Multivariate Procedures with R

Prof. Shalabh

Department of Mathematics and Statistics

IIT Kanpur

Week – 02

Lecture – 06

Calculation with Data Vectors and Built-in Function

Hello friends, welcome to the course Multivariate Procedures with R. You can recall that in the last lecture, we initiated a discussion on how different types of arithmetic operations are handled in the R software and how R works on them. We have taken examples of very simple calculations, just like a calculator: addition, subtraction, division, multiplication, and power operations. Now, the strength of our software is the way it handles the data vector. What is a data vector? You can recall that if you want to combine different values in the R software, you have to use the command 'c' (lowercase 'c'), and the data vector helps the R software understand or informs R that it has to perform the operation on each and every value inside the data vector. So, now the next question arises: how does R work with data vectors? That is what we have to understand.

This understanding will help us when we try to write our own programs. Our programs are based on our needs, and they depend on us. We have to decide how they should be completed. It is not R, but we have to follow the rules that R understands. Because they have been built into the software. That is why, in this lecture, it is important for us to understand how the R software works with data vectors.

And based on that, you have to do your programming to achieve what you want from R. So, let us take several examples and try to understand how R works. So, let us begin our lecture. Now, in this lecture, we are going to cover two topics. One is data vectors, and another is built-in functions. But let me first address the data vectors topic. Now, you understand what the data vector is. So, there are here 4 values if you try to see 2, 3, 5, 7, and you have used here a command here c. So, this is actually a data vector. One point where you have to be careful in understanding is that vector is a very general word and is used in different subjects in different ways. There is a vector in physics, there is a vector in calculus, there is a vector in mathematics, etc.

And there is a vector in a matrix also. But this is actually a data vector. The only difference between a data vector and others is that a data vector simply tries to combine different values in the same order in which they are written. For example, if I say c(2, 3, 5, 7), then R will try to operate first on the first position number, then on the second position number, then on the third position, and then finally on the fourth position. And this is different than if I write 2, 5, 3, 7. So, they are not the same. That is what you have to keep in mind.

So, whenever I use the word data vector, that means there are some numerical values which are combined with the c operator. So, now, first, let me try to take how mathematical operations are done. When one component is a data vector and another component is a scalar, well, just for the sake of understanding, I am going to take here only a very simple example where there will be only two values in general, So, that I can explain to you how R is going to work. So, and this I am going to do with the help of several examples. So, if you try to see here, I am taking here an example c(2, 3, 5, 7), and scalar 10 is going to be added.

Now, the outcome here is like this 12, 13, 15, 17. Next question come, what are you doing such that this outcome is coming? So, actually this is what is happening that if I try to write down here c(2, 3, 5, 7) + 10. So, this is a scalar.

So, this part, this is operated over each of this number. So, basically this happens like this 2 + 10, 3 + 10, 5 + 10, 7 + 10. That means in a nutshell what I am trying to say that this operator is going to be operated over each and every member in the data vector. and that will be the outcome. Here in this case, in this first case of explaining you the addition, please try to understand the basic logic and the same concept will be replicated with other operation like a subtraction, multiplication, division etc. So, here in the first part in the first case I will try to give you details in more but after that I will be faster So, and then you can see here this is here the outcome in the R console, right.

So, now the next question comes here if I have two components and which are data vectors and another part where you have to be cautious is that size of the data vector. Size of the data vector means the number of elements in the data vector. So, first I am trying to take here a case where the number of elements in each of the two data vectors which are going to be added is the same. So, in each of this data vector, there are four elements, four numbers, right.

So, now how does this operate? So, let me try to explain it to you. Please try to understand it. I will go pretty slow. The addition in data vectors is done element-wise or, say, position-wise. The first element of the first data vector is added to the first element of the second data vector, like this. Then the second number, or the number at the second position in the first data vector, which is here 3, is added to the number at the second position in the second data vector. So, this and this 3 and - 3 will be added. And similarly, I am trying to use here a different color pen So, that you can understand it.

The number at the third position in the first data vector is added to the number at the third position in the second data vector. So, this 5 and - 5 will be added. And finally, the last or the fourth element in the first data vector is added to the fourth element in the second data vector. So, if you try to see this operation, this is going like this: 2 + - 2, 3 + - 3, 5 + - 5, and 7 + 8, right.

And this + operator is being replicated from here, right? So, you can see here this outcome comes out to be 0, 0, 0, and 15. It is not difficult for you to understand. But if I try to give you a quick revision, it is something like this: c(2, 3, 5, 7), and if I try to write down here - another data vector c(- 2, - 3, - 5, 8). So, we try to add it here.

So, this and this get added, this and this get added, these two numbers get added, these two numbers get added, and we obtain the outcome here. So, this is how the addition works. Now, when I talk about subtraction, division, multiplication, etc., they will also follow the same rule. Only this + operator will be replaced by the respective operator. Now, in this case, we have taken two data vectors of the same size.

Now, I am trying to take here data vectors of two different sizes. You can see here, this is my data vector 1, which has four elements. And I have here a second data vector which has two elements. And if you try to execute it, the outcome will be like this. So, now this creates confusion in the beginning about what R is doing So, that this outcome is coming.

So, let me try to explain you. What happens that if I try to write down here four numbers here c(2, 3, 5, 7). This is my data vector 1 and the second data vector here is c(8, 9). Now, after this, when the operation starts, then first this 2 and 8, they get added and answer comes out to be here 10. Then similarly, the numbers at the second position in the two data vectors they get added and we get here the value here 12. Now, after this when

R tries to move forward, then it does not find anything here, yet what is to be added and in such a situation what R does, it will try to replicate the same numbers again. So, this 8 and 9 will be added here automatically and then 5 + 8, 13 and 7 + 9, 16. This will be the outcome which we are getting here.

So, this operation is happening when we have these two data vectors of different lengths, but there is a prime condition. If you try to look at the number of the elements in the two data vectors, they are the exact multiples. Exact multiples means two is here So, and here it is 4 So, 2 to the 4.

So, first of all, if I try to explain you here this c(8, 9) this comes here and it try to operate over 2 and 3 then c(8, 9) comes here and it try to operate over 5 and 7. So, this is how the R works with the data vectors. Now let me try to give you an example that where the number of elements in the two data vectors are not exactly in the format of exact multiple. For example, you can see here this first data vector this has four elements and the second data vector has three elements.

Now, in this case the outcome comes out to be here like this 10, 12, 15, 15, but it always gives you a warning message that the longer object length is not a multiple of the shorter object length in this one and that is correct the shorter object length is has value 3 and the longer object length has the value 4 there are 4 elements and 3 elements, and 4 is not a multiple of 3. This is how we are going to interpret different types of messages which R is trying to give us. Right.

But now the next question comes, what is really happening and how this object is fulfilled? and how this outcome can be justified. What R is doing? So, now let me try to take a similar approach and if I try to write down here the four values 2, 3, 5 and 7 here and if I want to add here another data vector, it will be here 8, 9 and 10, right. Now, as R starts working on it,

In the first place, what does it try to do? It will try to add the numbers which are at the first position in both data vectors. So, 8 + 2 becomes 10 here. Then R moves forward and tries to add the numbers at the second position, which is here 9 + 3, resulting in 12. Then it moves further and finds the numbers at the third position in the data vector: 1 and 2.

And it tries to add here 5 + 10, which is 15. Now, after that, when it moves further, there is no number here. Now, once again, R gets confused, So, what it tries to do is repeat the same numbers again: 8, 9, and 10. So, this 7 and 8 get added here, and it comes out to be

15. But now there is no number here in the first data vector where 9 and 10 can be added, So, this is not executed, and that is how this outcome 10, 12, 15, 15 is obtained here. Right? So, you can see this is a very peculiar way of doing calculations inside the R software, but I promise you this makes our computation and calculations much simpler when we try to write the program.

Now, in case I try to repeat it for this subtraction, the same concept applies. When I try to write here one data vector subtracted by a scalar, then this - 2 is going to operate over each of these numbers. And this outcome is obtained: 0, 1, 3, 5, which is the result of this execution. 2 - 2, 3 - 2, 5 - 2, and 7 - 2. Similarly, if you try to take here two vectors of the same length, then, as we have seen in the case of addition, element-wise addition happens.

So, here the element-wise subtraction is happening. This number 2 and this number 2, which are at position number 1 and 1, they get subtracted here like this. Then the second element in the two data vectors, 3 and - 3, they get subtracted here. Similarly, the numbers at the third position in the first two data vectors, data vector 1 and 2, 5 and 4, they get subtracted here like this. And finally, 7 and 1, they get subtracted, which are at the fourth position in the two data vectors.

So, if you try to see, this is the same operation that was happening in the addition also, but the only thing is now we have a subtraction. So, this subtraction operator is being executed. Now, similar to the case of what we have done in the addition, let me try to take here the next example where the first data vector has four elements and the second data vector has two elements So, that the number of elements in the shorter vector, they are exact multiples of the numbers in the longer data vector. So, the same operation comes here that this c(1, 2) gets operated first over this c(2, 3), and then after that, once again, it comes and it gets operated over 5 and 7.

So, this 2 - 1, 3 - 2, 5 - 1, and 7 - 2 that gets operated, and we get here this number. Right, and you can see here this is the outcome of the same operation. Right, and now in case if you try to take here two data vectors in which the first data vector has, suppose, four elements and the second data vector has three elements, So, definitely it is not the exact multiple. So, what will happen? The same thing we did earlier: c(2, 3, 5, 7) with c(1, 2, 3), and now it will be subtraction. So, this and this will be related to - 1 as 1, 3 - 2 here is a 1, 5 - 3 is a 2, but now there is no element over here in the last position in the second data vector. So, this 1, 2, 3 will be repeated.

So, this will become here 7 - 1, which is here 6, and this data outcome, which you can see here, is obtained, but a warning message is also given that you have to be careful about what you are trying to see. Well, in R, there are two types of messages that come: one is warning messages, and the other is error messages. So, it has the same meaning as the literal meaning of warning and error. For example, we always warn our students: if you do not study well, then you are going to fail, but we do not do anything after this. However, during the exam, when a student scores below the threshold mark, then the student fails. So, that is an error.

After that, we cannot do anything. But a warning is something like an alert that has to be corrected. So, R is trying to give you a warning to indicate that there is something which is not correct. But you have to see whether it was done intentionally or if it is a mistake. Now, as a programmer, I have to see whether this is what we want or if I have made an error.

And then we have to make a proper decision, and then things will be done. So, now before I try to take the multiplication example, let me try to show you these things on the R console also, So, that you gain confidence. So, if I try to take here, suppose, some elements—here, some data vector: 2, 3, 4, and 5 and if I try to add it here with another number, say, here, 10. So, you can see here that this means 2, 3, 4, 5 every number has been added with 10, and the outcome here is 12, 13, 14, 15. Similarly, in case if I try to take here some data vector suppose if I try to see here a data vector 1 and 2 So, now this is an exact multiple, So, the answer will come out to be 2 + 1, 3 + 2, 4 + 1, 5 + 2. And if I try to take here another dataset of size 4 with elements, say, 1, 2, 10, and 20, then you can see that the respective

Positions are going to be added by the respective numbers. And if I try to make it say in such a way that the length of the shorter data vector is not an exact multiple of the longer data vector, then it gives us a message that, okay, you are trying to do something. Please try to be careful. If there is a mistake, please correct it. Otherwise, leave it.

Now, similarly, I can take the example of subtraction. Suppose if I take here c(2, 3, 4, 5) and if I try to subtract here, say, 5, you can see here this 5 is going to be subtracted from each and every element in this data vector, and this will become (2, 3, 4, 5) with every element being subtracted by 5. Now, similarly, if I try to take here another data vector, say c(1, 2). So, now the respective positions are going to be subtracted, and it is going to be repeated. So, you can see here 2 and 3 subtracted by 1 and 2, then once again 4 and 5 being subtracted by 1 and 2. And similarly, if I try to take here the data vector of size 4, then you can see here the respective positions are subtracted, and we have this type of answer. And if I try to make the number of elements in the second data vector not an exact multiple of the first data vector length, then it will give me a warning message, and the outcome will behave like this, right?

So, this is how you can see that whatever I explained to you, the exact same thing is going to work inside the R console also. Right. So, let me try to come back to our slides and let me try to give you some idea about other operations. So, now the same thing will happen with multiplication and division also. So, you can see here this c(2, 3, 5, 7) when it is multiplied by scalar 3, then this 3 is being multiplied over each and every number: 2 * 3, 3 * 3, 5 * 3, 7 * 3.

Right. And we get here this number. Right. And similarly, if you try to take the two data vectors of the same length, then 2 and 2 - 2 are going to be multiplied. 3 and - 3 are going to be multiplied, 5 and - 5, they are going to be multiplied. And 7 and 8 are going to be multiplied. So, again, once again, this multiplication is happening with respect to the position number. First position with the first position, second position with second position and So, on. and these are here the screenshots So, that you can be confident that if you try to execute them in the R console then you can do it easily right and similarly as we have taken the example in addition and subtraction if i try to take a two data vectors in which once length it is is four there are four element and the second data vector there are only two elements So, the length of the smaller data vector is an exact multiple of the length of the second data vector

And then again, the same thing happens if I try to explain you 2, 3, 5, 7 and see here see 8 and 9. So, now there is multiplication. So, the numbers at the first position they get multiplied, numbers at the second position they get multiplied. Now there are no more numbers. So, this 8 and 9 are repeated and the respective multiplications at the respective position elements takes place.

And if you try to take here one more example where the number of elements are not the exact multiple, then in that case, as usual, this 8, 9 and 10, they are going to be operated on the first three elements, 2, 3, 5. And after that, this 8 will be operated over 7. But now, there are no elements where this 9 and 10 can be operated. So, you would get here a warning message and the outcome here will be like this.

Right and these are the screenshot of the same operation. So, you can see here that it is not So, difficult to understand provided if you have understood the addition and subtraction. Now, we try to consider the division which is also being done exactly on the same logic as addition, subtraction or multiplication. So, if I try to take here data vector here c(2,4,6,8) and if I try to divide it by scalar say 2, then this 2 is going to divide each and every element in the data vector and we get here this outcome 1,2,3,4. And similarly, if I take here one data vectors of size 4 and another data vector of size 2, then this division is going to be happening over the first two elements, then once again over the remaining two elements.

Then the logic is the same which we have used in the addition, subtraction and multiplication and you get here this answer. So, this is the operation which is happening here. Now, similarly if you try to take here an example where the number of elements in the first data vector are 4 and the number and in the second data vector it is 3. So, this is not the exact multiple. So, first data vector is having 4 and the second data vector is having 3 elements which are not the exact multiple.

So, you will get here a warning message as happened in the earlier cases also and yeah this is going to be the outcome. And this is here the screenshot of the same operation, which I have shown you here. Right. So, you can be assured that the same operation is going to happen. Now we come to our last operator, which is the power operator.

This is again the same. what was happening with the other three symbols. Suppose if I try to take here one data vector having four elements, and if I want to take the power operator 2, that means each of this number is going to be squared like this, and this is here the outcome. And instead of array hat, if you try to use here two stars also, that will also give you the same outcome.

So, you can see here just like other operation, this power operator is going to work on each and every element in the data vector, right. And similarly, if you try to take here two data vectors, one is of say length 4 and another is of length 2, then this operation will happen here, this 2 and 3 will come here and once again it will repeat as 2 and 3 here and this is here the outcome, right. So, this operation is being operated here. And similarly, if you try to take number of elements in the second data vector, which is not the exact multiple of the elements in the first data vector, then this power operator will come here and it will make a 2, 3, 4, this will operate and then it will start repeating again 2, 3, 4, but after 2, it does not find any elements. So, it will give you here a warning message and the outcome will be displayed here.

So, this is the screenshot of the same operation, which is being done here. right So, now let me try to show you these operations on the R software also So, you can see here if i try to take here the multiplication if i try to take here say any you say this data vector here two three four five and if i say here divided multiply by two this is the number and if i try to replace it by uh the scalar by uh data vector two and three.

So, this is the element phase multiplication and it is here like this and then in case if you try to take care the same number of elements in both the data vectors once again they will be multiplied by the respective elements and if you try to reduce here the length which is not the exact multiple then you will get the error and you can see here that the same operations I can do with the division also if you see here if I try to take the first data vector and second as say a 2 then the 2 is going to be divided is going to divide each and every element in this one. And if I try to take here the second data vector which is the exact multiple of the length of the first theta vector then the element wise division will happen here like this and if I try to take here the the data vectors of the equal length then it will be here like this and if I try to remove here one element So, that the lengths are not exact multiple in the two data vectors. Then you will get here then warning message also, right.

Similarly if you try to take here the power operator So, you can see here if I take the first data vector and second as a scalar this is going to operate it over the each of this number and you can and if I try to take here the second number as as data vector of the multiple length then it is going to be here like this and if I try to take here see here the two data vectors of the same length The power operator is going to be like this and if I try to reduce the length in the second data vector which is not exact multiple, then it is giving us a warning message and the answer is here, right. So, you can see here it is not a very difficult thing, but the main thing here is to understand it what are we going to do. So, after this small introduction to the calculation, let me try to give you here one more idea which is a very strong feature in the R software. We have here some functions which are built-in functions.

Built-in function means that they have been created by some authors and they are hidden inside the software and if you want to do something for which the built-in function is available, then you need not to do the whole programming, but you can directly use it.

For example, if I want to find out the arithmetic mean, So, arithmetic mean of x1, x2, xn, these numbers is divided, is defined as summation of all the number divided by the number of elements. But now you have two options if you want to compute this arithmetic mean, that you write the whole bunch of programs and then you try to use it or you simply call them by mean of here x, where x is containing all the data points x1, x2, xn. Right.

So, each of this built-in function, they have been designed to do some specific tasks. And that is the advantage that when we try to do the programming and suppose we want to find out the automatic mean, I can simply call the function mean of x and it will give me the data vector. And if I want to find out the square of mean, then I simply have to write down here mean of x and hat 2. That is all. So, this built-in function also works with the usual mathematical principles and they work with the data vector as well as the scalar.

So, let me try to give you some examples So, that you can understand it. For example, if you want to find out the minimum of some number. So, for that the command here is min and if I write down here min and inside the parenthesis I try to write down the value. Here, I am trying to do something special, which I will try to explain to you. I am trying to write down the same command here, but once I am using the operator lowercase c, and here I am not, no c operator is used here, and I want to explain you something.

So, please have patience and try to understand what I am trying to say, and do not get confused. So, if you try to just operate this function here, min, and write down the numbers inside the parenthesis, it will give you here the minimum value, which is - 7.8. But here I have not used the C operator that you can see. But now I try to use here the c operator and yeah, I try to give you here this the same numbers answer is coming out to be the same. And this is here the outcome.

So, now it raises a question here: what should I do? Should I use the mean, this c operator, or not? Well, I would like to inform you that this is happening only with some of the functions, and some functions will take the values only when they are in the form of a c command data vector. For example, I can give you the example here: mean.

You have to give the values inside the parentheses using the c operator. And if you try to give the value here, c, without giving any c operator, but only the value, then it will give you the mean of only the first observation, which is itself only the first observation. Now, here will be a bigger trouble. This answer is correct, and this answer is wrong. But when you are trying to write down this function inside the program, then whatever is the outcome of the wrong answer, this will be carried forward, and your entire program will become wrong.

It will give you a wrong value. So, on the other hand, if you try to use here the c operator, then whether this is mean or, say, minimum, both will give you the correct value. So, my advice to all my students is that please always use the c operator without any exception. No exception just use the c operator to combine your data values, right?

The same thing happened with the maximum also, but anyway, it is difficult to decide which of the functions is doing what. There can be different reasons, but I'm not going into those reasons why this is happening. But I wanted to inform all my participants and students that they have to be watchful and careful when they are trying to write down their program. Similar to minimum and maximum, we have many other functions, such as if you want to find out the absolute value, there is a function ABS. If you want to find out the square root, then the function here is sqrt. If you want to perform operations like rounding up and down, round, floor, ceiling, etc., there are functions for that. If you want to find out the sum and product of the numbers inside the data vector, you can use the commands sum and product. Different types of logarithmic functions are defined by log, log10, log2.

And exponential functions, trigonometric functions: sine, cos, tan, and then cosec, sec, and cot are defined by sine, cos, and tan respectively. Hyperbolic functions like sinh, cosh, tanh, etc. They all are available. And there is a long list. I'm just giving you an example here of what is available.

But my one advice to you all: suppose you want to find out the log. Sometimes we simply write log, then you have to see whether this log is giving you the value which is base 10 or natural log e. For that, I always prefer that first look into the help menu and try to see what the log function is doing, and then you try to do it. So, let me try to give you here some examples of how it works. So, for example, if I write down here absolute of -4, then you can see here it is coming as a scalar, there is only one value. So, there is absolutely no issue. But when you are trying to take here the data vector over the absolute value, now this absolute function is operated over each of these values, and then it is the outcome.

The absolute values of - 1 and - 2 are 1 and 2, and So, on. So, this outcome is obtained here. Now, similarly, if you want to find the square root function, the square root of a scalar, say 4, is 2 here. So, without writing any program, you can simply directly write sqrt. And if you use this sqrt function over a data vector, the square root function is applied to each element, and you get the correct answer here.

And here is the screenshot of the same operation. So, you can see here that these functions work over scalars and data vectors. Similarly, if you want to use sum here, So, if you try to use sum over the data vector 2357, it adds all these values. Similarly, if you want to multiply all the values inside the data vector, then the PROD function, when used with the C operator, multiplies all the values together. And we know that the round of 1.23 is 1, and the round of 1.83 is 2 because it is more than 1.5.

So, these are the different screenshots by which you can see here. And now, I will just try to give you a quick illustration inside the R software. You can feel more confident. So, suppose if I see the absolute value of - 2 here. You can see here that this comes out to be 2.

And if I try to write down here a data vector C of here. - 2, - 3, and 4, and here 5. So, you can see here this will come out to be 2, 3, 4, and 5, right? And similarly, if I want to find out the square root of 9, this is here 3. And if I want the square root of our data vector, say here 4, 9, 16, 25, and So, on, it will be here like this: 2, 3, 4, 5. So, you can see here that these functions are going to work over the scalar value and the data vector both, right. Okay, So, now we come to an end to this lecture. You can see here I tried my best to give you a fair idea, a quick idea of how R works with the scalar and data vectors over different types of operations. The concept is the same.

So, if you try to understand the first example where I have illustrated the addition with the scalars and different types of data vectors, the same concept is being applied to other operations like multiplication, division, power operator, etc. And another big advantage which I explained here is the use of built-in functions. Actually, these are the couple of features because of which R became a very popular software that instead of doing or writing long programs for finding out sums, sums of squares, products, etc. You can just write them in a single line.

For example, if I want to write a function for the sum of xi squared, meaning square the numbers and then add them, I can simply write as sum inside the parenthesis x hat 2. That is all. Whereas if you want to write down the whole program, that will be a longer program. By using the basic concept like sum equal to 0, num equal to 0, which you possibly used to do in your childhood.

So, these are the very peculiar features of R software because of which R became popular. So, my request to you all will be to please try to take up some examples and practice them. I know you already know, but it is always better to prepare yourself and brush up your knowledge before going on the journey. So, we will start our journey with the statistical tool very soon. Until then, try to brush up your knowledge and get prepared for the journey.

And I will see you in the next lecture with more concepts on this computation. Till then, goodbye.