[SQUEAKING] [RUSTLING] [CLICKING]

**HELENA VAILLICROSA:** All right, hopefully you had the chance to practice the exercises that we went through on the last video. And now, we're going to start going directly how to import data in R because we've been working on how to type or create data in the script itself. But now, it's the turn to import data from other sources.

So I'll just erase everything, so we start with a white board. So if we click here, we're going to clean everything we've been working, also with the plots. And we can do the same with the console. OK? So now, we start with everything brand new.

So first of all, where is the data taken from? If you type getwd, which it means working directory, it's going to say what's your working directory that's been linked to. So in this case, I'm going to Users, Helena, and Desktop. And you can change this working directory by creating setwd, and here write a new route. I'm just going to go with the same one right here.

And by default, R allows you to import CSV files. This is the command you should write to import comments as a CSV file, just read.csv, and then your working directory here. This works for my laptop. But of course, you would have to change it depending on the location of the file you want to import and always between these commas right there. So if I click on here, you see this data has been imported. And if I click on it, I can visualize everything that's inside. You can try it with whatever CSV file you have.

But what happens if I want to import files that are in, for example, Excel format? Well, for that, we would need a package. In the beginning, of the sessions I briefly talked about packages. When you go to this tab, you can see what packages you have installed. So now, it's time to go deeper with packages.

Packages mostly are an accumulation of different functions that somebody has uploaded into the R interface. And you can download them using Tools, Install Packages. And here, you type the name of the package you want to install. Let's just go with the one that we want to install right now. Phase 1, Install, Yes.

OK. Now, it's already installed in our system. Also, another way to do it would be to type install packages and, between commas, the package name. So that works either way. And now, we have to charge this library into our session.

So OK, the package is installed. It's in our computer. But now, we have to call all these functions into our R session. And to do, so we just type library and the package we want to charge. So I'll just do this. And now, if we go to the packages list, we should see that open XLSX scroll all the way. You see? It clicked. It means that it is loaded. You could click here and click here and it would work exactly the same.

We've gotten good at installing and charging packages. Let's just use them. I'm going to charge this file that is an XLSX file. So it's an Excel file. It mimics the function of the read.csv. So it's read.xlsx and the same functioning. So you just put the route here between commas and it loads itself. Here, the name I'm going to put to this database is T1.

It says that the file doesn't exist. So that could be because some of these parts of the route is not well-written. Also, there is something important that might be handy for you is that R has its own R format file, which is the .r. So it's a very compact way to store data. And you can also load R files by just typing load and the route you have right here. So I just charge these with R files. And I have the data frame attached.

OK, I just realized that the font might be a little bit too small. So I'm just going to make it bigger for you. And also, I'm going to show you how to make the font bigger for your own purposes. So if you click on your command tab here and then you go to the sign +, you see that? Things are getting bigger. And if you're going to go to the other way around, you just Command and -.

OK, so that could be handy for you. And also, I'm going to show-- I'm going to talk about a very interesting tool that R offers, which is the Find and Replace tool. So if we go here, we have this magnifying glass. You click on it and you can type any characters you want to look for. Let's say-- I don't know-- I want to know where I typed read.

So it's going to guide you through the different places in your script that you've wrote read. And also, you would be able to replace all read things into something else. I'm not going to do it now because that would screw my whole R session. But that is a very useful thing to do whenever you have, let's say, a variable that you want to change. So initially, you were working with temperature. And you want to change temperature for precipitation. And you can do it all at once in a very simple way.

All right, that being said, also, I have this organization right here in the right. So you can follow the structure of the script that I'm having here. And I can click on the different names of where I was. So I was about to rejoin the explanation right here. So this makes it easy to find as well. How do I make things appear here? I just have to put four of these signs here and for at the end. So let's just try it. Now, hi has appeared right here. So this is a way that I personally find easy in order to have tidy scripts and not get lost into all this code.

OK, let's get back to the exercises. Now that we have learned how to upload data from our computer, I'm going to work with this database right here, which is the above that I already mentioned. OK. First of all, what I would like to see is how the data looks like. We can use this summary function that we already talked to previously.

This is going to provide me, on a glance, what's inside this database. So where's the beginning? Right here. So in each column, we're going to have each column name. So that's the column name. And then depending on if the variable is categorical or it's numerical, we're going to have different features applied there.

For example, season, it's a character. But the ones that are numeric-- let's say carbon-- we get the minimum, the different quantiles, the mean, and the maximum value, and the number of N/A-- so here, I have 500 N/As-- and so forth. That's going to be applied to each column, so each variable.

Also, another way that we can use to see summary of our database is the one that's provided by this package. So just to remind you how to install and charge packages, you go right here-- Tools, Install Packages. And here, you type the package you want to install and install.

And then to call the package in your session, you can do two things-- this library situation and in. But if you don't want to charge forever this package in your session and you just want to use it in this line code, you can use this. So you put the name of the package here, like these two signs, and then the function you want to use. R is going to understand that you instantly call that package, but then you don't want it anymore.

So I'm just going to see what this provides. This is a very cool way to approach the data because it offers in another color wherever we have negative values. And also, it provides us of the same information, like the mean, standard deviation, different quantiles, and also the missing values.

And this very interesting feature, which is a histogram of each column-- so we can see how the distribution of the data looks like, which could be sometimes very handy. Other ways to approach the database would be to call the headers, which is the different rows of the data we have. I'm just going to replace the cars for the above. Here, we have it. So we're going to see only the six first rows.

Just to remind you, how can we check the column names? Just call names. This is all the variables that we have in this database. This is something that is also very useful, which is unique. So it's going to say, for example, we have a character associated to this database.

Let's say-- I don't know-- topography. What are the topographies that we have? So we have bottom, slope, and top. These are the different categories that we have inside this variable. This is something that we already talked about, which is a quick reminder of consulting column by column.

So let's go to above in column number 1. Here, we have the column number 1. But we also can call the things by using the dollar sign. This is the number 1. It's going to be the same. And we can individually provide a histogram of the variables.

And let's say I'm just going to see how carbon looks like-- histogram, the database, dollar sign, and the column. And this is how our data looks like. Obviously, you can also change some of the features of this graph. You could even change the different x or y-axis. Or you could split the data in more bars. But that's something we're going to cover in future videos.

So following this histogram that we just created about carbon, we're just going to make a subset of the database we have based on the amount of carbon. So let's say we want to keep the ones that have low carbon, just subset, just change the name of the database, right here.

And now, inside the database, the variable we want to focus on. And we want it to be lower than 50. Just go ahead. And you see that now we've created another database right here where the columns are the same, but the only difference is that we have less rows, less observations.

And if we see how the histogram looks like, we see that now it has changed. And the maximum is 50. And we have all the rows that have a value of carbon below 50.

OK. So I've already mentioned in other videos that R itself has a library of different databases. I'm just going to go through that and just write data in parentheses. So we're going to see all the data sets that R includes without installing anything on the side.

And I'm going to pick a database called NPK, since my field of expertise is field sciences applied to plants and climate change. This is a database. I'm just going to charge it, so you can have an idea how it looks like. It has different blocks of experiment right here. And also, it indicates in every row if that block has been fertilized with nitrogen, with phosphorus, or potassium. In this case, 0 means that it hasn't and a 1 means that it has been fertilized. And it records the amount of yield that has been produced right after this fertilization.

So we're going to play a little with this database. First of all, I'm going to create an extra column to combine nitrogen, phosphorus, and potassium fertilization at the same time. So how am I going to do this? Well, I'm just going to create directly a column in our database.

Now, just to remind you again, it only has these rows here. But I'm going to create an extra one, right there. And it's going to be called All Ferti, All Fertilization. So I want to paste the column that includes the nitrogen information, the phosphorus information, and the potassium information. And I'm going to separate each one with this symbol here.

So I'm just going the go ahead, run the command. And now, we've created a new column here that includes the different fertilizations. And we can see only in one factor what has been the different treatments that yield has been passed through.

OK, I'm just going to come with a little exercise here. I'm asking you if you could subset this database taking only the data that has been fertilized with nitrogen.

So how are you going to answer this exercise? First of all, let's just go back to the data and see, OK, nitrogen, there's 0's and there's 1's. And we want the ones that have just the number 1, which means that they have been fertilized. So we just make up a name. Subset, what's the database we want to subset? What's the variable? And we want the ones that are equal to 1. And here is how we create a subset filtering only by the ones that have been nitrogen fertilized.

All right, other things that can be useful-- we have the rbind and the cbind, which basically paste vectors into the database. So let's say we have the different columns right here and we want to add an extra one or if we have the different rows and we want to add a row on the bottom. They have to match by length. But other than that, R is going to accept. Let's just put that into practice.

Previously, we created this new column, right here, just to remind you, that one. OK? But what if we just create the column on the side? OK. Here, you see that we've created a vector that includes the same information that we have previously included in the database. And we want to just use the cbind just to combine these two.

So that's the new name that I came with. Apply the function. I want to merge this one here and this one here. So if they match in size, R is going to allow us to do it. And now, I have-- this is the one that I first created. And this is the one that I just added. So this is two different ways of doing the same thing. R has multiple ways of doing sometimes the same. So just pick the one that works for you the best.