Multivariate Procedures with R

Prof. Shalabh

Department of Mathematics and Statistics

IIT Kanpur

Week – 02

Lecture – 07

Matrix Operations

Hello friends, welcome to the course Multivariate Procedures with R. You can recall that in the last couple of lectures we were talking about different types of calculations in the R software and you have understood that the way R does different type of mathematical procedures like as addition, subtraction, division, multiplication that is possible with the scalar as well as data vector. And these operations are little bit different than other softwares. But I am sure that now you have understood those operations. So now in this lecture and in the next lecture we are going to talk about the matrix theory. The first question comes here why should we learn this matrix theory? You see what is the basic objective of this course? This is to have the multivariate procedures.

Multivariate means there are more than one variable. So, whenever we are trying to deal with the multivariate procedures be it a multivariate normal distribution, be it a multiple linear regression analysis, be it a discriminant analysis, principal component analysis or anything. The data is available on more than one variable. What do you mean by variables? So, there are two variables age and weight.

So, there will be one person we try to record her or his age as well as weight. Now, the second person comes and we try to record the weight at an age etcetera and so on. And similarly, you can think that there is more than one variable or more than two variables which are going to affect the process. So, every set of observation will have not only one value, but more than one values. Suppose if you want to record the data on the health of say 20 persons and now you define that health is going to be measured by say age, weight, height, blood pressure, blood sugar level etcetera.

Then each of the 20 persons will have observations on all these variables. So, the question comes here, how are you going to handle such datasets? The first question comes we have to give the input to a software and in most of the cases you will see that such observations are arranged in the form of vectors and matrices. Remember I am not

talking of the data vector, I am talking of the vectors in the matrix theory like as row vector or column vectors. So, it is important for us to understand that how R is handling the matrix and its operations. Now, before going forward I assume that you all are aware about the matrix theory and its various operations and you will agree with me that many of the operations in the matrix theory they are very much different than the usual scalar operations.

And when we try to apply those procedures in the R software one aspect which you always have to keep in mind is that R is going to follow the same mathematical rules for the matrix theory. In case if you make a mistake in giving the input to the data or somewhere else R may not be able to exactly diagnose what is the mistake, what is the problem. But this is only you have to see and find out what is the problem. For example, in case if you are trying to multiply two matrices. So, wherever you are trying to multiply two matrices the rule is that if there are two matrices say A and B and suppose one's order is m by n then the order of the second matrix has to be n by p or something like this.

So, such type of operations I am assuming that you know. So, in this lecture I am going to introduce this matrix theory to brush up your knowledge. So that you are aware and then you have a quick review of the basic concepts in the matrix theory. Although there are many many operations in the matrix theory and it is practically impossible for me to cover all those operations. But I will try to take up some standard operations in this lecture and in the next lecture and hopefully after that you will be confident that if you want to use any other matrix procedure you can use it and you can find out the way or the syntax or the command in the R software.

So, with this objective let us begin our lecture and try to have some glimpse of the matrix elements in R software. Right. So, in this lecture first we try to define that what are matrices and as I said that matrices are important objects in any calculation and particularly you are trying to do the multivariate analysis then this matrix theory becomes an inseparable part of the whole analysis. What is a matrix? Matrix is the arrangement of elements in rows and columns. For example, if I say in mathematics, we try to write down a matrix like this and then I will say here this is the first element a_{11} this is a second element a_{12} and similarly in the next element a_{21}, a_{22} and so on. So, these are called rows and these are called columns, right. And this subscript if you try to see what is this a11 this is indicating the number row number and column number.

Similarly, if I say here a_{12} so a_{12} is indicating that is the element in the first row and second column, right. And if there are such here 1, 2 up to here p rows and suppose here 1, 2 up to here q columns or say n columns then this is called as a matrix of order p by n, right. So this matrix is a rectangular array with p rows and n columns and as we write in the book as I explained you suppose if I say, if I try to denote the matrix by here x then

usually by x_{ij} we write down its elements, but this x_{ij} is a written in the book that is how we try to understand the matrix theory from the book, but in the software and in the programming language, usually we try to write this thing as x[i,j]. So, here i goes from 1 to p and j goes from 1 to n. Now, when we are trying to deal with the matrix operation in any software or in any programming language like as R then the element of this matrix can be an object for example, that can be a number that can be a string and, but in mathematics when we are trying to do the mathematical operation then usually, we are interested in the numerical matrix. Numerical matrix means all the elements in the matrix should be some numbers, they should not be character otherwise we cannot do the mathematical operations although in order to do the data analysis many times we try to arrange the characters also in the format of a matrix, but anyway at the moment we are not going to consider those things and we are going to assume that the elements in my matrix they are the real numbers, right.

So, now I have two options here that first I try to give you the basic command of the matrix theory and then I try to explain you how you can do it in the R software or let me take here some examples. And through those examples I will try to explain you that how you can make different operations in the matrix theory in this R software. So, let me try to take here first example suppose I want to create a matrix. Creating a matrix means that data has to be arranged in the matrix. Now when we are trying to create a matrix what do we need? The first ingredient is that we want to know what is the number of rows then we want to know what is the number of columns and then we want to know what is the data that means what are the entries in the matrix.

So, suppose if I want to create a 4 by 2 matrix that means 4 rows and 2 columns. So, if you try to see if there are 4 rows and 2 columns then how many data entries you require this will be 4 into 2 that is 8. So, now I have to supply this information to the R software. So, in order to create a matrix or to inform the R software or to request the R software to create a matrix the command is matrix m-a-t-r-i-x and then inside the parenthesis you have to give different type of options or information. So, I want to inform the R software that there are going to be 4 rows.

So, this is obtained by writing n row which is indicating the number of rows and I write here number of rows that is n row equal to 4 and similarly here n col n col means here number of columns. So, ncol is equal to 2. So, that is indicating that there are 4 rows and 2 columns. Now I want to inform R that what is the numbers or what are the real numbers which are going to take place inside the matrix. So, I try to give here 8 values which I have combined here in a data vector.

So, and I give it and I use here this option data. So, data will communicate to the R software that these are the values which are going to be inside the matrix. So, once you execute this thing you will get here this type of outcome x will look like this. See here the

if you try to see the data, I have given here for the sake of understanding in an increasing value 1, 2, 3, 4, 5, 6, 7, 8. Now if you try to see that how this data is going to be entered in the matrix, x.

```
> x = matrix( nrow=4, ncol=2,
                      data=c(1,2,3,4,5,6,7,8) )
> x
      [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```

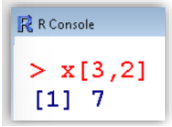
Here 1 comes first, then comes 2, then 3, then 4 and from here it goes to the next column 5, then 6, then 7, then 8. So, if you try to see the data here is arranged column wise. And the same operation when I try to do in the R software comes out to be here like this that you can verify this is the same output which is here. And what you can observe here I will explain you later that what are these indicating for example, this here 1, this here 1. And if you try to observe how it is written.

This is written like first you observe these brackets, then it is here the number 1 and then here comma. Whereas in the column case please try to concentrate where I am writing it is here bracket, then there is blank space comma and then it is 1, right. So, if you try to see this is indicating first column. And similarly, this is indicating second column. Similarly, this is indicating first row and this is indicating second row and so on, right.

And in case if you want to write what is the address of this number here 2. So, this you can see here this is on the second row and first column. So, this is how we try to give or the R is trying to give the addresses to different locations to rows, columns as well as individual number. So, this is I just want to had your attention so that later on when I am trying to explain you that how to access any particular row, column or element this is going to be helpful, right. So, now you can see here that this here this parameter nrow that is defining the number of rows in a matrix ncol is trying to define the column numbers of the matrix and data is assigning the specified value to the matrix, right.

And when you are not giving any option then this matrix is trying to assign the observation column wise you can see here. And if there are more columns it will go like this, right. So, this is what you have to keep in mind, right. So, now in case if you want to access any particular element of this matrix because it may be required in various multivariate procedures that you want to do something on a particular element and then you need to write a command so that a particular element can be obtained. So, as I explained you just now, suppose you want to extract this number 7.

```
> x[3,2]
[1] 7
```



So, now if you try to see here this number 7 is lying on the third row and in the second column. So, what you have to write here the name of the matrix x and using the square brackets you have to write 3, 2. And in case if you try to write down here x inside the square bracket 3, 2 you will get here the value here 7, right. One thing I can explain you that as in mathematics we have different types of brackets like as parenthesis, curly brackets and square brackets. So, in R when we are trying to do the mathematical operation then we use only the parenthesis like this.

The curly brackets and square brackets they are used for some other assignments. For example, they can have different role for example, here you can see when you are trying to find out the particular value in the matrix then we are trying to use this square bracket, right. So, now in case if my objective is that I want to arrange the same data, row wise. Why row-wise? As I explained you here that if you do not give here any option then this data is going to be arranged column-wise. Now I want to arrange it row-wise.

So, how to do it? So, I will use the same command here same data set, but I want to arrange in the row-wise. So, I am using the command here matrix, nrow, ncol and data, but now I add here one more parameter byrow, right. And I write here byrow is equal to TRUE, TRUE in all in uppercase alphabets. Actually, here this TRUE and FALSE they are the logical operators. And instead of TRUE and FALSE you can also use here say a T or F also, the first alphabet.

So, if you try to understand what I am trying to inform to the R software. I am saying byrow is equal to true that means, I am asking R that when you try to arrange the data then you have to arrange the data by rows or and this statement is true that means, you please arrange the data row-wise. In case if I write down here FALSE byrow is equal to FALSE that means, I am trying to inform the R software that you need not to arrange the data row-wise. That means obviously, the opposite will be correct that the R has to arrange the data column-wise. So, if you try to see here by using this new parameter byrow equal to TRUE how this data is going to be obtained.

```
> y = matrix( nrow=4, ncol=2,
          data=c(1,2,3,4,5,6,7,8), byrow = TRUE)
> y
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
```


So, you can see here this matrix I am calling here as a y and the y is coming out to be like this. Now, the same data is here 1 to 2 it is going first in the row first row then it come to the second row 3 to 4 then 5 to 6 then 7 to 8. So, this is how we try to control this data arrangement and you can see here that in this screenshot I can show you that here I am not using here byrow option. Whereas in the second case I am trying to use here the byrow equal to TRUE and the data is here the same 1 to 8 numbers. Number of rows and number of columns are also the same, but here the data is arranged column-wise, but in the second case the data is arranged row-wise, right.

So, let me try to show you first these examples on the R software so that you can be confident about them. So, let me try to just copy this thing. So, then I try to copy this thing, right. So, even if you want to write it in the same line that also can be done that is the same thing, right. Now, if you try to see what is here x, this is the same operation which I just explained you, right.

Now, in case if I try to change this command and I try to write down here byrow is equal to TRUE. So, okay I will just write down here as a T only so that you can be confident that T and capital T-R-U-E both are working and let me give this matrix a name y. So, you can see here y is now here like this. So, that is the same operation which I just explained you, right. And suppose if I want to extract any particular element of here x, suppose I want to extract here x[3,2].

So, this is going to be here 7. Whereas if I try to arrange or retrieve the same element in the y matrix, then it is going to be here 6, right. And that you can see here this is the element in the x matrix and this is the element in the y matrix, right. So, now you can be confident that yeah this is working without any problem. And if you try to understand here one concept that try to look at this screenshot and, in this screenshot, I am not writing anything about byrow in the first here say byrow. So, what is the value of here byrow? So, obviously byrow here is actually FALSE, but this is the default option that even if you do not write anything about byrow, then R is going to assume it to be FALSE that is the default option.

So, you have to be careful according to your requirement and need, you need to use this operation, right. Okay. So, now we move forward. Now I would like to introduce you that

sometimes data is coming from external sources and it is arranged in some matrix. So, in this example what you have done here, here you have tried to create a matrix of order 4 by 2. But suppose some unknown data set is coming from outside and you do not know what is the number of rows, what is the number of columns, etcetera and in order to move further you need to have such type of information.

So, the question is that how would you know what are the dimension of the matrix, what are the number of rows and what are the number of columns. So, if you want to find out the dimension of the matrix, then the command here is d-i-m, dim which is the short form of dimension. If you simply write dim and inside the parentheses x the name of the matrix, you will get here 4 by 2 which is 4, 2. So, you have to observe how it is written, right. And if you want to know the number of rows in the matrix x, then you have to use the command nrow inside the parentheses x, right.

And it will give you the option here, the answer here as say 4. And similarly, if you want to know the number of columns, then you have to use the command ncol x and it will give you the number here 2, right. So, this is how you can have different type of information. And I would like to inform you that you see I am trying to consider here only very limited number of options which are more popular or they are going to be used in our course particularly, but there are many more options. So, I would recommend you that you please try to look into the help and you simply say help inside the parentheses within double quotes matrix and you will get here all the details about the matrix.

And there are many other options which are in several like as matrix, is matrix. Suppose that means if there is some unknown data set and you want to know whether it can be treated as a matrix or not, then this is.matrix() is going to help you. But definitely I am not going into these details and but I would certainly like to show you the use of this dim command and nrow in x, right. So, let me try to clear the screen and let me try to have this matrix here x.

```
> dim(x)  # tells the
[1] 4 2     dimension of matrix
```

So, you can see here this is here like this. And now if I try to write down here dim(x) which is here like this 4 2 that means there are 4 number of rows and 2 columns and if I see here nrow x, so this is here 4 and if I see here ncol(x) is here 2. And similarly, if you try to take any other matrix, then the similar observations can be obtained. For example, if I try to show you that if you try to make any mistake, then what will happen? Suppose if I am in the y matrix, suppose if I try to add here say 2 more numbers 9 and 10, so there is total 10 values but you want to arrange them in the 4 rows and 2 columns which is wrong. But it will try to say that it is y is giving here like this but it is giving you a warning message and if you try to see what it is trying to do, it is trying to take only these

8 values. And now similarly if you try to make another mistake that if you try to make here the number of rows to be here 6 that means you need here 12 observations but now you have given only here 10 observations, then what happens? It is again trying to give you the warning that there is something is wrong, you try to find out and if you try to see here how it is doing.

So, it is here 1 to 8, it is arranged say here 10, it is arranging in the 5 rows and 2 columns but after that it is repeating. So, this is you have to keep in mind that this is how the R is working. So, if you do not give the proper input and if you ignore the warning messages in your simulation or in your computation, then the R is going to use this value but you wanted to use only up to here 10. So, this is what you have to be careful. So, let me try to clear the screen and come back to our slides.

So now I try to give you here different types of examples in which we are going to give you the idea that how different types of matrices can be defined. Well, these examples we will help you when you are working in the multivariate procedure and if you want to input a particular type of data or if you want to do a particular type of operation. Suppose you want to have a matrix in which all the elements should be the same, right. Suppose I want to create a 4 by 2 matrix in which all the element should only be 5. So, I can use here the same command matrix, then nrow equal to 4, ncol equal to 2 but I will say only data equal to here 5 and as soon as you try to do it here, you will get here this type of observation, this type of matrix where you can see there all the 8 elements in the matrix x, they are the same 5.

```
> x = matrix( nrow=4, ncol=2, data=5 )
> x
     [,1] [,2]
[1,]    5    5
[2,]    5    5
[3,]    5    5
[4,]    5    5
```

So, this type of operations are many times needed, right. And suppose if you want to create a diagonal matrix, for example identity matrix is a diagonal matrix in which all the elements on the diagonal elements are 1 and off diagonal elements are 0. What are the diagonal and off diagonal elements? If you try to take a square matrix, then if you try to write down here elements like a_{12} and so on a_{21}, a_{22} etcetera like this, then this is actually here the diagonal. So, these are called the diagonal elements and all these elements which are not on the diagonal, they are called as off diagonal elements and this part is called as upper diagonal and the elements in the lower part, they are called as lower diagonal, right. So, if you want to create a diagonal matrix, suppose you want to create a identity matrix of order 2 by 2, then the command here is diag() and you see here 1 which is the value of data and you define the number of rows and number of columns

as nrow = 2 and ncol = 2 and if you try to see here this outcome will come here like this, right. And if you want to have any other value on the diagonal, then also you can play with this value 1 and I will try to show you it on the R console, right.

```
> d = diag(1, nrow=2, ncol=2)
> d
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

And there is another concept in the matrix theory which is transpose of a matrix. Transpose of a matrix is defined as when you try to interchange the rows and columns. Suppose if I see here my matrix here is 1, 2, 3, 4 and now I try to define here the transpose which is denoted by x prime which is here x transpose or then the first row will become here first column and the second row will become here second column. So, this is a very popular operation and if you want to do it, then the command here is t(x), but let me try to show you with this an example. Suppose I try to define here the same matrix x which is 4 by 2 matrix and the data is here 1 to 8, right.

```
> x = matrix(nrow=4, ncol=2, data=1:8,byrow=TRUE)
> x
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
```

This is another way of writing the elements 1, 2, 3, 4 up to 8 like this, right and byrow equal to TRUE. So, you get here this matrix, right, 1, 2, 3, 4, 5, 6, 7, 8 like this in which the data is arranged forwards. Now we want to find out the transpose of this matrix. The R command to find out the transpose of a matrix is t().

```
> xt = t(x)
> xt
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8
```

Inside the parentheses you have to write down the matrix. So, I write down here t(x) and then you can see here the outcome and suppose I store this transpose in another matrix xt. So, this is here the outcome that now this 1 and 2 they are coming on the first column 3 and 4 they are coming on the second column 5 and 6 they are coming on the third column and 7 and 8 they are coming on the fourth column and if you try to compare them in this

screenshot you can see here your x matrix was like this 1, 2 were the first row, but now 1, 2 are becoming the first column. The second row is becoming here the second column, third row is becoming the third column, fourth row is becoming the fourth column, right. So, this is how we try to do, but before going forward let me try to show you these operations on the R console so that you feel more comfortable. Suppose I want to create here a matrix 4 by 2 in which all the elements are going to be here 5.

So, you can see here 5 and similarly if you want to change this number suppose I want to give this number to 15. So, you can see here x is coming out to be here 15 right and similarly if you want to create here a diagonal matrix. So, you can see here this is here your diagonal matrix of order 2 by 2 which is an identity matrix and if you want to change the order of this matrix that means if you want to take a 3 by 3 identity matrix. Now you can see here there are 3 elements and if you try to change the value on the diagonal suppose if I want to make it here 2. So, you can see here now this is a diagonal matrix in which all the diagonal elements are 2.

And on the other hand I can also show you that if you want to make a diagonal matrix of different numbers. So, I can see you here see 1, 2, 3 and you will see here this will come out to be here like this that the first diagonal has value 1, second diagonal 2 and third diagonal here 3 right. And before I try to give you other thing let me first try to show you the transpose concept and then I will try to show you something where you have to be careful. So, this is your here x matrix and if you want to find out here transpose of x this is coming here like this, right. So, now if you try to see suppose if I try to write down this command here which I use earlier, right.

If you try to recall this thing x this is your here x. Now suppose if I do not write this parameter here and suppose if I write down here 4, I delete nrow, I delete the word ncol and I delete the word data and if I try to execute it, it will give me the same value. No, it is not happening because the first value it has taken which is here the data right. So, you have to be very careful that you should not go for a shortcut, but I always recommend that you try to write down this nrow, ncol all these values very carefully. And if you for example, if you try to see here if I try to write down here data equal to this then what happens you can see here it is giving you the same matrix, but in this case what is happening it is trying to take here the first value 4 as the number of rows and second value as a number of col, number of columns and if I try to suppose change it you can see here this becomes here like this. But as a layman it is very difficult for me to understand by looking as 4, 2 and 2, 4 that whether it is the number of rows first then the number of column or vice versa.

So, my simplest position to you all is that always write the parameter name such that it is 100 percent clear to everyone what we are trying to do and there is absolutely no confusion. So, now we come to an end to this lecture and you can see here I have taken

very simple topics and my objective was to introduce you and make you familiar with the matrix in R software that how are you going to handle it and you can see here it is not difficult, but as I shown you in the last example that you have to be watchful, you have to be careful when you are trying to define the values in the matrix options, for example, the values of different parameters you have to define it carefully and please write all the commands always in detail. So that after sometime if you are looking at the same command at least you can recall them. Besides, this thing it is very important R will do all matrix operation following the rules of matrix and this is only you who has to give the input in the right format.

So, this is important for you to understand the basic functionalities of the matrix theory basic mathematical rules in the in the matrix theory and then try to write down the syntax and commands accordingly. So, I would request you all that you try to take some simple examples and try to practice that how you can define different types of matrices try to play with the number of rows, number of columns, diagonal matrix etcetera and try to see that how you can do it and make yourself familiar with the matrix theory. And in the next lecture I will try to take some more basic operations in the matrix theory like an addition, subtraction, inverse etcetera. So, I would request you that you also try to revise those concepts from your book on the matrix theory that how you used to obtain them. So, you try to practice it and I will see you in the next lecture till then, goodbye.