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

Mohit Garg

Central Library

IIT Delhi

Week: 04

Lecture 13 : Object and Different Types

Dear Learners, Welcome again to the course. In the last two lectures, we have installed R and RStudio. We have also seen different menus of R interface and RStudio. Why are we using RStudio as an interface of R. I hope you all are now comfortable with the interface of RStudio and still if you have any questions or queries regarding the installation of both the software and please don't hesitate in asking or sharing the questions in the discussion forum. So, this whole course is about the analysis of research publication data and this data looks like this if you see on my slide.

So, this is just a sample kind of a dataset that we will be handling. Here if you see this is text data. So, this is text data. Then this is also a kind of text data. Then this is a number. Then this is text. Then this is also a number. So, if you see in this particular dataset though this is not a complete dataset but we will be handling the data which look like this particular dataset. Therefore, it is very important to know how R understands this particular text data and or number data or any other data format.

The whole concept of data analysis in R is based on one concept which is an object. Everything in the R is considered as an object. So, what is an object? So, this one column is an object. So, this is an object. This is also an object. This is an object. So, these all are individual objects and each object has their like name. So, this particular object is the author. Author is in one object whose details are author details. Then there is another object which is the title which contains the details of the title of the research paper.

So, this is how you see there are different types of objects. Of course this whole dataset is also an object. So, there are like objects, the author is there, object title is there, object year is there and then this whole dataset is also an object. We will see each of the objects one by one then you will have a more clarity about what exactly this object is. So, for the simpler understanding of what an object is you can consider each object as a container where the values are stored and then we do the analysis on these containers.

We can say that this is a container 1 which stores the text data. Then this is another container 2 which also stores text data but this particular container stores the title details. And this is just one row of data. We will be handling thousands of rows for the analysis of this data. So, it is very important to know how the values are stored in this container.

In this particular lecture I will be discussing all these different kinds of containers and how we will be using these containers for the analysis. Syntax for creating an object in R is this. So, on the left hand side this object name will come. And then this symbol will come <- and then the value. So, this is how an object can be created. So, it is very simple to create an object in R. Now let us see another example of text data. So, this is like a number. Now we will see how we can create an object which has a character or text data.

So, here if you see on this particular part NPTEL and then that is the assignment operator and then we have the value 14. So, it is in the same format. So, NPTEL is an object name then this assignment operator then the value. So, what exactly it is doing is R is assigning the value 14 to object NPTEL. If we run this and now if I see the value, the value is 14.

So, this is how an object can be created. So, it is very simple to create an object in R. Now let us see another example of text data. So, this is like a number. Now we will see how we can create an object which has a character or text data.

So, for that we have to keep whatever the text we want to assign to the object it should be in the double inverted comma " ". So, here the NPTEL we are taking the same object NPTEL. and then we are assigning this value "Mohit Garg" and if I run this. So, now Mohit Garg is assigned to NPTEL. Let us take one more example of assignment and another value to object NPTEL.

So, if I do this. Now if I run this value it will not show the previous value. So, earlier we have assigned 14 to NPTEL objects but now it is showing True because presently NPTEL object name has this True value. So, you have to be very careful whenever you are doing the analysis that sometimes you have to create many objects. So, also you have to remember here one thing that I am writing the code here and then I am running the command. So, you have either of the option either you can write here and then you can select that command and then you can click on this run command or what you can do is you can start writing here and then simply give the value.

So, it is totally up to you how you do it but I recommend that it is always a good thing that whatever the command and whatever the code you are writing always write here and save it so that later whenever you are doing this analysis this code will be saved to you. and you will know what all commands you have given during that analysis. Another thing if you have noticed here is that this particular line does not start with a hash. So, what this # represents that generally in programming language we need to write some commands during the code. This is just simply to remember what exactly we are doing. If

I do like this say for example NPTEL 15 and if I run this now what could be the possible value of NPTEL.

So, I request you to pause the video here and just try to think what can be the value of NPTEL here. So, if you see here if I run this and if you check here NPTEL the value is 14. So, what exactly happened here is that R considers only this particular command as a valid command. and this command is also processed in R but the value is not assigned. So, this particular hash is very important whenever you are doing the large analysis we will be seeing all those examples during when we will be doing the analysis of different examples. So, this is very helpful so I recommend that whenever you are doing the analysis you always use this interface for coding and then use this hash for commenting for that like you have declared this variable what this variable is if you have done any operation what that operation is.

So, there are some guidelines and restrictions which are to be followed when you are creating any object in R.

So, the first guideline is that whenever you are creating an object it should be started with either a letter or a dot. What exactly it is other than that if you use numbers or any other symbols in the starting for creating an object it won't accept. So, we will see the examples.

For example this is an object which I named as an author and I just assign 12. If I have to see the value of this author the 12 value is assigned. If I have to do something like this, for this, it will say an unexpected symbol in one particular command. Also if you see why I am recommending that you always know what the commands you are using. You do in this source code. This is because as soon as you make a mistake in writing the command it will automatically show you that there is something missing.

And because this is a valid object name so it is not showing anything here but if you see in this case it is showing that this is an unexpected token. Something is missing. But in case you are doing this, let's say for example this in this case if you do in this console part it will only show the error only when you run this command.

So, that is why the source quadrant is mostly preferred by the R users.

Let us take one more example of creation of an object that starts with a dot. So, if I do like this and if I do this and the value is assigned. So, there is an object created dot author which has the value 12. But instead of dot if I use any other symbol say for example I am using this. So, see here it is automatically showing that something is wrong here. If I run this it is showing unexpected results.

So, this is the first guideline that you have to follow that an object name can only be started with either a letter or a dot. Also here I just recommend that you always start the object name with a letter. Don't use the dot because dots are used to represent functions and methods. Always prefer that whenever you are creating an object in R it should start with a letter.

Now the second guideline is that after letter and period now only numbers, letters or underscore or period can come. So, after that no other thing can come, only these four things can come for creating an object. So, let us take some examples. So, Author123 if I do here and if I assign some value and if I do like this, if I run both this and if I have to check the value it is 12 and if I have to check this value. So, the value is assigned. So, this is a valid object name. So, this is the second guideline you have to remember whenever you are creating any object in R.

Now the third is a very important guideline whenever you are creating any object in R is that an object name is case sensitive. So, what exactly does case sensitive mean? So, an object name in upper case and an object name in lower case are different. If you are doing like this and if you are assigning some values 23 and if you are doing like this to here like 36. So, if you run this here now if you see the value author 23 and author 36.

So, that is why always remember that whenever you are creating any object you follow one kind of writing style either like all caps or lowercase or whatever the style you are using. otherwise it will confuse you. Just see the example here. I will write Au and you see there are a lot many examples there. So, because I have created different object names in different writing styles. Whenever like these there are only 6-7 object names. But whenever these object names are like 20 or 30, it will be very difficult. It is always recommended that you use the object name in one kind of writing style.

Now the next guideline for whenever you are creating an object name but you have to remember is that there are some reserved words there. So, every programming language has some reserved words. So, it is recommended that you never use those reserved words for creating your object. So, if we see this particular example, here Else is a reserved word. So, this is used where we used to do some conditional operations.

So, Else If. We assign any value to Else it will say something like it is automatically showing that something is wrong with this particular command and if I run this it is showing that unexpected error.

So, this is again another guideline which you have to remember whenever you are creating any object in R. Then the last guideline which you have to remember is that an object name could not have any special character like $, & signs and all those things. So, if you will be doing something like this if you are doing it here it will show the error that something is missing. but like if I run this. it is showing that something is wrong with this particular command. So, these are all some of the guidelines which you have to remember whenever you are creating any object in R.

I hope you have got the understanding of what an object is. Before going ahead just give a try on these four object names. Please let us know in the discussion forum that all these object names are valid and what all are invalid object names. So, in these three examples what I have done is that I have assigned the different kinds of values to the same object name. Whenever I am seeing the value of an object it is showing the latest one.

Now let us take the example where what we will be doing we will be creating the three objects with three different values. So, NPTEL1 is one object. So, if I am doing this so this is done if I will do this and if I do this. So, here we have created three objects NPTEL1, NPTEL2, NPTEL3. These are valid object names because it starts with the letter then it has a letter and then in the end it has numbers 1, 2, 3. So, if we have to see the value of what we will do we will simply see the value NPTEL.

So, we will do NPTEL and we can select any other one so NPTEL1. So, NPTEL1 has a value of fourteen. If we see from our eyes these three objects have three different values. But we have to know how R understands it. For that we have to use this class command. So, what exactly it does it gives the exactly the type of value assigned to this object. So, if I do like this and run this it is showing numeric. So, this fourteen is a numeric value which is assigned to the object. So, NPTEL2 if I check the class of this it is a character. So, it is correct that the NPTEL1 is an object which has the numeric value NPTEL2 has an object which has character value.

So, here I request you again to pause the video here and then just you think in your mind what could be the possible class of NPTEL3. If you have given the answer character it will be wrong. So, why let us see now here. So, if we see logic here, it is logical. So, NPTEL3 objects have value True which is a kind of logical type.

Now you will say in NPTEL2 we have assigned the character value under the inverted comma but here we have assigned a value which looks like character but we haven't mentioned those inverted commas. These True-False are like logical values. We will see what these values are. So, in R there are six kinds of these value types which can be assigned to an object like one is a numeric, then another is integer, third is character, fourth is logical, and there are two others are complex, and row. complex value type is basically used for doing the analysis on complex and row is basically for row vectors of bytes. But in this whole course we will be mainly doing the analysis on four kinds of value types: numeric, integer, character, and logical.

So, I will be discussing only these four types in this lecture. So, the first value type is numeric. Numeric is a value type which contains all kinds of numbers with or without decimal points. We will understand this by examples.

So, let us take the example. So, I have created an object NPT_num and if I am doing this I have to check the class so it is saying it is numeric. So, this is a decimal number then if I have assigned a negative decimal value. Let us now just assume we will add this number in this. Just to see whether R has understood that this is a negative number. So, if I do this it is showing 0.2. What exactly happened here is that we have assigned this value 1.2. Then in the second command what we have done we have added 1. So, the answer will be -0.2. So, that is why it is shown here -0.2. If we see the class of this NPT_num here it is also numeric and if I assign the 1 so again this is also a numeric. So, a numeric is a value type where it can contain all kinds of numbers with decimal point or without decimal point.

Now the second value type is integer which is without any decimal point. So, for assigning the integer value to an object you have to explicitly mention L after the number. So, like say for example if I am doing this and now if I check the class of this it will show an integer.

So, if you do not write here L like say for example if you do not write L here and if you do like this and now if you see the class numeric. So, it is automatically changed to numeric. So, if you are thinking that this 1 is an integer number. So, you have to use this L after the number. So, now let us take one example with a decimal number and then we will convert it to an integer. So, say for example NPTEL_int is our object name and we are assigning the value 1.9 and then this. and now we are converting it to an integer. So, if we see the class of this NPTEL_int we have just the object name but we have created just now it is numeric. This particular value is numeric but we want to have value as an integer. So, we will convert as an integer. So, to convert that numeric value to integer we will be using the command as integer. So, we will convert this to an integer and this NPTEL and we will run it again and now if we see the value.

So, what could be the possible value? So, you can just pause here and just you can think on what could be the possible value. So, if you have thought this value is 1 so that is correct. So, as an integer it will not take the decimal value it will take only the integer part it will ignore the decimal part. Say for example you have citations. You have a citation kind of a data and that are all like 12, 13, 14 these are citations of an author are going and you are doing some analysis some average and also you want to say so it like as a analysis it comes like something value like 5.9.

Now citations cannot be in the decimal part. Citations will always be like 5, 6, 7, 8 so it will be either 5 it will be either 6, 7, 8, 9, 10, 11 like that. That is why the understanding of integers becomes very important whenever you are doing the analysis. Now take the third value type which is a character which we have stored in the inverted comma “ ”. So, let us take the example here.

So, character value types store the data which is in a character form. So, let us understand this by an example. Here I am creating an object NPT_car which has a value welcome to MOOC. So, if I run this, if I check the class of this it will be character. Now if I have to check the value of this it will be an NPT_car and then if you run it its welcome to MOOC is assigned. But you might be thinking, Why are we using the double inverted comma? What will happen if we do not use the double inverted comma?

So, let us take that example also. So, if I do like this, of course if you are doing it here it won't allow it to automatically show you that there is something missing. but for the time being I am thinking that I am correct. I am ignoring this suggestion and I am running it so that it is correct. So, if I run here it will show that unexpected symbol. This is why whenever you are creating or whenever you are using any of the character dataset, you have to use this in the double inverted comma. Also if you notice that if you see the output here for this particular object it is showing the value is 1. But if you see here the output is also shown in the double inverted comma.

So, this is why whenever you are creating any object which is assigned the character value you have to use double inverted commas. Now move to our next value type which is logical. Logically so logical value type basically becomes important whenever we are doing the analysis.We won't be creating an object which has the logical value type during the analysis but the output of some of the analysis can be logical. So, let us now see some examples of logical value types. So, as the word suggests, as you know, logical values can be either True or False.

So, that is why whenever you are declaring or getting the output as a logical type you will get either T for True or True or you will be getting F for False or False as a full return. So, if you see here if I do like this NPT_log T and if I check the value it is True. So, automatically I have assigned the value only T in the cap it automatically takes. That is a True value. It is a logical kind of thing and if I do like this. So, what will happen if I do like this? Now see what will happen if I do like this and if I run this it is showing that object True is not found.

So, for declaring the object under a logical value type you have to use either T in the caps or True as a cap. If you write True in this way it won't accept it will think that you are assigning some character value to this object. So, if I check the class, it is logical. Okay, let us see how this logical value type becomes important during the analysis. Say for example I have created an object named as Oath1 and assigned the value 14 if I run this it will be this.

So, the value is assigned 14. Now if I have to see whether the value assigned to 14 is greater than 10 what we will do we will run this it is True that means whatever the value Oath1 has it is greater than 10. If I have to check whether it is lower than 10 we will use the symbol lower than symbol and if I run this so it is False. So, if you have to check whether the number is equal to 10 we have to use it like this and if I run this it will say False. Of course 14 is not equal to 10 so that is why it is showing False. So, if we see these three commands only one is True where it is showing that Oath1 is greater than 10 okay and other those two commands are False.

This we will see when we will be doing the analysis and it will be like in various cases the output will be useful for doing the analysis. So, this is a common problem whenever you are doing the data analysis there will be some of the value missing, some of the value will be mentioned NA okay. So, that is if you have a big dataset and if you want to check whether the dataset has any NA values. okay like say for example I am assigning NA value to this Oath2 NA. Now Oath2 has the value NA okay. Now again here you will say that for NA also we have not mentioned in double inverted commas by default it is understandable that if NA is there it is not available that missing value is there okay.

Now if I have to check whether any NA is there in Oath1 what we will do we will use this command is.dot NA Oath1 and it will be False because Oath1 has the value 14 there is no NA is there but for Oath2 if we check NA okay. So, if you have a large dataset it will give you the idea that where NA values are there where NA values are not there this is how the understanding of logical value type helps during the analysis. if you have a large dataset and if there are any missing values or NA values are there so you can check the presence of NA values in a dataset using the command is.dot NA.

So, the last thing is that whenever you will be doing the analysis you will be creating a lot many objects to know what all objects you have created during the analysis. okay so that you can do it using the command LS. if you run this LS here and if you run this so it will display all the object names whatever you have created okay. So, now for example in these object names you want to remove these object names for any reason. okay so for that you can use the command RM which helps in removing the object okay. So, how you have to do that it's very simple. you have to use this RM and under this you have to like say, for example I want to remove value objects.

ṣokay if I run this okay and now if I do LS again to list all objects. It's showing that it's okay. So, earlier if you see here the value was showing so earlier if you see the value was showing as 45 but now it's not showing okay now say for example if you want to remove all the objects. So, for that you have to use what exactly it is doing first LS will list all the objects. and then RM will remove all those objects. So, if I run this now if I do LS so now you will see there is no object name. Now if I do like this there will be no auto suggestion. okay because I have deleted all the objects. Be very careful whenever you are removing the object's name using this command. Now you will be a little worried about how to remember all these commands. So, you don't have to remember all these commands. So, many of the commands are just straightforward. So, if we see that for this particular task what we have to do is we have to check if there is any NA in that particular dataset. So, if you see that particular command there is also not any. Okay or if you see here. So, for this particular command we want to check the class. What type of value is there so for that command is class.

So in this lecture I have discussed the objects and how we can create an object in our what are the things we have to remember whenever we are creating any objects in our what are the different value types we can assign to the object. So, we have mainly focused on four of the value types. That is numeric, integer, character, and logical.

I hope you get the understanding of creation of these value types in our and here I request you to please create two objects. One is the object which stores your name and another object stores your pin code of your area. And please let us know these two objects in the discussion forum. We will be very happy to see what all objects you have created. In the next lecture I will discuss different data structures. Thank you and see you in the next lecture.