Science Communication: Research Productivity and Data Analytics using Open Source Software

Mohit Garg

Central Library

IIT Delhi

Week: 04

Lecture 15 : Lists, Matrices, Factor, Array

In the last Lecture, we discussed two important data structures that are vectors and data-frames. In this lecture, I will be discussing another 4 data structures that are lists, matrices, vectors and arrays. So, what happened is that when we were discussing vectors, this is a vector. Now this is a data-frame. So, in the data-frame, if you see here, this is a vector1 and this is a vector2. So, a data-frame is a combination of vectors.

Now if you see, if we want to store the value like say for example, this is a vector1 of a number of publications and these are the author names. With this, we want to store the value like the total number of publications. So, for example, these are the number of publications. These are the author names and total number of publications, say for example 131. If we want to store this kind of data, we have to use another data structure that is known as a list.

So, we will see a practical example in which exactly we get the error and then we come up with why we need the exact list. So, let us assume that this is a vector researcher. So, I have this vector. So, another vector we have publicationdoc with the details 12, 14, 15 and then we have another vector where we store the two values only. Say for example, these two values are the metrics score of these particular authors.

So, if I will create this vector. It's created. So, now if I create a data-frame of these 3 vectors, let us see what happened. So, if I run this, this is a command to create a data-frame and these are the 3 different vectors. So, if I run this, it says that there are different number of rows. So, 3 and 2.

So, what exactly it means is that in a researcher, there are 3 rows. In pubdoc there are 3 rows. But in metscore there are only two values. So, that is why we need another data structure which we call a list. Let us now take one more similar example.

If in this case we have 4 values in the researcher. So, I am running this and here if I run this. and now again if I go with these values. So, in these 2 vectors we have added one more value. but this metscore has only 2 values. So, if I run this, now if I create a data-frame, it is created. So, what happened here is. So, in the earlier example when we were creating the data-frame with the 3 values in the 2 of the vectors and 2 values in another vector, the data-frame was not created. But here the data-frame is created. So, what exactly did I do here? So, we will see the example here.

So, if you have noticed here that the metscore we have only 2 values, 2 and 3. But when we have created the data-frame, it is automatically added these 2 values in the last 2 rows. But we don't want like this, we want only 2 and 3 for the particular dataset. So, that is why we need another data structure which we call it as a list.

So, a list is another powerful data structure that stores and manipulates heterogeneous data types. It is a highly flexible data structure that can have multiple elements of different data types like numeric, character, logical, etc. Lists are helpful when we have to deal with dataset that have numeric, character, logical or even more complex data structure. So, let us first create some dummy lists and then we will do all the different operations on those lists. So, as we have already discussed, the vector is the key type. So, to create a list, first we will create a vector and then from those vectors we will be creating a list.

So, say for example, this is a vector, researchID is a vector, I am creating like this. Then we have another vector researcher with these 3 researcher names. I am doing this. And now if I say that these 3 authors have a total number of publications 131. So, I am creating another object with the value 131. So, if I run it. Okay. These are the basic things we have created.

Now we will create the list. Okay. So, the basic syntax of creation of list and data-frame is the same. Okay, you have to call the function list and inside the function you have to specify the name of the vector which you want to create the list. Okay. So for data-frame, what we used to do instead of list we used to write data.frame but here because we are creating the list. So, we will use this list function.

In this list function we have these 3 objects. So, what I am doing here is I am creating a list of these 3 objects. So, if I run this here, my list is created. Okay. So, this is the first element of my list, this is the second and this is the third. Now I will assign a name to this list.

So, yes, this is my list. My list is published. So, these are the values of published. Now, we will see the names of each element of the list. Okay. So, if we have to see the names of each of the lists, we will use this names function and if I run this, null. So, if you remember when we were creating the data-frame, as soon as we call the function data.frame and we give the vector's name, it automatically assigns the column name also from taking the vector name. So. But here it doesn't take automatically. Okay, we have to mention that these are the names of the elements. Okay. So, if we see here, that is why this name function is showing the null value. So, now we will add the name of each element.

How will we do that? So, for that there are like 2 ways are there. One way is that when you are creating the list at that time you explicitly mentioned that these are the element names of this list. or afterwards you can also add the names of the list. So, first we will see how we will be doing at the very beginning whenever we are creating the list. If I run this, here are the 3 object names: researchID, researcher and totalpub. and I am giving the element name as ID to researchID, name to researcher, and total to total publication. So, if I run this here again. and now if I check the values of this publication.

Okay, and now if I run it again here this function names. it is showing that these 3 element names are there. But earlier it was showing null. Okay. So, whenever you are creating a list you have to remember that you must mention the names of each element. Okay, now let us assume that you forget to declare the element name in the starting. We can add later on also. Okay. So, how will we add those names? We will be using the name function again and in the name function, we will be saying that these are the 3 element names you assign.

Okay. So, here instead of the same name, I am taking different names. For example, I say it as ID iDoc. then instead of the name I say it as researcher. then in the total I say it as totpub. Okay, if I do like this, it is done. Now if I check the values of this particular list. Okay. So, if you see here the earlier name elements were ID, name and total but now the element names are iDoc, researcher and totpub. Okay. So, this is how you can assign the names to each element of a list.

So, after creating the list you want to access the element of that particular list. Okay. So, to access the element of a list we have to use the same symbol as the dollar symbol. but we have used it in the data-frame. So, we will see how we can use the dollar symbol to access the element and to access the particular value of that element. Okay. So, if I have to access the only name element. Okay. how we will do so we will first call the name of that list. Then we will use the dollar sign then we will mention the element name. Okay. So, if I run it here like this. So, why is it showing null here. So, it is showing null because we don't have an element name as name. Okay, because we have already changed here to researcher. So, if I do a dollar if I run it here like this, it's a researcher. It's not there. So, first I'll just change the names of this list.

So, this is done. Now if I do the dollar name is there. and now if I run this. So, these are the 3 authors who are there in this list. Now if I have to access like this is the way you can also access the particular element. So, the first element of the list is indexed at 1 then at the second and at the third location.

Okay, if I do like this, publish 3. It will give me the total number of publications which is a third element. But now if I say like this, there is no fourth element. Okay, there are only 3 elements there. that is why it is showing null here for this particular element. So, you can access the element by either using the dollar and then you give the name of that particular element. or you can use the location of the element.

Now say for example, you want to access that particular value of that element. Okay. So, for that what you will do. Like say for example, you want to access the particular value like in the name element you want to access only Mohit. Okay. So, what you will do. you will do like this. and if you do like this Mohit is retrieved. Okay, accessing the element in each list will be very helpful during the analysis. So, you must practice on how we can access different elements and their corresponding values in a list.

Okay. Now say for example, you have a list and in that particular list, you want to change the name of one of the elements or either some of the values of the list. Okay. So, it's very easy. how we will do so let us understand first how we can update the list by this example. So, what we are doing here is. So, we have created a list which has the name then we have a total publication.

And then we have an ID. Okay. 123. See these are 131. Instead of locations you want to add another value. What you will do you will just remove this and you will like say for example, you let them okay. So, in the same way we can do in R also. So, modifying a list in the R is very straightforward. So, to modify or to update the value in a particular element of a list you have to first mention the location where you want to change the value of that element. For this particular example we are changing the value of the name element and at the third location instead of locates we want to replace this by a room. So, if I run like this, and now if I see this name element. So, this Tarun is there earlier if you see here earlier Lokesh was there. So, now it is updated.

Now if you want to add more value to the element, you will do the same way as we have updated the value in the element. For example, we want to add at the fifth location. So, we will run this and if I do like this. And now if I see the value of this. Okay. So, now Tarun is at the third location also and at the fifth location also. if you have noticed that it is automatically assigned NA at the fourth location. So, why is it added NA value here, because we haven't declared the fourth location value.

Okay. So, our earlier list has only 3 values in this particular element. and then we are added directly at the fifth location. We have ignored the fourth location. So, that is why it automatically added the NA here. So, a list is such a kind of flexible data structure where you can store the data-frame and the vector in the same list. So, let us now see how we will be creating the list where we will be storing the data-frame and a vector.

So, this is our data-frame we have already created. So, if I do like this. So, our list is created. and I will assign a name to this. And if I now run this. So, now my new list is having a dataset. and in that particular dataset it is having the total publication also.

Okay. So, this is a dummy dataset. Okay. So, this is how we can store the data-frame, and a vector or another object in that particular list. Okay until now we have discussed a single list only. Okay. But we can store the multiple lists also okay in the same list. How we will do that. Now say for example, we have this particular list of institutional publications. So, I am creating these 3 objects. So, if I run this, this is a list 1. So, list 1 is created. And now I have data from another institution. and I will create these 3 objects. And I will create another list. Now to create a nested list or a single list the function is the same. Okay inside the function you have to give the details that whether these are the data-frame or a vector or these are the list. These are the different lists you want to combine as a single list. Okay. So, this is the same function and now I am instead of a data-frame or a vector or 2 different vectors. I am saying here that I want to create a list of these 2 institution lists. So, if I run this, I check the value.

So, now I have a single list which has 2 more lists inside it. So, 1 is this list and the other is this list. Okay. So, this kind of complex list we won't be using during the course. but it is very helpful in knowing how a multiple list can also be stored in that list. Okay. So, this is about the list.

So, now we will discuss another data structure that is a matrix. So, Matrix a 2 dimensional data structure in R looks like a data-frame. So, however, there is a difference between a matrix and a data-frame and so the matrix contains the element of the same atomic data type. Okay. So, we'll see the example. So, if you see our data-frame is this like name, author, publication. So, this is a character and this is a numeric. Okay. But the matrix looks like this. Okay. So, 1, 2, 3, 4, 5, 6, 7, 8, 9. So, the values in the Matrix are of the same data type. Okay, wherever if you see in the data-frame, if you see it looks like a matrix if we see it here it looks like a Matrix like Raman, Mohit, Lokesh then 12, 13, 14.

So, here this is character and this is numeric. Okay. So, this is one of the differences between a data-frame and a matrix. Okay. Understanding matrix computation will help in doing data manipulation, principal component analysis, imageș processing, machine learning, etc. So, we'll see how we can create the Matrix in our. So to create the Matrix in R, we have to use this function called Matrix and there we have to provide the values that are the Matrix we are creating. So, let us first create a matrix in R. For example, this is a matrix I am assigning myMat is a name and these are the eight values. Okay, if I do like this and if I have to check the value. So, this matrix is created.

Okay, if you see here that it looks like a vector kind of thing. Okay. So, all the values are in the single column. Okay. But it is a kind of matrix which has 1 column and there are eight rows. Now if I check the dimensions of myMat. I run this dim function. So, it's showing eight, 1 is so eight is the number of rows and 1 is the number of columns.

Okay. So, this particular matrix is like a kind of a vector which has eight observations. Okay. Now if I check like the number of rows, the way we have checked for the data-frame. So, there are eight rows and the same way we can check for columns also. So, for that we will be using ncol and I'll do 1 so the number of rows are eight and the number of columns are 1.

Now you want to create a matrix. But you want to specify the number of rows and number of columns also. Okay. So,you can explicitly mention when you are creating the Matrix. that these are the number of rows that would be there and these would be the number of columns. Okay. So, take the example here that in Matrix 1 to eight are the values and the number of rows are 4 and the number of columns are 2.

Okay, if I run this and I assign this value here to this myMat. myMat has a matrix which has 4 rows and 2 columns. Okay. But now if you give the wrong number of rows and no wrong number of columns, it will show the error. Okay, let us assume that you are giving like this. Okay. So, you are not mentioning the number of rows you are mentioning that you won't pay for the number of columns only. Okay. So, if you run like this whenever you are creating a matrix in R, you have to remember that the data length should be the sub multiple or multiple of the number of columns.

Okay. So, if you see here. here 4*2, 8 is there with the number of values. If I do this, like if I have 4 these Matrices will be created or if I have 2 here the Matrix will also be created. or if I have eight Matrices will be created. or if I have 1 matrix will be created. But if I have value 7, this matrix will not be created. The same way if we do 4 rows, if I do like this, it won't be created but as soon as you are doing like this. So, here whenever you are creating a matrix you have to remember that data length should be the multiple of the number of rows, or the number of columns.

Okay, as we have seen in the list, we have to declare the names of each element of the list. In a similar way we can define the names of the rows and columns in the Matrix also. So, say for example, our myMat is already created. So, if I run this, this is our matrix. So, there are no names. Okay. So, what we can do is like whenever you have a dataset and you can name it by using this function row names.

So, first we will check whether we have any row names to our Matrix. So, how we will do it is that we will use these row names and if I put myMat and if I check the value, there is null. Okay, similarly if I do for colname and if I run this, null. Okay. So, we can assign the row names and column names to the Matrix. So, let us assume that these 4 values are of the fourth of the authors.

So, I am assigning first, second, third, fourth. So, this is where I have assigned and then I will assign the colname. So, these are the column names. So, now if I check the value of myMat, it will show that this is a matrix and it has 2 column names pub1 and pub2 and these are the row names. Okay, and then you can do the analysis by accessing this particular element of a matrix.

Okay. Now if you want to access the particular element of a list. Okay. So, for that you have to mention the row number and the column number. Okay, say for example, this is a matrix and if you want to access this particular value. Okay. the 5 values. So, what you have to do here is you have to mention that this is a value of the second row and this is and the column number is second. So, this is colnumber 1, this is colnumber 2, this is col number 3, this is row 1, this is row 2, this is row 3. Okay. So, if I have to access the 5, I have to give the exact location that this is a second row and this is a second location.

If I have to access the value nine, I have to mention that this is row 3. So, 3 rows and the column is also 3. Okay, if I have to access the seven, how will we do it? So, we have to mention the third row and the column is 1. Now we will see this in R. So, if I do like this, myMat. So, this 6 is coming. So, 6 is the second row and second column. So, this is the value. Okay. This 6 is a value. or if I have to access the eight, what we have to do is myMat. So, we have to access the eight. So, eight is in the second column and in the fourth row. So, in the fourth row and the second column. If I do like this 2, 4. it will show the error that there is no second row there. But the fourth column is not there, only 2 columns are there. Whenever you are accessing the particular value in the Matrix, you must see how many rows you have, how many columns you have, and then where that particular value is located. Okay, also the format is that first you have to give the location of the row, then you have to give the location of the column. If you reverse it, it will give another value.

Okay, let us say for example, if I do like this first row, second column. So, the first row, second column, this value is 5. but if I reverse it, if I reverse it, this value is 2. So, what it means is the second row, first column. So, this is why you have to be very careful whenever you are accessing the element in the Matrix. Okay, now say for example, you want to access the whole of that particular row.

Okay. So, how will you do it? So, this particular row is accessed. Okay. So, it gives the values of the second row in all the columns and in a similar way if you have to access the column, you can do like this. So, you are not mentioning the row number, you want all rows and the second column. If I run this, this is the second column. Okay. 5678. If you want to access all of the elements of a matrix. Okay. So what can you do? You can do simply this.

So, this is just an equivalent to saying like this myMat. Okay. So, these are exactly the same. But you must give a try at accessing it this way also, because this is how we learn and this is how we do different processes in R like what are the other ways of doing the same thing. Okay. So, this is about the Matrix. Now let us discuss another data structure that is a factor. So, factor in R is used to handle the categorical data. So, it is a type of data which is represented in the form of categories instead of numeric numbers.

Okay. So, for example, this red, green, blue. Okay. So, these are the categories of that particular dataset. Okay. So, this is very important whenever you are handling any categorical data and then you have to use this particular data structure. Okay. So, during this whole course, we will be handling many of the categorical variables like the type of open access model. For example, green open access, diamond open access, then hybrid open access.

So, these are kind of categories. Another example is like the document type publication article review, note or editorial. So, these are some of the kinds of categorical data that we will be using in R analysis. Okay. So, to create a factor in R, you have to use this factor function. Like we create a data-frame, we use a data dot frame to create this list, we use this list function. Then to create a matrix, we use the matrix function and in a similar way to create a factor, we will be using the factor function.

So, if I do like this, these are the 3 factors, green, gold, diamond and if I check the levels. So, level basically the categories, what are the different categories we have created in this particular factor data. Let us take another example, if I do like this, and now if I check the levels of this. Okay. Basically it categorizes that there are 3 different values there, that is green, gold, diamond. But here 2 times diamond is there and also if we see the value. Okay. So, the values of this particular factor data, it has 4 values, green, gold, diamond, diamond. But the levels, the categories are diamond, gold, green. Okay. So, this is how we can create a factor in R.

Now if we convert this factor data as a numeric, let us see how it looks. So, we convert, it is showing that 3, 2, 1, okay. So, it assigned the numbers that this is a third number, this is second number and this is the first number. Now if we check the class of this factor data, it will show that this is a factor. This is how the categorical data is stored in R. So, we will see during the analysis more about how we will be putting different operations on this factor data.

Let us now see our last data structure type that is array. So, an array is another data structure that is used in R. If you remember like that vector. Okay. a vector is a kind of an array which is one-dimensional. then if you see a Matrix, Matrix is a kind of an array which is two-dimensional. So, an array is a kind of data structure that has multiple dimensions, okay. So, whenever you have to create an array, you have to mention the dimension that is the number of dimensions this array has, okay.

So, if you take the example that I am creating an array of one dimension, okay. So, if I do like this and if I run it like this and if I check the value, it is a kind of the way we have created a vector, okay. So, it is kind of an array of one dimension. Now if we create an array of two dimensions, this and now if I check the value of this, if you remember like when we have created the Matrixs, it is the same kind of example. This is also an array of two dimensions.

If we want to create an array of 3 dimensions. Okay. So, we can create it like this. So, if I run this and now if I check the value, this is a 3-dimensional array. So, this is how we can create an array and also we can do multiple operations on these arrays. But during our course, we will be mainly focused on the vector and data-frame. Although some of the functions give the output in the form of a list, at that time we will see how to manipulate those lists, how to unleash those lists to create a data-frame, okay.

So, this is all about the data structures in R. So, in the previous lecture, I have discussed the two of the data structures: vector and data-frame. In this particular lecture, I have discussed the other 4 data structures that are lists, then Matrix, vectors and arrays. I request you all to practice all these 6 data structures by using the code. I will be sharing all the code whatever I have discussed.

In the next lecture, I will be discussing the packages and help and that will be our last lecture of this particular week. Thank you. See you in the next lecture.