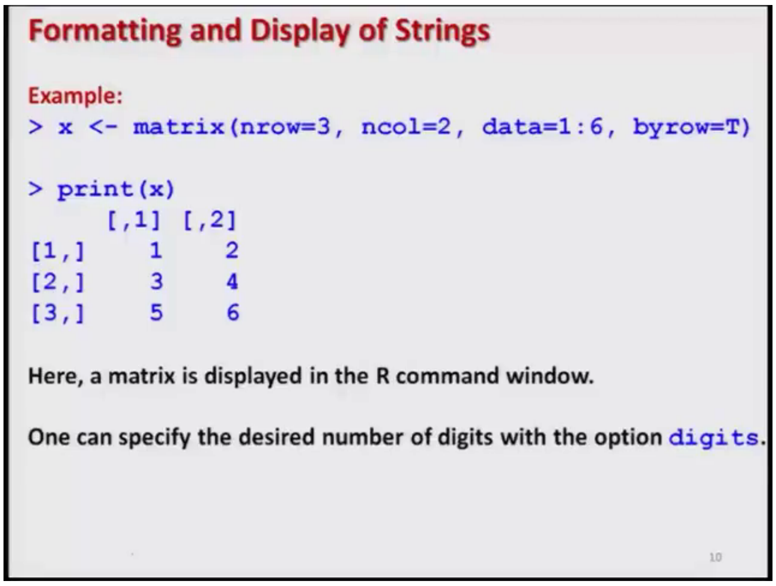


```
> print(0.5)
[1] 0.5
> print( format( 0.5, digits=10, nsmall=15 ) )
[1] "0.500000000000000"
>
> print( format( 0.5, digits=10 ) )
[1] "0.5"
> print( format( 0.5,  nsmall=15 ) )
[1] "0.500000000000000"
> |
```

For example if I try to take this example over here, you can see here what happens for example if I say here I simply want to print here is 0.5 what do we get here, it is simply 0.5, but if I try to give it here see here with the format command, where the number of digits are 10 and number of any spawns are 15, we get this command over here. And suppose even if I remove this option which is which is an optional you see what do we get over here right. And then in case if I try to remove these digits and only I gave here and small you can see here what do we get out here.

So, this is precisely actually what you have to do that take; take a function, try to experiment with all combination, and try to see what happens and then try to understand what is the role of this function, and this is how you can understand any argument or say any function or any command in a much better way ok.

(Refer Slide Time: 15:35)



Formatting and Display of Strings

Example:

```
> x <- matrix(nrow=3, ncol=2, data=1:6, byrow=T)

> print(x)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

Here, a matrix is displayed in the R command window.

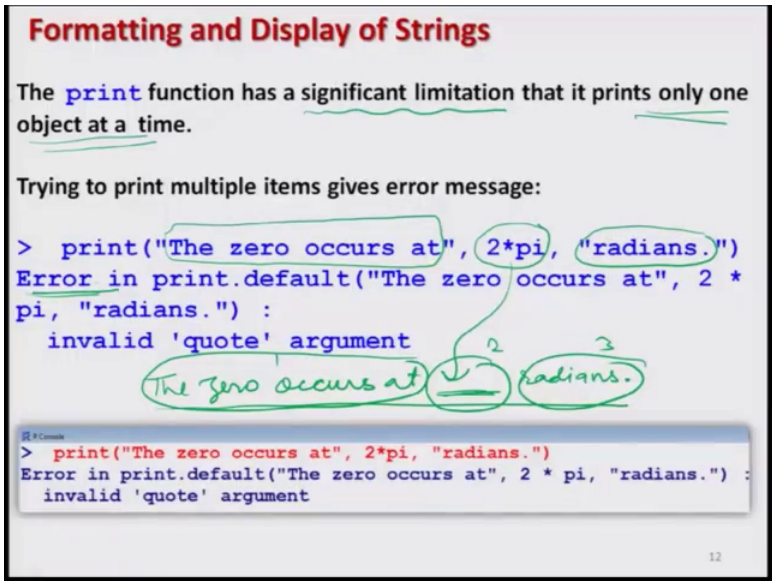
One can specify the desired number of digits with the option `digits`.

10

Now, let us come back to our slides now, I try to take another example and where I am trying to use the print command over a class matrix, as we had discussed this. This print command can be used with any of the object class.

So, I try to define here a matrix here like this, which is a 3 by 2 matrix with 3 rows 2 columns and data is from 1 to 6 which is arranged row wise, and if I try to give here print x this is printing the outcome of the x right. And you 1 can also specify the desired number of digits with the option digits depending on your requirement. Let us try to see whether this matrix is printed over R console or not concision print right, and now you can see here that is the see screenshot of the same outcome.

(Refer Slide Time: 16:52)



Formatting and Display of Strings

The print function has a significant limitation that it prints only one object at a time.

Trying to print multiple items gives error message:

```
> print("The zero occurs at", 2*pi, "radians.")
Error in print.default("The zero occurs at", 2 *
pi, "radians.") :
  invalid 'quote' argument
```

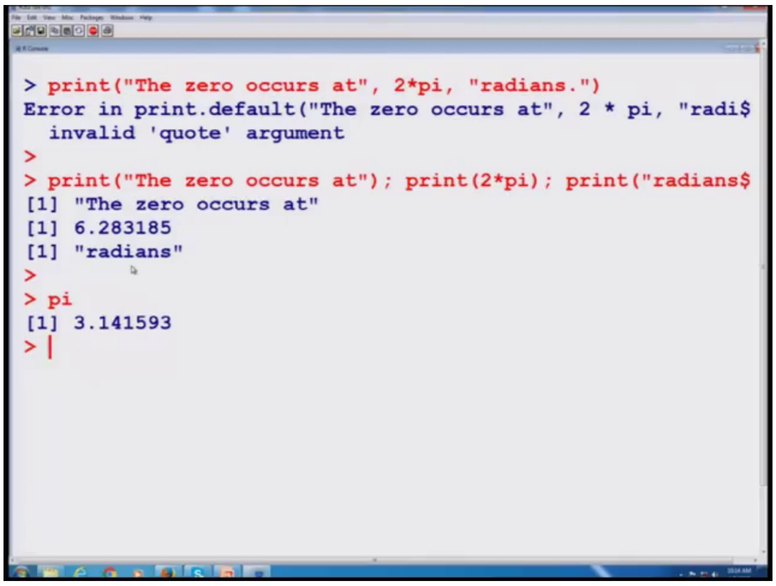
The zero occurs at
2
3
radians.

```
> print("The zero occurs at", 2*pi, "radians.")
Error in print.default("The zero occurs at", 2 * pi, "radians.") :
  invalid 'quote' argument
```

Now, something more about this print function and some of the problems, and then the answer comes how to solve them. Brain function has a very significant limitation that that it prints only 1 object at a time this means what, so let us try to take an example and we try to see what I want. Suppose I want to print the outcome in the particular way, that first I want to print the 0 occurs at as such that is a string, this whole sentence should appear exactly in the same way the 0 occurs at and whatever is the value that is here a variable 2 into pi; pi, is a given value inside the R package, and then after that it should happen here radians.

So, I need here the value that the 0 occurs at this some blank radiance and fully stopped, and here this value of 2 pi will be inserted I want to write in the same line; that means, 3 things in a sequence 1 2 and here 3 they should come in the same sequence 1 2 and 3 order, but now if you try to do so, you get an error over here, let us try to see what happens you get here an error.

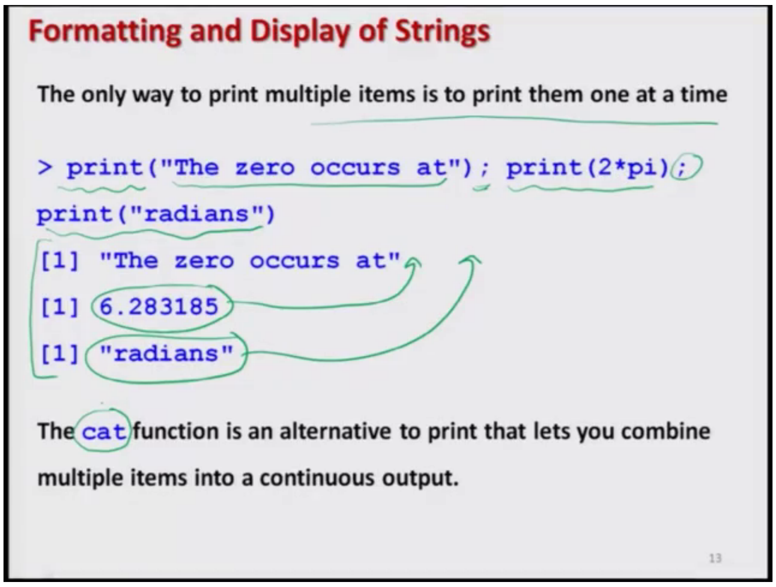
(Refer Slide Time: 18:46)



```
> print("The zero occurs at", 2*pi, "radians.")
Error in print.default("The zero occurs at", 2 * pi, "radi$
  invalid 'quote' argument
>
> print("The zero occurs at"); print(2*pi); print("radians$
[1] "The zero occurs at"
[1] 6.283185
[1] "radians"
>
> pi
[1] 3.141593
>
```

There is something wrong so; that means, print is not working, when we want to print more than 1 object at the same time.

(Refer Slide Time: 19:03)



Formatting and Display of Strings

The only way to print multiple items is to print them one at a time

```
> print("The zero occurs at"); print(2*pi);
print("radians")
[1] "The zero occurs at"
[1] 6.283185
[1] "radians"
```

The cat function is an alternative to print that lets you combine multiple items into a continuous output.

13

Now on the other hand, means if you really want to print it we have an option. The option is that print multiple items only 1 at a time, still that is not giving us what we really want I want to print all the things in the sequence , now if I try to use my print command, and if I say print the 0 occurs at then separated by the semicolon, and then I write print to pi, and then separated by a semicolon then print radiance, then it will give me this type of outcome, but it still that is not needed, means I want this thing to be here and this radiance to be here after this.

So now, how to get it done for that we have another function that is called a c a t cat function, but we will talk about the cat function in the next lecture, but before then let us try to see what happens with this type of print command. So, if I try to print all the 3 things what I wanted to print in a sequence, but separately you can see here that it is printing like this, so in the first case this was giving me error and 1 thing you can see here what is the hair pi; pi, has a value 3.14 that we know, right.

So I would like to stop here. And I would request you to experiment with this print function and format function try to define your own objectives and try to see whether you are getting the display and printing in the required format or not. And we will talk about more options in the next lecture, till then goodbye.