[SQUEAKING] [RUSTLING] [CLICKING]

HELENA VAILLICROSA: In this video, I'm going to go directly to R. So what is R? And what are the advantages of using R? R is a coding language that's broadly used in some scientific areas-- for example, ecology or climate sciences. And why is that?

Well, first of all, R is free of charge, meaning that everybody can use it. And as long as they have connection to the internet, they can download it. And there's no subscription at all. Second of all is that it's collaborative, meaning that everybody can share their code through the R platform itself or also by alternative sources. So this code can be shared. And there's no need to start from scratch.

And third of all is that it can be repeated or shared. So as I said, you can get your script and share it to somebody else. You can publish it along with your research. Fourth of all is that you can go from the head to tail to the process of the research. So that involves the data cleaning processes, also the statistical processes and the GIS, meaning that you can also use maps on R. And finally, you can produce nice plots, so you can communicate your research.

So in this video, we're going to go through the first time we face R. So once we have downloaded R and we have downloaded and installed also RStudio, this is what we see. You might be surprised because your interface might have a different background. It might be a white background. And the font would be black. But that's something that can be changed. By default, it appears like that. But my personal preference, it's to have it in black, so it's not as violent for my eyes. So this is something that you can change as well, if you'd like.

And here are the different panels you'll find. The first panel here is the one where you are going to write your script. So all the different orders you're going to give the computer are listed right there. And whenever you ask the computer to quantify or to execute an order, it's going to be displayed right here.

And then in these two different panels, we're going to have, in the top one, the data that we have stored, the objects that we have stored and created. And here, we are going to have different things, which could be the files in our computer, so we can navigate through that. Also, we could show here the different plots that we have created.

Here are the packages. We're going to go through the packages-- no worries-- in the future. And also, this is very important. This is going to be the Help tab. So whenever we face a new function that we are not familiar with, we can read some description of what each function is doing and how actually we can use it.

OK. So let's go to the code itself. Now, I'm going to be presenting the different types of data that we can have. So there are the different data types. Then we're going to move through the vectors. We're going to go to the matrix followed by the data frames, and finally the list. And I'm going to wrap up the whole section by challenging you with some exercises, so you can practice what we've been talking about.

So R works as a big calculator. So we are going to be assigning numbers to different variables. So the assigning sign is this arrow here. So this is the variable we're going to be creating. In this case, it's going to be the x. And we're going to assign the value-- or the number, number 2, to that x.

How can we execute an order? So there are two ways of doing it. The first of all is to point at the line we want to execute. And then if you have a Mac, you can click on Command and Return. So if you do that, you see now that this piece of code has been executed here. And we have created the x right here. And it's assigned to a 2.

The other way of doing it is to just highlight the line we want to execute and click into Run, right here. That's especially beneficial if we want to run different lines of code. So if I would like to execute all these at the same time, I would just highlight everything and click on Run.

You can also see that there are some fonts here that are in gray. So they don't appear as bright. That's because they are annotations. So this is not meant to be executed. So there's nothing related to this code. And you can create these annotations by locating it right after this sign here.

So also, to follow up with the assigning of values to the x, if you rewrite the x with another value-- we are just going to smash the previous 2. So that 2 is going to be a 3 instead. And we can just use this value we just created to assign it to another value.

So now, we have this y right here. So to create the outcome of this variable, we're using the previous x that we just created. And we are assigning it to that value here. As you can see, now, we have 2. The x and the y. The x is a 3. The y is a 13. And we can keep going on, using different mathematical expressions right there.

Also, this is the first function you are seeing. This means exponential. So we're going to be calculating the exponential of a 2. In this case, the value we want to be targeting is going to be between parentheses right here. So if we run this piece of code, again, with pressing Command and Return on the line, we're going to have the exponential of y.

Also, if we don't know how to use exponential and we don't know what that means, we just click where the function is and go in your keyboard with Fn, which is function, and F1. And you see that it points out to Help, right here. So you can see, OK, logarithms and exponentials. You can read the description of the function. And you can see, right here, how it is going to be used. So that hopefully is going to help you to face the different functions you can see throughout the code.

Also, what happens if we want to remove a variable? Well, you just write rm, which stands for remove and, in between the parentheses, what we want to remove, right here. We just click on it. And we see that the x, boom, it's gone.

OK. So now that we've gotten through how to assign numbers to variables, we're going to start asking logic questions to R. So first of all, I need, again, to create some value for x, now that we have it. I'm going to ask R if x is bigger than y. So for that, let's just run the command. And it's obviously false. So in this case, x is not bigger than y. So it has to be smaller than y. So there's the other way around of the question, right here.

We can also ask if two values are equal by using two equal signs one next to the other. Or we can ask if it's not equal. So by applying this sign next to an equal, we are creating the negation of the equal, so not equal. And here is how we ask logical questions to R.

So let's move on to the data types. We have, obviously, numeric data types, which means that there are numbers assigned to those variables. And we can ask what's the class of this value to R by writing class and the variable that we want to ask about. So in this case, it's saying that it's numeric because, if we go to this value here, we see that that's a number.

And we can ask also if that's an integer, which it would be a number without decimals. So we ask, is integer. And in this case, it says no because, as you can see, we have a lot of decimals following this number. And we can force one number to be an integer. So how do we do that?

It's like put this variable here as an integer. So if we click on it, it says that now it could be 7. And we could store it in a different variable. So in this case, let's just put this one. And as you can see, now it's the same that we had, but just in integer form.

We can also assign characters to variables. In this case, so we want to write something here, let's just write lobster and assign it directly to z, which would be a very reasonable way to assign a word to a variable. But in this case, it doesn't work. You can see here in another color that's an error. So what is complaining is that, OK, R is expecting here this lobster to be already a variable that has been created.

How can we fix that problem? We just have to put lobster between commas, like that, and put the C, which means concatenating. And by using this format, R is going to understand that that's a word that we're going to assign to the z. So if you go to the different values, now z is a lobster, right here. And let's ask, what's the class of z? Well, it's a character.

OK. So let's move on to vectors. What's a vector? Well, so far, we've just assign one value to one variable. But we can just concatenate different values for the same variable. In this case, for example, we want to create the vector number 1. And we want to assign 1 and 2 to the vector number 1.

How do we do that? We've seen already this construction form, which is the C and parentheses. So in between the parentheses, we can separate by commas the different numbers we want to assign to V1. So if we enter this command, we see that now we have created this V1.

And it's a numeric character, if you follow these letters here. And it has 1 and 2 concepts inside. And these 1 and 2 concepts are the number 1 and the number 2, conveniently. But we can create vectors with three categories. And in this case V1, instead of having two, it has three, like 1, 2, 3. And the first one is 11, 22, and 33.

But we can also create vectors with character variables inside. So instead of having numbers, we just concatenate different words inside. And if we go to how vector 2 looks like, we see that there's a gathering of characters right here, that we have five that goes from 1 to 5. And inside of these, we have Monday, Tuesday, Wednesday, and so forth.

OK. What if we want to create a very long vector, but we don't want to spend time by writing each number on it? We have this very useful tool, which is the repetition of numbers. So we write this. And if we want to know more about the function, we can, again, go to Fn in our keyboards and F1.

So by clicking that, we get some help right here. This is because I have different packages installed. So that could be part of different packages. But we're interested in the package base, which is what comes from R directly, which is the pure part. So we just click in here. And we're going to be directed to this function here.

And we can read through. We can repeat integers, length. And x stands for the vector we want to repeat. And so by reading that, we can have a good understanding of what rep is doing. So let's just execute that order. And we see that we are repeating 22 two times. But if I would change that, we want to repeat 22 20 times. I can just do that. And we repeat 22 20 times and so forth.

OK. What if we want to create a concatenation of numbers that goes from 1 to 100? It's as easy as just saying the first number I want to be 1 and the last number I want to be 100 and just separated by these two dots in the middle. We're going to create the vector number 4 by executing that order. So we see that we have a vector created by different integers that go from 1 to 100. And they're just 1, 2, 3, 4, 5, and it goes to 100.

What if I want to see the complete vector? Because here, we don't have enough space to just get to see the whole numbers. Well, it's just select V4, and we Command and Return. We can see here the whole thing. And also, we can write on this box here. I want to see V4. And just click Return and you see the same thing. Now, you don't see the difference because here it is. This is the one that we first asked and here is the second we ask as well.

All right, I'll just keep going. What if we don't want a list that goes from 1 to 1001 one by one and we just wanted to do two by two, so we just get 1, 3, 5, and so forth? So there's also sequence here. We create we can create a sequence that goes from number 1 to 100, but by 2. So we just skip one and we just go two by two. So we have vector number 5. Let's see how vector number 5 looks like. So as you can see, we've got 1, 3, 5, and so forth until we get to 99 because, of course, 100 was out of the pairings.

OK. So what if I just want to get one of these numbers in the middle? So we have, for example, just for simplicity purposes, if we go to V4 again that we are going from 1 to 100, what if I want to see what's in the slot, let's say number 20? So if I want to make a consultation, just let's go to the vector number 4 and just put the [? cloud ?] [? ladder ?] here, concatenating.

Let's just say, for example, I said the number 20. So I want to see what's inside the number 20. So in the slot number 20, we have the number 20, in this case. So this is how I just select one number. Or if you want to select more than one, in this case, we go to V5. We use the concatenation form.

So I just want to explore what's in the slot number 10 and what's in the slot number 12. And I'm creating a vector inside of the vector number 5. So I'm just going to make this consultation. And you see the numbers that we have in these particular slots.

It wouldn't work if we would do it like this. I wrote this for the purpose that you can see that there's an error applied to that code. So that's because you should put these numbers inside of the construction, like with the C and the parentheses, so R is able to understand.

What if I want to get rid of one of the numbers that I had inside a vector? I can do that too by-- in this case, I'm just going to create the vector number 6. We're going to use vector number 2. Let's just remind what was vector number 2. It was that list of the different days of the week.

And we don't like Wednesday, for some reason. We just want to get rid of Wednesday. We use the slot number 3. We say minus 3. So V2, like vector number 2, minus the slot number 3, which, in this case, is Wednesday. So let's just apply that. Now, just check out what's in V6. You see, boom, Wednesday is gone.

What if we want to create WPA. Further details about functions-- oh, I've written here after the different functions what they are supposed to do, so you don't have to console every time the Help section. So I created a comment right after them. So these are kind of similar. These are different functions that are going to create random constructions of data.

So every time we would run each one, it's going to be different because that's the purpose of the randomization of the data. And in this case, we are going to go from 10 to 50 by 10. Let's just see how it goes. So we have a random selection of 10 numbers that are going from 10 to 50.

What happens if I rerun the same command? Hopefully, they are different. OK, so just by applying the same structure of data, we get different outcomes because the purpose of that is that we resample the data. We can also put this randomization into normal distributions of data. So that's why we have this rnorm here.

In this case, the sample is just random. But this is forcing this randomization of the data into a normal distribution. So I'm just creating this. I'm just going to go with a V9 that I like it more. Now, if we take a look at V9, we have a lot of numbers, different numbers.

But if we want to see how this data looks like, we just create a histogram by putting hist right here. So also, you can get familiar with how the plots look like by doing this. I'm just going to see how a histogram for this data looks like. So here, we have a normal distribution of data by applying this function. Here, I'll just do it one more time, so you can see that the distribution changes and so does the plot.

OK. So now that we've been through the construction of different vectors, it would be interesting to ask R about different summary of those vectors. So one of the things that we might be interested when we are facing vectors is get to know the length of those. So let's just ask R, what's the length of, in this case, vector number 4?

So we got it like the length is 100. And what happens if we just ask the summary? What is it going to summarize? So well, it summarizes this, a bunch of information. So we go from the minimum to the maximum. So the minimum value, in this case, is 1. The maximum value is 100.

But we also have the different quantiles, and the median, and the mean. So this is a very useful function that we can use to get a good summary of the data that we are facing. Also, we can use these different functions right here to sum a vector. So in this case-- I don't know-- let's just use the vector number 1. So it's going to sum each individual number that's contained inside vector number 1, in this case. We can also ask only the minimum of any vector, also with the maximum. And we can calculate the mean of any numerical vector as well.

OK. So now that we are familiar with vectors that we just went through a lot of information related to vectors, let's just move on to matrix. Matrix is, at the end, an accumulation of vectors. So we are moving into two-dimension data. So we have different rows and different columns. But they are only related to numerical variables. We don't get to have characters inside of matrix. So just a heads up to put attention into that detail.

In this case, we are going to be creating a matrix that's going to combine vector number 5. We're going to put it in rows. So we want 10 rows and we want five columns. We just click on that instruction. We see here that now we have a different subset of data.

So far, we've had just values and vectors, right here. But now, we just created a new section, which is called data. And here, we have a matrix. We can consult this matrix again by just clicking into Mat 1, just command and return. We see it here. We can also type here Mat 1 and Return. But we can, in this case, click on the data itself. And it's going to be displayed just like that, so we can visualize easily how we've constructed the data.

So what this function has been doing is to split the vector number-- in this case, number 5 that we had before-- just putting it in order, separated in different columns. Also, this needs to be-- it has to match with the length of the vector because, in this case, I would say let's just not structure it in 10 rows. Let's just make 11 rows.

What should happen is that R is saying, hey, this is not accurate because you're asking me to put the data in a format that it doesn't really fit. Just the difference between warning and errors-- so previously, we've seen an error. It's just R is saying, no, I'm not going to do that. But if there is a warning, it's saying that, hey, there's something wrong here, but I'm able to do it. So that's mostly the difference between warnings and errors. You can just close that tab.

And OK, we can also put names on these matrix. So we just consult with R the column names of those matrices. In this case, it's null because-- let's just consult again the matrix. We have nothing here. Let's just remember what was in vector number 2. So we have the different days of the week. So we're just going to assign these five characters as a title of our matrix, just like that. And now, we see that where we had nothing, now we have the different days, right here. So we just assigned column names to the matrix that we already had.

And how can we navigate through matrices? So first of all, we have to name the object we want to navigate on. So mat number 1. And this stands for row, like the first number, it stands for the row. We separate it by a comma. And the second number stands for the column. So in this case, I want to see what's on the row number 4 and the column number 3.

So it says that it's the number 51. And just for our information, it says that the column number 3 is named Wednesday. And what if I want to get the whole row? So I just put nothing in the second part. So I erase the 3. I keep the comma. It's very important to keep the comma. Otherwise, it would give an error. I'm just going to try.

Oh yeah, it's going to go to the number 4. Yeah, so it takes the matrix and it goes for the slot number 4-- 1, 2, 3, 4. But what we want here is the whole row. So we'll just put the comma here, execute the thing. And it's giving us the whole row. So we want the whole column is just keep the 3, erase the 4, we're going to get the whole column number 3, just like that.

And of course, if we want to reassign a value that we're getting-- because here, we are not recording this anywhere. We're just consulting. But if we wanted to record this data or assign this data to a new variable, we are always welcome to

So by just creating a name of the variable-- like xx, for example, just put the arrow right there. And we would assign this data that we're consulting into a new variable. So now, the xx variable here, it corresponds to the column number 3 of the matrix number 1.

OK. So in this case, what am I doing here? I'm creating a vector, which is in the second part of this structure. So I'm messing with the columns. I'm getting the column number 1 and the column number 3 of the matrix that we just created. And we are storing this data into this new variable. Let's just check it out, how it looks like. And indeed, we have the two columns right here.

Moving to data frames right now, data frames are an evolution of matrix. And the good thing about data frames is that they allow us to put also factors inside, allows us to put as characters inside. So it's not going to be only numbers. So we can have a mixture of different data structures.

Also, R by itself has a library of data frames. So if you ever want to play with data just for fun, you can access this data that's included inside the R. In this case, the one that I'm using is one that's called mtcars. I'm going to assign this name to cars right here.

So just click on it, you'll see that this new data has been created. It's in the same section as the matrix. But we see that, in this case, we have this little arrow, blue arrow right here that, if we click on it, it's going to display the different column names and information of each column that is assigned to that database.

So just for visualizing purposes, I'm just going to click on top of cars. And here is what we have. This database is related to different car engines, miles per gallon, the different number of cylinders that it has. So it allows us to play quite a lot with this data.

We're going to just ask the length about cars, as we did before with the vectors. It also applies to data frames. So here, we have the length of that. What's the number of columns that we have involved? In here, we can also ask that to R. We can also ask the number of rows that are involved. We can ask what's the column names. And again, we can use the same data structure that we use for matrices by consulting what's on the row number 4 inside of cars, so using the same data structure we used before.

And there's something else that's a new feature that I'm showing here is that you can also select or consult a column by using its column name. So in this case, there's this column named HP. I just put the dollar sign here. So I'm saying this is the cars object that we assigned, dollar sign. And inside of the cars, I'm consulting the column called HP.

If we don't know what other names are involved in this data frame, we just go from cars just right to dollar sign. And it's going to be displayed in a list what are the different columns that we have. We can just scroll down and click the one that we like, just intro, and it works, just like that.

OK. So now, we have already been going through a lot of different data structures. We went from individual characters to numbers. Then we moved to vectors. Then we moved to matrices, data frames. But there's the summary of everything. You could store every single type of data in one data structure, which is the masterpiece of everything. This is called the list.

So we can create a list of everything we want. We can put together, as you can see, different vectors. But we can put together a matrix there. And we put data frame. So in this case, we're putting everything in that list. And we just create this. And if you can see here, it has been created.

If you click on that arrow that we had also in the data frame, we see the different data structures that we have inside that matrix. The first one is a numeric vector. Then we have a character vector. We have another numerical vector, a vector that's created with integers. We have a data frame. And we also have a matrix, right there.

So this is a way to store data easily that you don't even have to think about what's the structure of that data. We just put it in there. And you can consult, OK, let's just go to the space number 6, which, in this case, is-- 1, 2, 3, 4, 5, 6-- is going to be the matrix. Let's just see. And of course, the matrix is here. And this is how we can store everything in the same place.

And now that you've been exposed to such amount of data, I just prepare some exercises for you to start emptying your brains and putting everything into practice. So I encourage you to take these little questions here and try to answer them. And if you are eager for more, I encourage you to go through different internet resources. One good thing of R is also that you can find everything on the internet. So there's going to be a lot of resources for you to practice as well, if you are hungry for more.