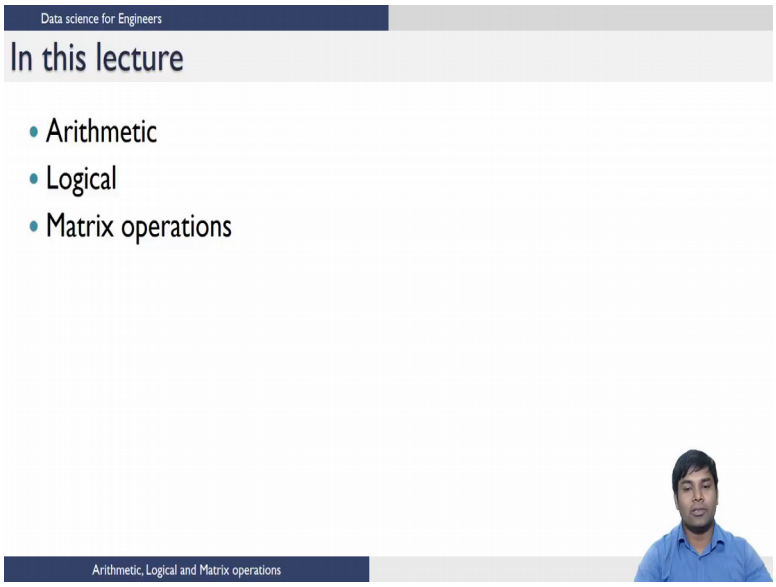
Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 07
Arithmetic, Logical and Matrix operations in R

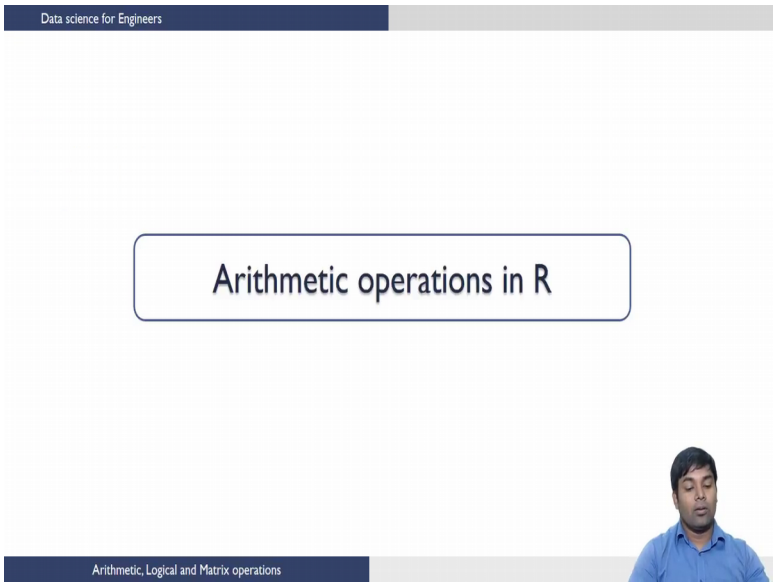
Welcome to lecture 6 of the R module in the course Data Science for Engineers. In the previous lectures we have seen various data types of R, how to access R delete the elements of the different data types and so on. Now, it is time to see how to perform arithmetic, logical and matrix operations in R.

(Refer Slide Time: 00:42)



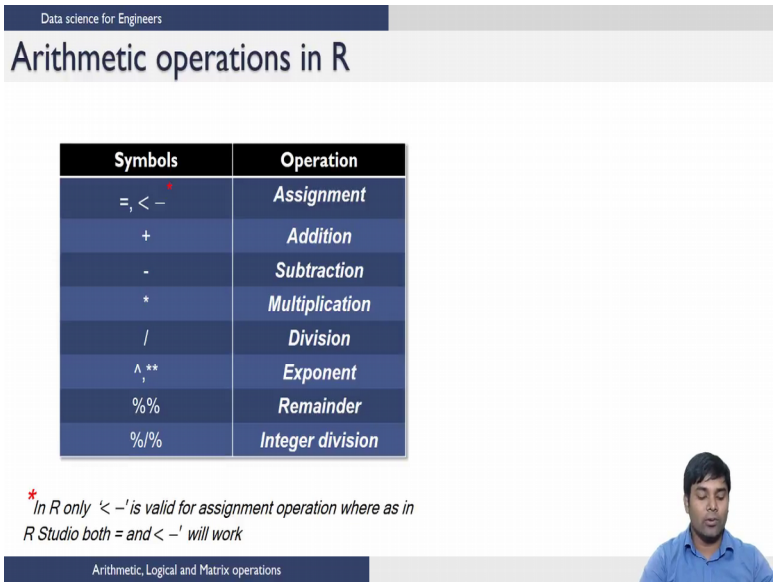
In this lecture we are going to see how to do arithmetic operations, logical operations and matrix operations in R.

(Refer Slide Time: 00:50)



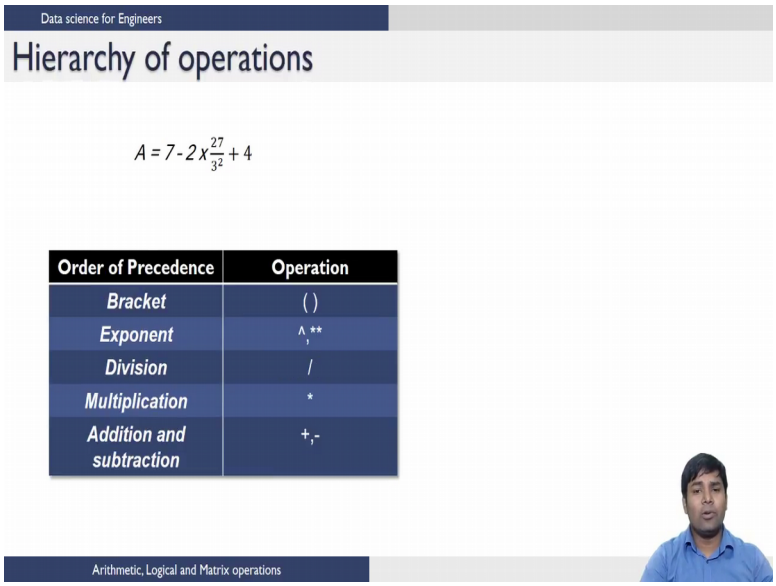
So, let us first look at the arithmetic operations.

(Refer Slide Time: 00:54)



R supports all the basic arithmetic operation, the first one is assignment operator. You can use either is equal to or the back arrow to assign a value to be variable and standard addition, subtraction, multiplication, division, integer division and remainder operations are also available in R. In R back arrow is only the valid assignment operator whereas, as an R studio both is equal to and back arrow R proper assignment operators.

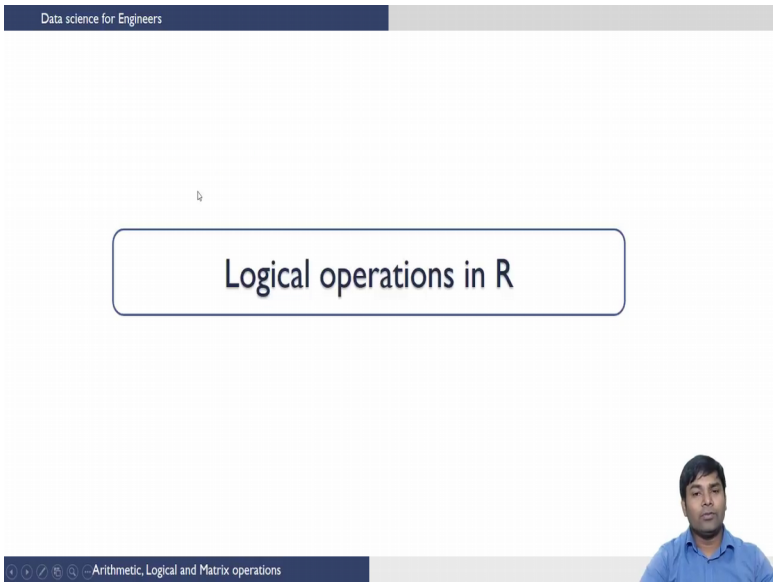
(Refer Slide Time: 01:26)



Let us look at the hierarchy of operations while performing the arithmetic operations in R. So, it is similar to our normal Broadmarsh rule with bracket has the first importance exponent has the second priority and followed by division, multiplication, addition and subtraction. For your understanding you can type in this expression and then see what is the value of a would be if you want to understand the order of precedence first we do not have any brackets in here.

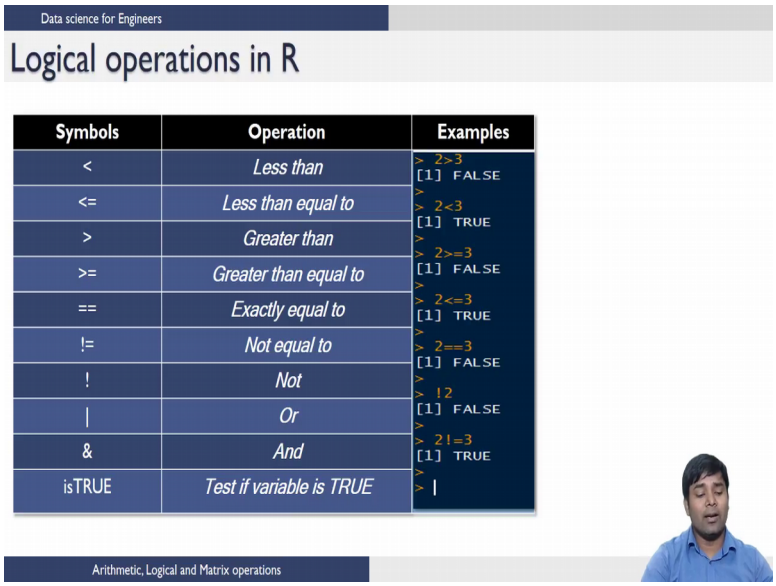
The next one is exponent the first this part 3 square will be evaluated that is 9 and the next operation is division 27 by 9 will give you 3, 3 times 2 is 6 because the next operation is multiplication. So, once you have 6 here what is the next operation? Addition 6 is minus 6 because you have minus 1 here 7 plus 4 is 11, minus 6 and which gives you value of A as 5.

(Refer Slide Time: 02:35)



Next we move on to the logical operations in R. So, we have standard logical operations such as less than less than or is equal to, great then, greater than equal to, equal to and so on.

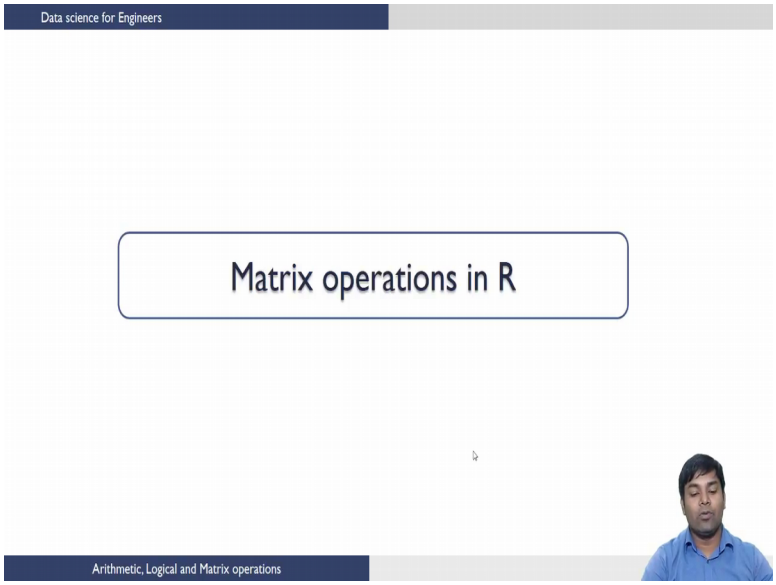
(Refer Slide Time: 02:39)



There are examples where you can see if you ask 2 is greater than 3 it will true a value false because this statement to greater than 3 is not true. Similarly if you say 2 is equal to 3 it will also say false because 2 is not equal to 3. When you execute this command 2 not

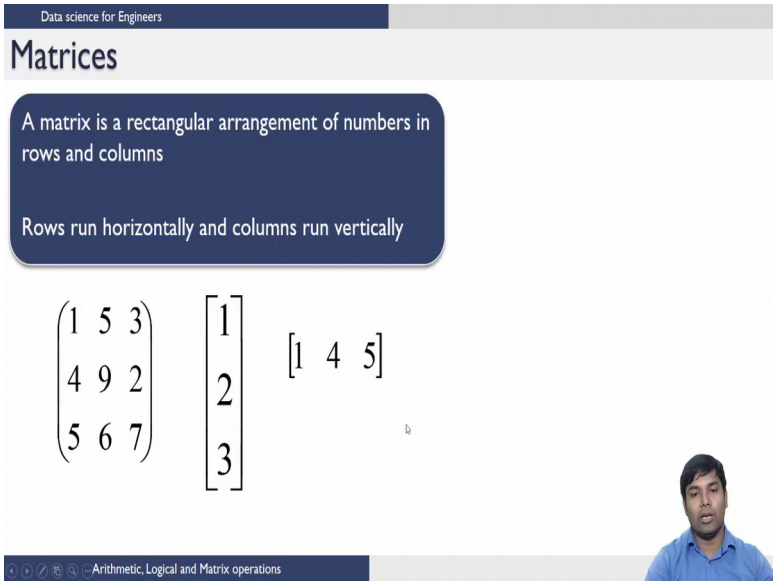
equal to 3 it will give answer as true because 2 is not equal to 3. So, this is the summary of logical operations that can be performed in R.

(Refer Slide Time: 03:21)



Next we move to the important class of operations that are needed for data analysis problems. Most of the data we will treat them as matrices. So, matrix operations play a key R important role by solving the data analysis problems.

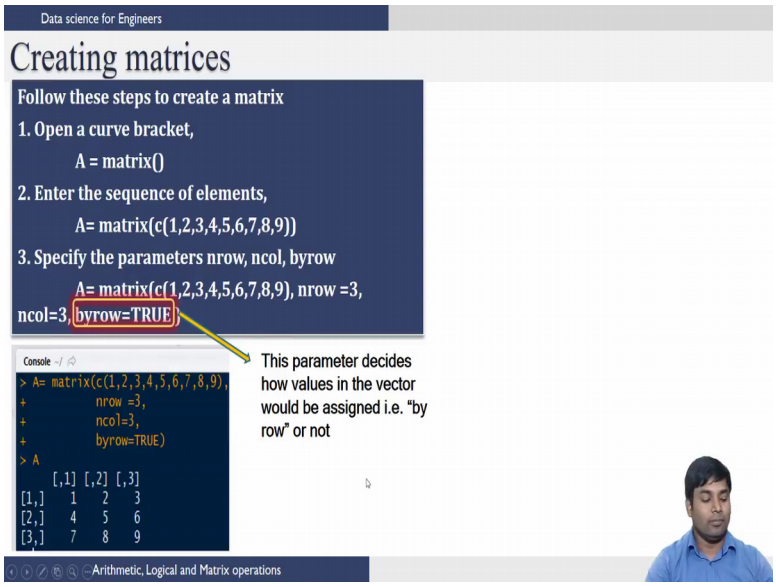
(Refer Slide Time: 03:38)



Let us first define what matrices are. A matrix is a rectangular arrangement of numbers in rows and columns in a matrix as we know rows are the ones which run horizontally and

columns are the ones which run vertically. These are the examples of matrices. This matrix has 3 rows and 3 columns, and this matrix has 3 rows and 1 column, and this has 1 row and 3 columns.

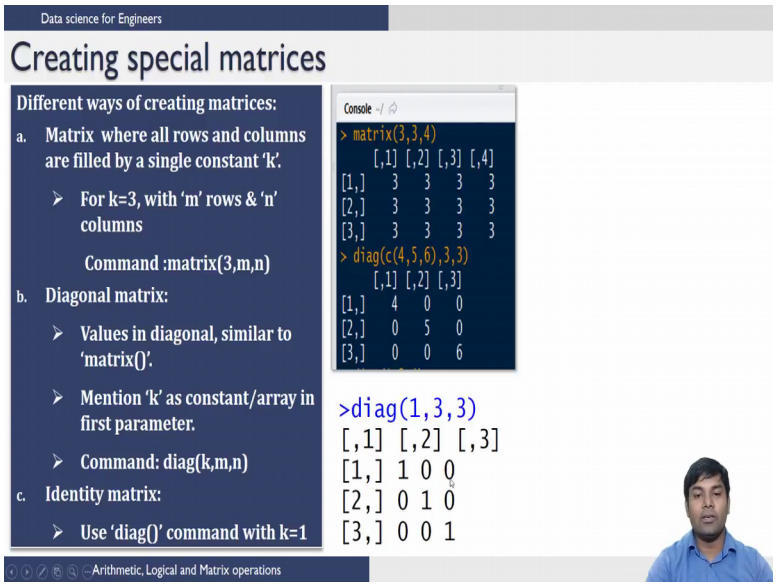
(Refer Slide Time: 04:06)



Now, let us see how to create matrices in R. To create a matrix in R you need to use the function called matrix. The arguments to this matrix are the set of elements that are needed to be the elements of the matrix. You have to pass how many number of rows, you want to have how many number of columns, you want to have in your matrix and this is the important one by row usually R arranges the elements you have entered in a column fashion, if you want the elements that are given to be entered in a row as fashion you have to say by row as true the default option for by row is false.

Now, we have seen; what are the things that are involved in creating a matrix. Let us create a matrix with the elements 1 to 9 which is containing 3 rows and 3 columns and you want to fill the elements in a row wise fashion this is the command which does this and if you see the output is 1 2 3 4 5 6 7 8 9 that are filled in a row wise fashion.

(Refer Slide Time: 05:10)

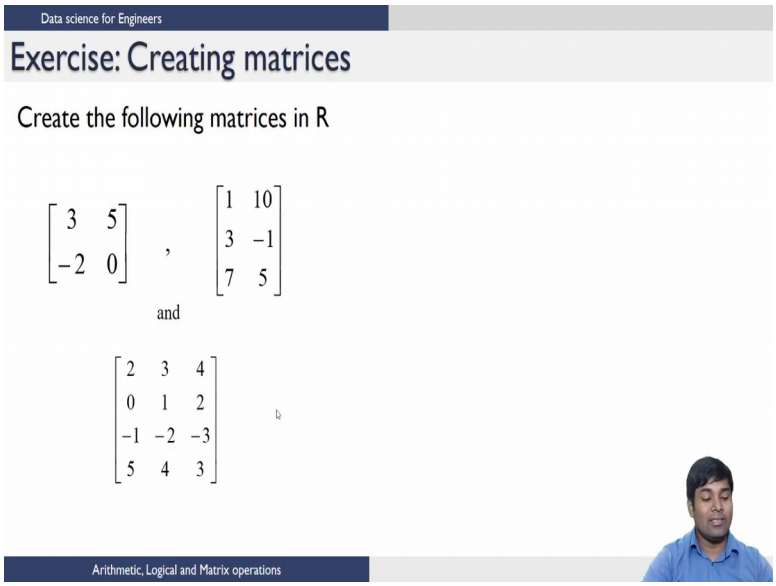


Now, let us see how to create some fashion matrices in R the first one is scalar matrix which contains all the rows and columns that are filled by single constant k. So, we need to specify the value to be 3 and you have to specify the number of rows you want and the number of columns you want. So, you want to fill all the rows and columns with the element 3 which is a matrix which contains 3 rows and 4 columns. So, you have specified 3, 3 and 4 when you do that you will get the matrix printed like this.

So, the command is matrix this is the element you want to print in all the rows and columns you have to specify how many rows and how many columns. Next we see how to create diagonal matrix the inputs you have to give for the diagonal matrix is the elements which you want to have in the diagonal and the dimension of the matrix. So, this is the command diag, the elements are vector of elements you want to have as diagonal elements and the rows and number of columns. So, see this example we want 4 5 6 ask the elements of our diagonals and you want to have a 3 by 3 matrix you can use this command and you can see that 4 5 and 6 are your elements in the diagonal and the rest of the elements are there.

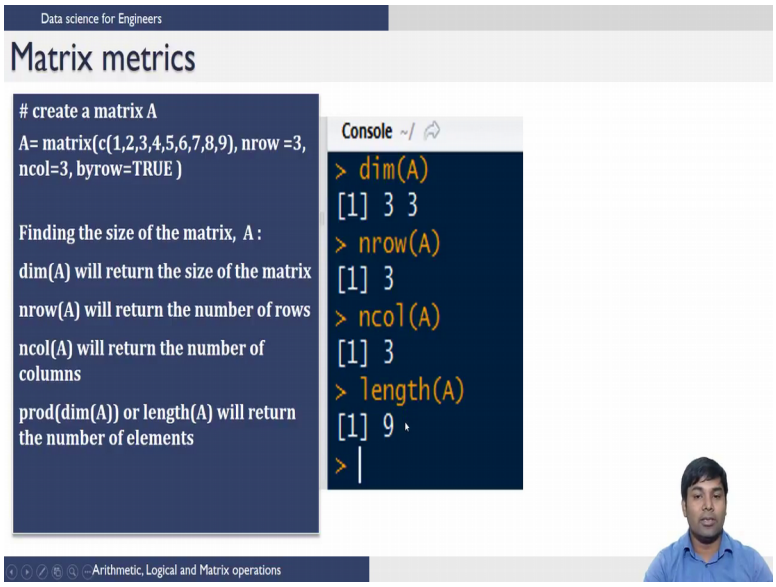
How do you create identity matrix? You can create an identity matrices in the diag command with the values in the diagonals has to be 1 and then let us say you want to create a 3 by 3 identity matrix you have to specify then rows as 3 and number of columns as 3 and it will put 1 in the diagonals with all other elements as 0.

(Refer Slide Time: 07:00)



Now, as an exercise you can try creating the following matrices in R.

(Refer Slide Time: 07:04)

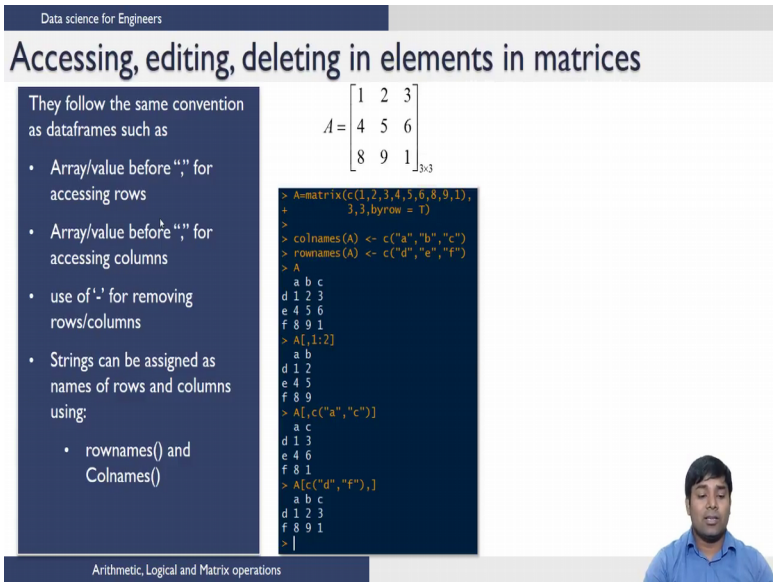


Next we move on to matrix metrics once a matrix is created how can you know the dimension of the matrix? How can you know how many rows are there in the matrix? How many columns are in the matrix? How many elements are there in the matrix is the questions we generally wanted to answer.

We can use the following comments to know all of this. Dimension of A will return the size of the matrix that will say what is the size of the matrix that is it is a 3 by 3 or 4 by 5

and so on, n row of a will return you number of rows and n column of you will return you number of columns. Either length of a or product of dimensions of A will return the number of elements that are existing in the matrix. For the matrix A which is created by using this command we can find that dimension of A will give you 3 by 3 because it contains 3 rows and 3 columns number of rows is 3 and number of columns is 3 and the number of elements that are present in the matrix is 9.

(Refer Slide Time: 08:09)



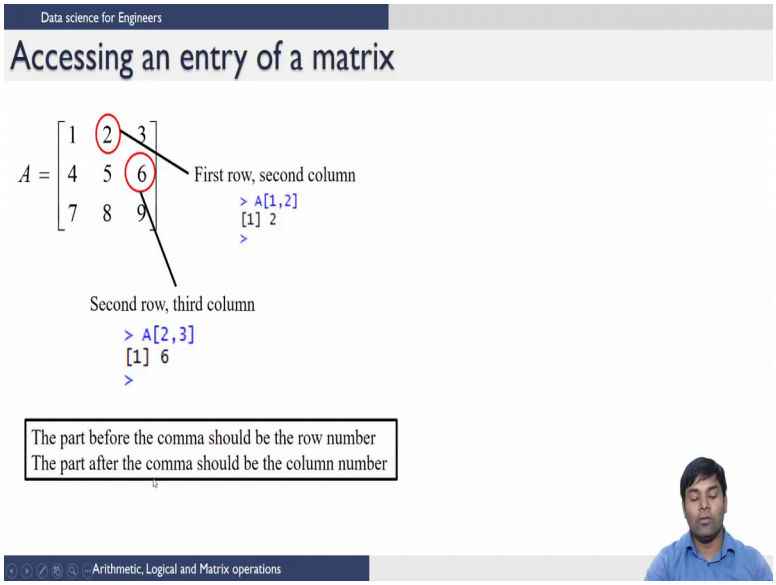
We can access, edit and delete elements in the matrices using the same convention that is followed in data frames. So, you will have a matrix and followed by a square bracket with a comma in between array and values before the comma is used to access rows and array or value that is after comma is used to access columns. If you want to remove some columns you need to add a negative symbol before the rows or columns, and you can also assign strings as names of rows and columns by using the commands row names and row columns.

Here we have created a matrix A which are having the elements 1 2 3 4 5 6 8 9 1 and it is a 3 by 3 matrix and we want to fill the elements row wise and we can now name the columns as a b c and name the rows as d e f. Once you do that and print a you can see that this column is named as a, and this column is named as b, and this column is named as c. Similarly we can see that row one is named as d, row 2 is named as e and row 3 is named as f.

Now, let us suppose you want to access the first two columns you can use the same convention as what we have used for data frames, A with the square bracket nothing before the comma and then you want access 1 to 2 that is first two columns of a you have to give that array here and then it will access the first two columns of A.

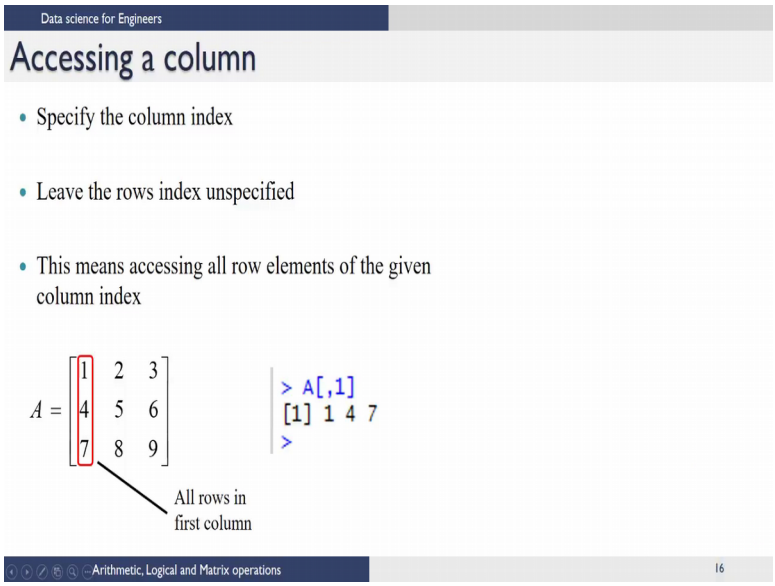
You can also access the columns using the names of the column as we have seen in the data frames. So, you want to access the columns a and c; that means, columns 1 and 3 you can do so, by specifying the names of the columns. Similarly you can also access the rows by using the names of the rows. You want to access first and third row which are having the names d and f, you can do so by using this command you want access row d and row f and all the columns. So, the output is shown here.

(Refer Slide Time: 10:30)



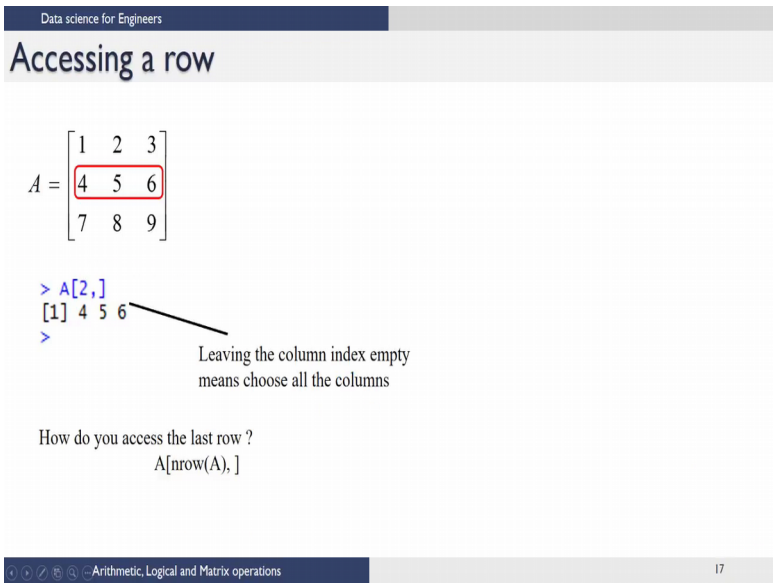
If you want to access an entry of a matrix you can use the similar convention. For example if you want to access this element it is in the first row and the second column the command you need to use is in the matrix A fetch the element which is in the first row and in the second column that will give you the output 2. And for example, if you want to access this element 6 you have to say it is in the second row and the third column you have to say A of 2 comma 3 it is give an output 6. As we have seen earlier the part before the comma should refer to the row number and the part after the comma should refer to the column number.

(Refer Slide Time: 11:12)



Now, let us see how to access a column of a matrix. So, specify the column index which you want access and leave the rows index unspecified. This means you are accessing all the row elements of a given column index. So, for example, if you want to access first column of the matrix A, what you need to do is A of all the rows and first column which will give you the output 1 4 7.

(Refer Slide Time: 11:41)

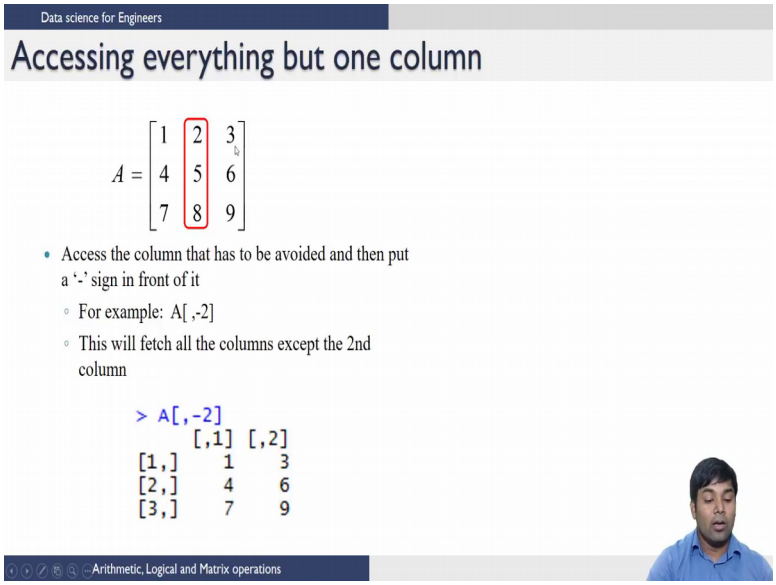


Similar to accessing a column we can access a row of a matrix. What you need to do is you need to specify the row which you want to access and specify nothing in the column

index which says access all the columns. If you want to access row 2 you have to specify in the row ID as 2 and leave empty space in the column ID and so that row two all the columns will print it and you will be able to access 4 5 6.

For you to think about how do you access the last row. Can you do something like this? You figure out by trying on your own.

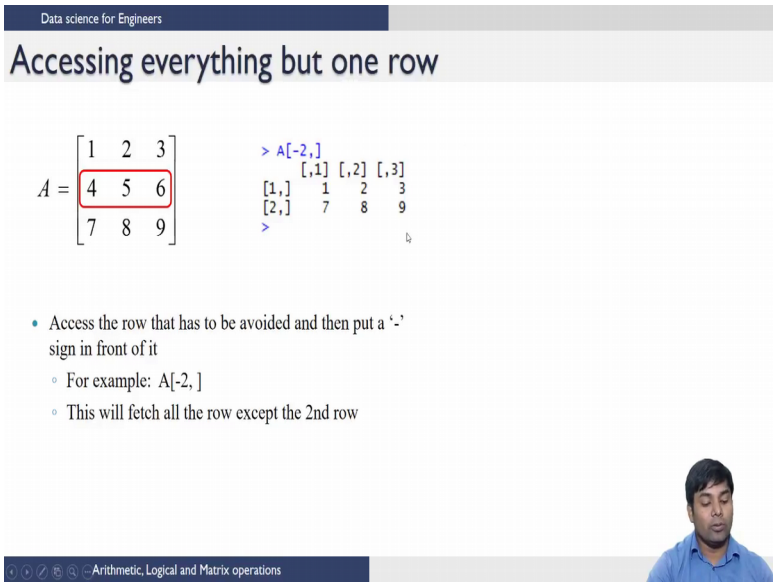
(Refer Slide Time: 12:20)



Next we will see how do access everything, but one column. I want to access in this matrix this part 1 4 7 and 3 6 9 I do not want this column to be in the matrix where I want to access.

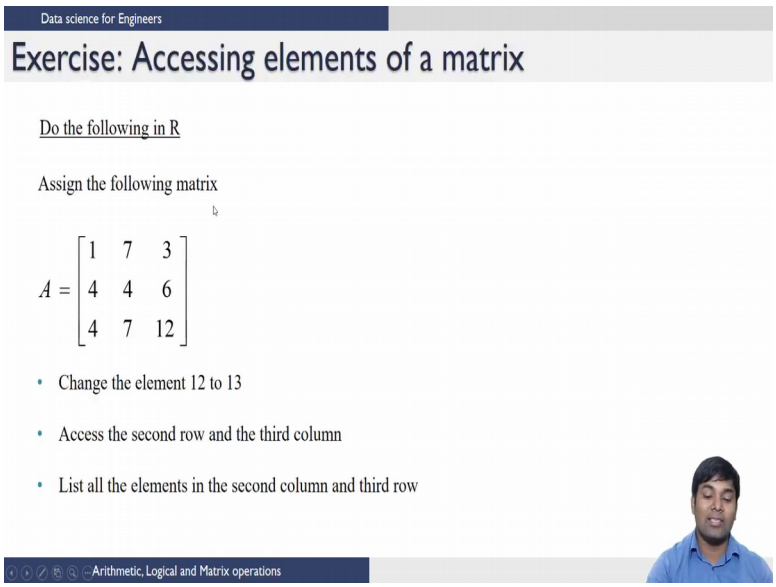
So, now what I have to do is it is like eliminating this column from the matrix you can do so by having a negative symbol before this is the second column you can say all the rows I want and I want to take this second column off and if I assign it back to A, I will get A as 1 4 7 and 3 6 9 or if you just print this a of all comma minus 2 it will give the desired result which is 1 4 7 and 3 6 9.

(Refer Slide Time: 13:09)



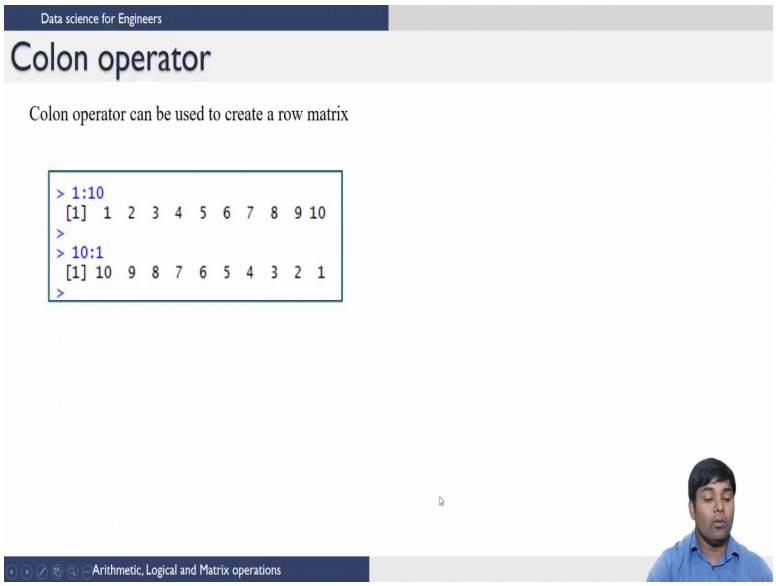
Similar to the one which you have seen in the earlier slide you can also access everything, but one row all you need to do is for example, if you want to access all the parts of a except this row you can do so by using this command I want to take the second row off and I want to have all the columns. Now, once when you do this command you will say 1 2 3 and 7 8 9 will be printed as your output.

(Refer Slide Time: 13:40)



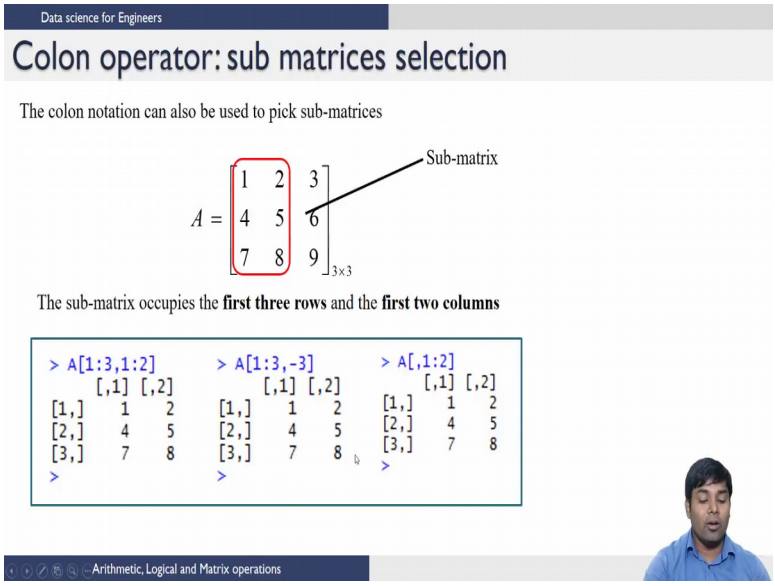
As an exercise to access elements of a matrix you can try solving this problems that are given.

(Refer Slide Time: 13:46)



Now, we will introduce what is called as a colon operator. Colon operator is used to create an array of elements with equal width for example, if I type in 1 to 10 it will create numbers from 1 to 10 with gap of 1. I can also reverse the order it will print from 10 to 1 with a gap of 1. Why is this colon important? If you would have realized I would have used something similar while accessing the number of rows or columns in the previous slides. Let us look how to do this.

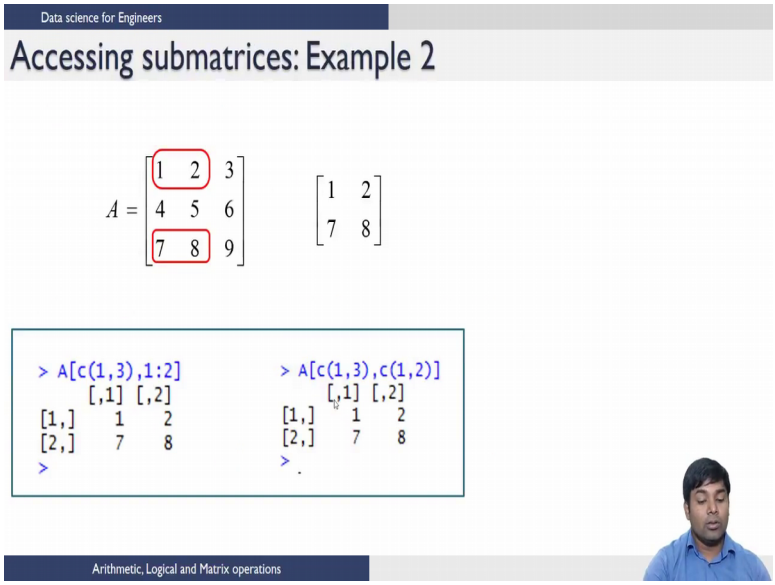
(Refer Slide Time: 14:22)



For example if you want to select a part of matrix which has sub matrix you can use this colon operator ok. So, let us now see if I want to access the first 3 rows and the first 2 columns of this matrix, how do I do this? I want to access rows 1 to 3 and also access columns 1 to 2 do. So, you can see this colon operator is helping us in accessing the sub matrices from the matrix.

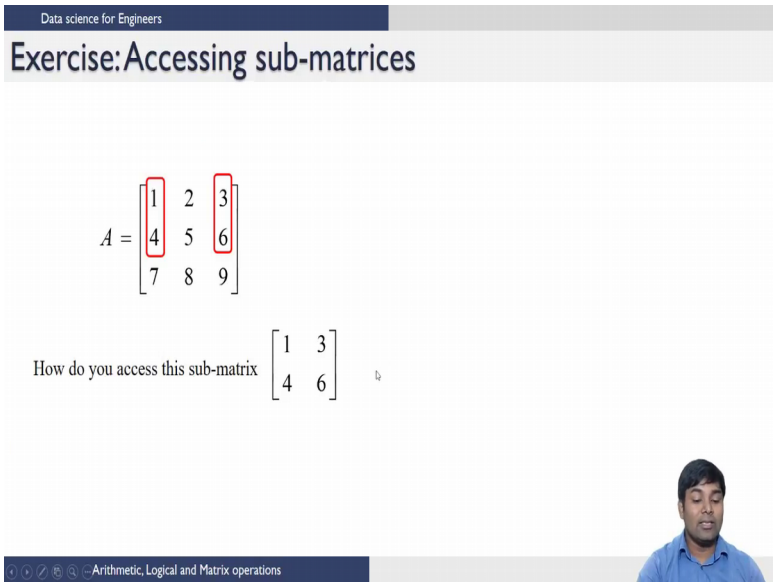
In this example what does it says is I want to access all the 3 rows and I do not want the third column. This is same operation, but done in a different fashion. You can also do the same I want to access all the rows, but it has to be coming from first two columns only. So, you can see that you can access sub matrix in different fashions depending upon the way you are comfortable with.

(Refer Slide Time: 15:20)



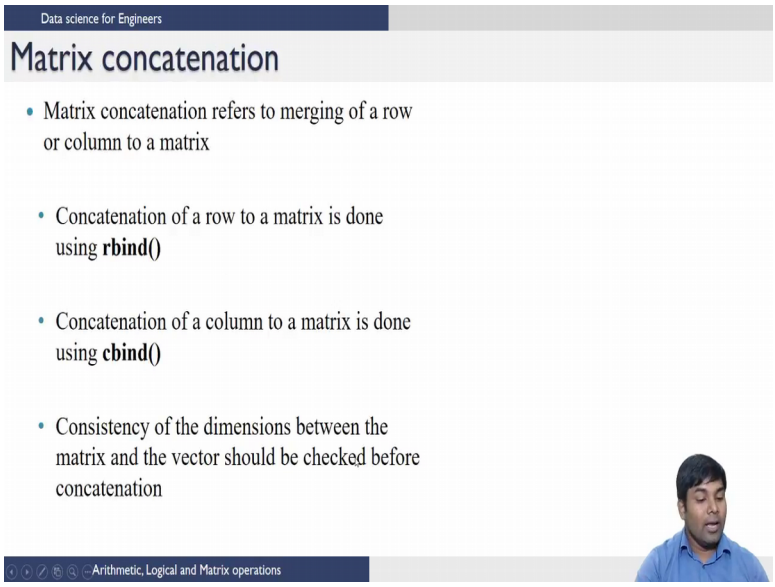
So, this is another example of accessing sub matrices if I want to access this 1 comma 2 and 7 comma 8 and have it as a sub matrix separately how do I do this. I want to access rows 1 and 3 and what are the columns I need to access in the columns 1 and 2. So, I have to say in the columns 1 and 2 access the elements which are in the row 1 and row 3, that brings me the matrix. You can use the concatenation operator also for both the arguments like shown here you can use c of 1 comma 3 and c of 1 comma 2 which gives you the desired result.

(Refer Slide Time: 16:00)



You can try this as an exercise for accessing sub matrices.

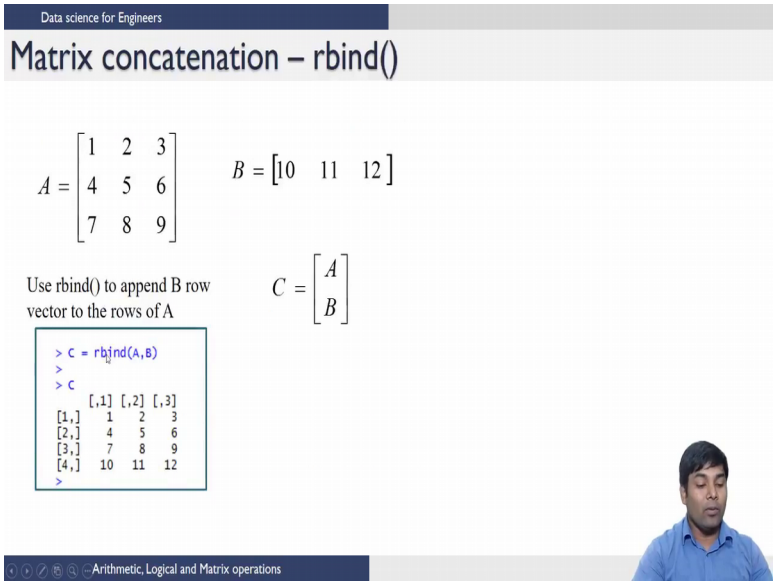
(Refer Slide Time: 16:06)



Next we move on to another important operation on matrices which is matrix concatenation. Matrix concatenation refers to merging of rows or columns to an existing matrix. If you want to add a row to the existing matrix you can do so by using R bind command. If you want to add a column to a matrix you can do so by using c bind command. So, one thing you have to keep in mind is you have to make sure the

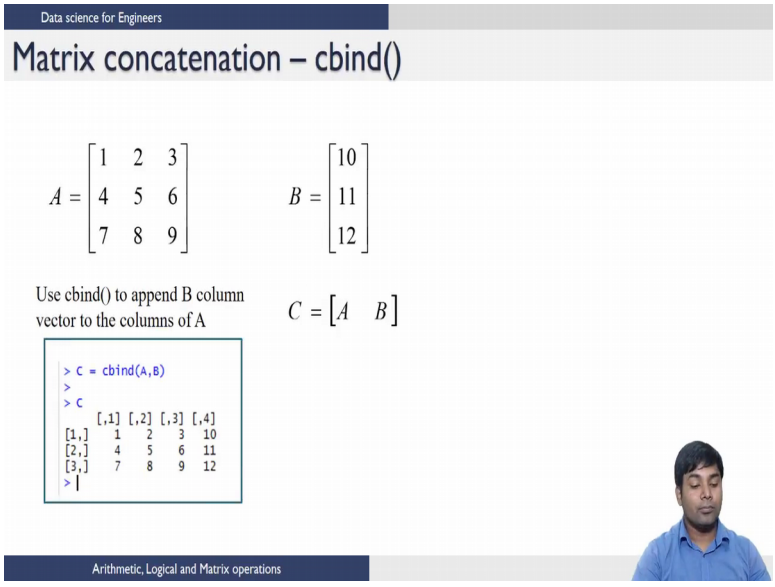
consistency of dimensions before you do this matrix concatenation. Let us illustrate how an R bind works.

(Refer Slide Time: 16:38)



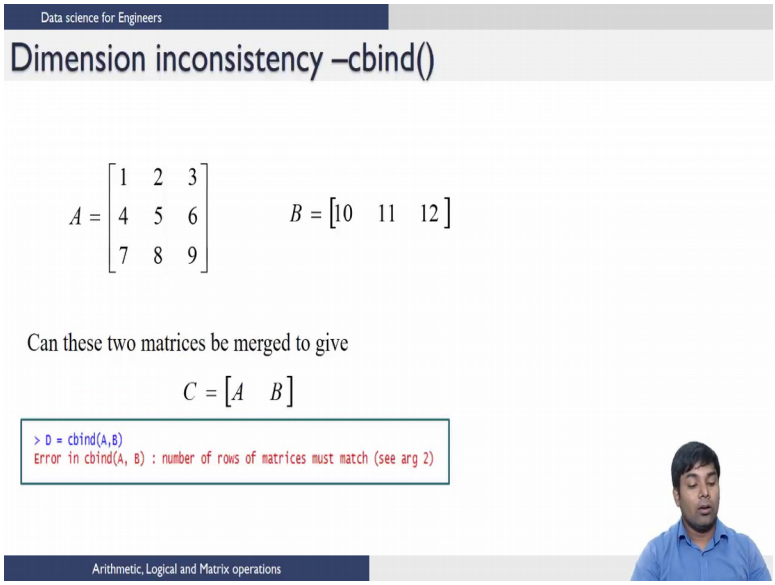
Let us suppose we have a matrix A and matrix B and you want to concatenate this matrix B as a row in matrix A that can be done using the R bind command which is shown here. I am concatenated matrix B to the matrix A and I am assigning it to the variable C. So, when you do this command you can see that the matrix C is having the row 10 11 12 which is the matrix B and is concatenated to the matrix A.

(Refer Slide Time: 17:13)



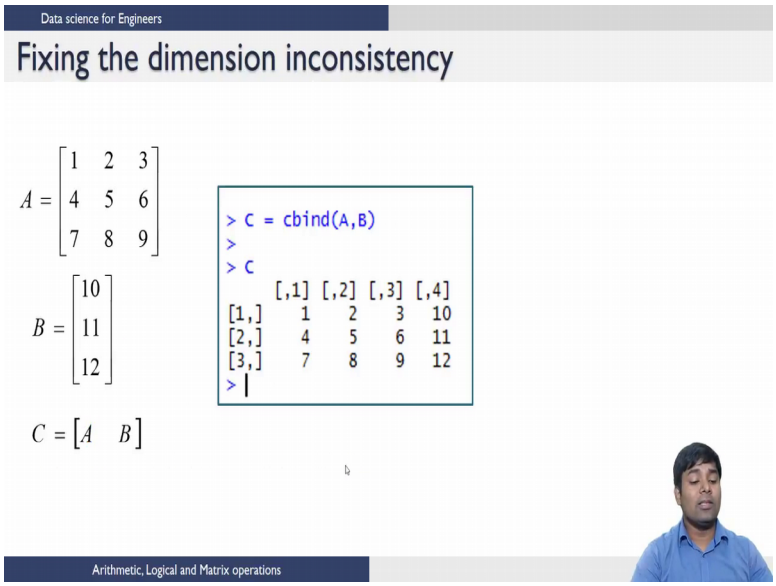
Now, let us see the C bind. Let us say you have this matrix A and we have matrix B which is shown in the screen you want to concatenate this B matrix with the columns of A. You can do so by using the C bind command which is shown here C by pass the first matrix A and second matrix B and assign it to the variable C. When you print the C you can see that the matrix B has been concatenated as a column to the matrix A.

(Refer Slide Time: 17:50)



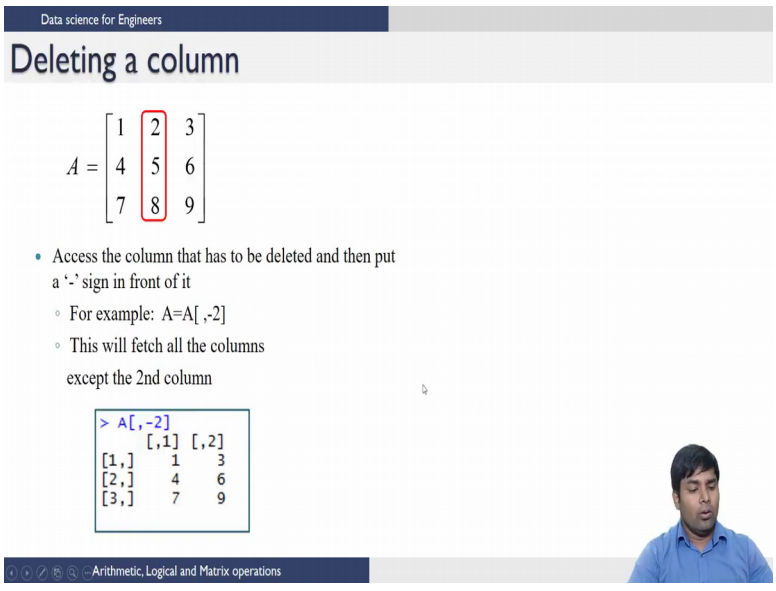
Now, let us try to concatenate this B to this matrix A using C bind. What would do you expect? We expect an error because A is having the dimension 3 by 3, but B is having 1 by 3. If I want to do a column bind the dimension of matrix B would have been 3 by 1, but it is 1 by 3 which is inconsistent that is why you will get an error, error in C bind of A number of matrices must match.

(Refer Slide Time: 18:26)



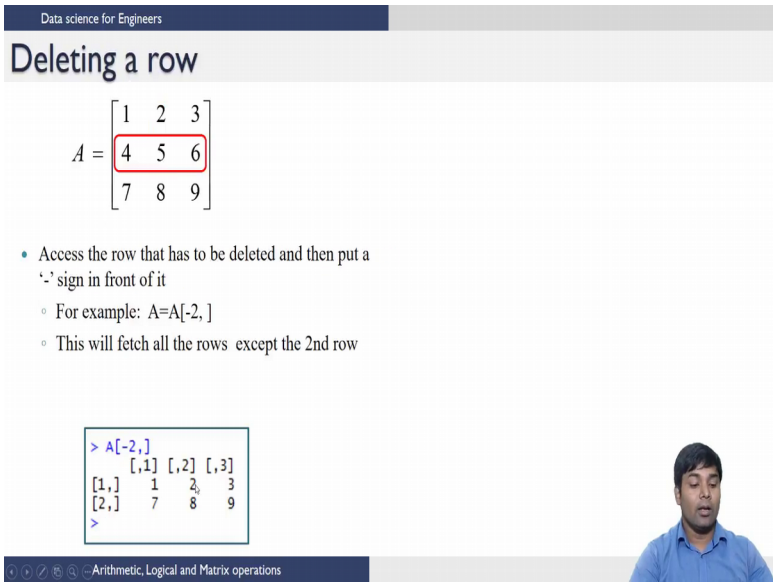
Now, if you want to resolve this dimension inconsistency you have to transpose this B and then have this as 3 by 1 and now A is 3 by 1 now you can easily do the C bind operation by using C bind command C bind of A comma B and assign it to C. Now, you can see that this C bind it happened and the B is concatenated to the matrix A.

(Refer Slide Time: 18:52)



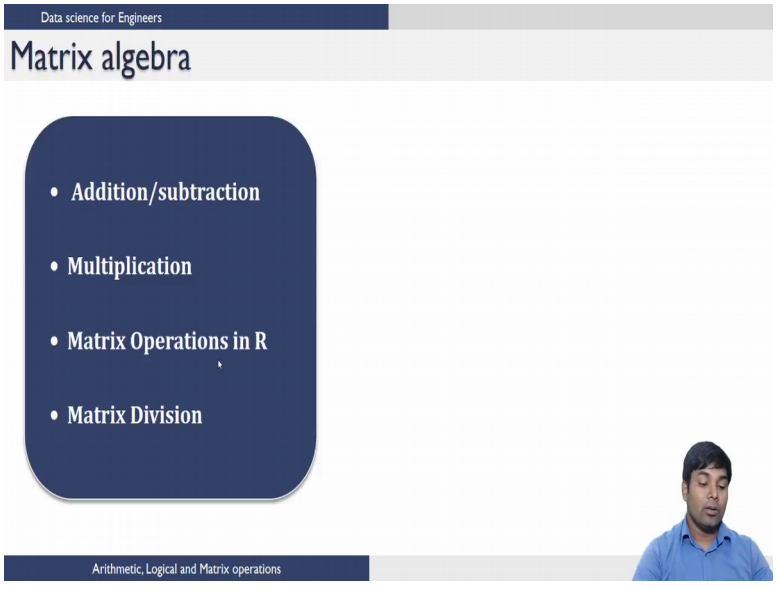
You have seen how to delete a column, you can use negative symbol before the columns which you want to delete and then assign it to A you will see that the required output is printed.

(Refer Slide Time: 19:05)



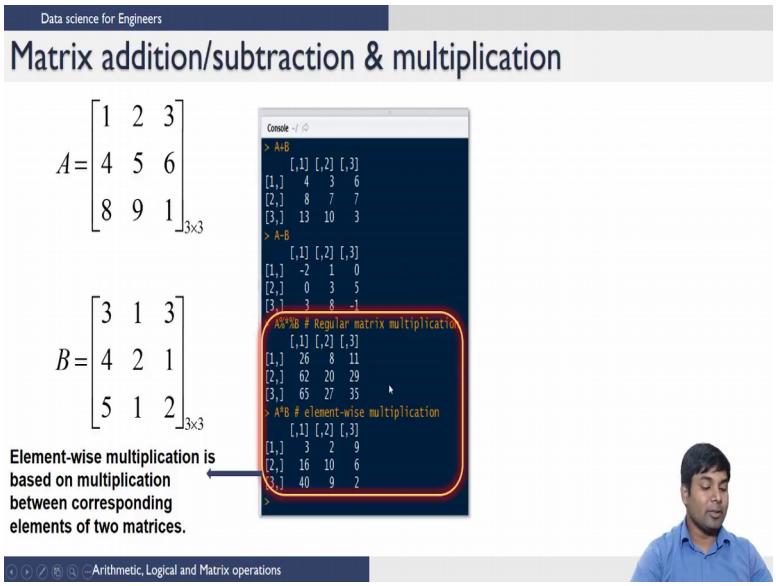
Similar to what we have seen in the earlier slide we can also delete a row from the matrix which is, let us suppose we want to delete this row 2 you have to say minus 2 and then all columns and then assign it back to A. You can see that in the output the row 2 is deleted.

(Refer Slide Time: 19:23)



Now, let us see how to do algebraic operations on matrices such as addition, subtraction, multiplication and matrix division in R.

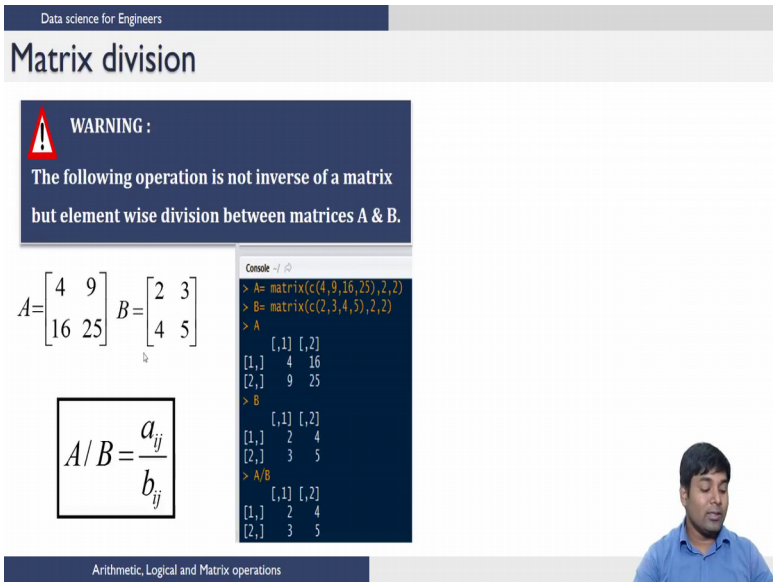
(Refer Slide Time: 19:35)



Let us suppose we have two matrices A and B which are shown here. Matrix addition is straight forward you can say A plus B you will get the output. So, 1 plus 3 is 4, 2 plus 1 is 3, and 3 plus 3 is 6 you will see the element wise operation happens that is what normal matrix operation is also about.

So, you can also do the subtraction, multiplication is little bit trickier when you say A has trick B it will perform element wise multiplication such as 1 into 3 is 3, 2 into 1 is 2 and 3 into 3 is 9. But if you want to have a regular matrix multiplication you have to use percentage symbol before and after this hash trick that will perform the regular matrix operation.

(Refer Slide Time: 20:26)



Now, let us look at matrix division. Let us say you have two matrices A and B which are 4 9 16 25, and 2 3 4 5 respectively. Now, if I do A by B what it does is element wise division, but not the inverse of a matrix. So, you have created matrix A matrix B and then if you do A by B you will see that 4 by 2 is 2 9 by 3 is 3, 16 by 4 is 4. So, let us suppose you have two matrices A and B as shown in the figure when you do A by B it will perform an element wise division, but not the inverse of a matrix.

In this video we have seen how to do arithmetic logical and matrix operations in R. In the next lecture we are going to discuss about how to write functions in R, and how to invoke them, how to use them to perform the task we wanted.

Thank you.