

[SQUEAKING] [RUSTLING] [CLICKING]

HELENA

In this part of the code, I'm going to guide you through the gap-filling process based on maps. So this is a process

VAILLICROSA:

that I use quite a lot in my research, which is when I'm facing a new database sometimes comes with some empty slots, some N/As in the data. And I want to fill them based on field data.

Obviously, data coming from maps is not as accurate as data coming from field experiments. But this is a good proxy we have to maximize the data we have and use them all in our models. So what I'm going to be doing here is to use the latitude and longitude of the coordinates that comes into the databases, transform them into spatial-explicit data points, and use these data points to extract the data coming from a map.

So let's just go ahead. I'm going to download real data that I created that is in my GitHub account. You can download it freely here. It's a database that calculates nitrogen uptake global scale based on field data. But I encountered that some of the drivers-- temperature, precipitation, soil information-- has not always been collected, or maybe wasn't reported in the original papers. So that's why there are some data gaps. So well, you can see that it's a big database. It includes a lot of columns. Some of them are complete. Some of them have N/As.

So I'm just going to see how are they called. These are all the names of the columns. And we are going to be particularly interested in the latitude and longitude, which is the third and fourth column right here. I'm going to make sure that we don't have N/As there.

So I'm just going to subset the data that doesn't include N/As. So it is N/A. So it doesn't include N/A in the latitude. And it doesn't include N/As the longitude. This is one of the several ways we have to make sure that we exclude N/As in these particular columns.

OK. Now that we have this sorted out, there are two ways of transforming these data points, so this latitude and longitude in data-explicit points. One is using `SpatialPoints` here. And the other one is using `SpatialPointsDataFrame`. In both of them, I need to specify where latitude and longitude are located, what's the database that I'm using. And in the first case, I need to specify also the resolution I want the data to be.

I personally like this more. But with this one right here, it's going to also work for our purposes right now. We see that the outcome we get is a slightly different one in `SpatialPointsDataFrame` and the other one is `SpatialPoints`. But if we plot them they look very similar.

And we can plot them with the shape of the Earth. We don't have a good understanding of what is this plotting. Here, I added this map that we already used in other parts of the videos. So here, we can have a good understanding of where this data is coming from. And it also certifies that we have built our map appropriately. Sometimes, if you mismatch latitude and longitude, you're going to have a twisted map. So let's just try it for fun.

Let's see how it looks like. You see, now, it's totally changed 90 degrees. But just a heads up to put latitude and longitude in the appropriate location.

OK. So going back to the process, I'm going to use as a base this map that we've been using recurrently, which is this temperature map. And I want to extract the information using these points. So I only need to use the function `extract`, the map where I want to take the information, from and the points I want to use to extract the data.

So I use these. And I see that the result is this right here-- oh, sorry-- here. And we see that we have a new value here, a new vector that is 159 data points long, which is exactly the same number of columns-- of rows, sorry, we have in our data frame.

And if instead of having one map here we would have had a stack or a brick, we would have had a column per map. So in this case, we only have one because we have one map. But the more maps we have, the more columns it would add in our result.

Now, I'm going to guide you through an exercise that hopefully is going to be a good summary of all the functions and all the concepts we've been working with in this GIS session. I'm going to guide you through it. So initially, there's nothing done here. So I'm going to type while doing it live.

And first of all, I would ask you to download these 10-meter resolution average temperature maps from WorldClim. And this is the link. But in case you don't have it handy, this is how the web page looks like. I would ask you to go to this average temperature and download this one.

Feel free to download other resolutions as well. But I just went to the course one to make the explanation more fluid and the calculations easier to calculate. So let's download these and store it somewhere in your computer that you know the out of. And also, I'm going to ask you to load all these maps in your R session and calculate the average of all the maps, so to obtain the yearly average temperature.

OK. To do that, I'm going to use code that we've already used. I'm just going to jump around. I think I have this code here. Yes. I'm going to use this.

So I'm going to charge everything at once, instead of going one by one. I'm going to do it faster. This and this-- I don't really want this code here. I'm going to create a stack of these. And now, we want to calculate the mean of all these maps. So to do that, it's going to be called-- I'm going to use this object here and calculate the mean. This is what we got.

So see that the information looks good. The values-- minimum, the maximum-- they look feasible. So it seems that everything worked out pretty well. I'm just going to plot it to see how it looks like. So yeah, it seems reasonable. So I think we've successfully completed the first step.

Second step-- I want you to download the nitrogen uptake data from this page, which is my GitHub page. This is the database we've used in the previous exercise. So you might have it already. And I'm going to ask you to create a new dataframe by subsetting the columns numbered 3, 4, and 18, which are the latitude, longitude, and mean annual temperature, which is what we just calculated in our maps.

So to do that, I'm just going to load the data that I should have here. This is the data. We're going to subset. I said the column number 3, number 4, and number 18.

Now, we have this object that has exactly the same length-- so the same rows, but only these three variables that I asked you for-- latitude, longitude, and mean annual temperature. I chose these variables because we have some N/As, as I mentioned. You see some empty values that we would like to fill in further steps.

OK. Let's move on to the step 3. I want to create spatial points by using the coordinates in the subsetted data frame. And I also want to extract the mean annual temperature of the data points you just created from the map and include them in a new vector to the data frame, resulting in a four-column data frame.

So I'm going to go through the code that we just used. Oops, sorry, sorry, sorry.

Find them here. This one. We're going to use this one that's shorter.

OK. We created that. And now, we want to extract this information. So it's going to be called-- MAT stands for mean annual temperature. And I want to extract from this variable that we created-- that's the mean temperature-- and the PNT.

OK. We created this vector here. And now, we want to add this vector to our initial data frame. So our data frame was created-- was called this. It's going to be called Coming From WorldClim.

This is one way to do it. There are other ways. So if you find an easier way that works better for you to do it, feel free. So now, we have this initial database that had only three columns. Now, it has four columns. And we just added this new column here with a new variable.

We could even see how these two match. So this comes from ground measurements. So it's based on field data. And this is the map. We could even see how these two match. We could, for example, plot these two informations or create a linear model. I'm just going to go with the model. It's going to be called Model 1. It's going to be a linear model, where I have these, based on WorldClim. This data is going to be this one.

Let's see. Well, it's almost the same data. So they agree in a 94%. Yes, almost 1, a perfect regression. So this is a good proxy. We could also plot them in a linear model-- in a regression plot. It's just like this. And we see how the data mostly agrees.

OK. And now finally, I want to combine these two columns. So create a new column in your data frame where you gap-fill the mean annual temperature variable with the data you just extracted. So what I want to do here is to use-- so mainly use this field experimental data, but use the map to gap-fill these empty spaces that I have here.

To do so, I'm going to use code that I already have, which uses a for loop, so it's faster. OK. What I'm doing is to-- ah, this data-- I'm going to change the code using this tool. So what initially was General DB, now it's going to be called N in the section, which is what I selected. And I replace them all. Yes, four occurrences.

OK. And now, the data-- so what I'm going to be doing is I'm asking R to generate this for loop that goes from 1 to 159, which is the number of rows that I have. And I'm saying, if this is N/A-- so it's an empty space-- if this variable is an empty space, go to the same variable and fill this space with what you will find here. And if that's not an N/A, just leave it like is.

You could even erase this part. So we could not need this else part because it's basically leaving the thing as they are. So you could decide to erase this. And things would work either way. So I'm going to make this run. Maybe it gives us some error. It seemed that it went smooth.

OK. Now, we see that this variable here doesn't have N/As because we filled these information with the information coming here. In this case, I have destroyed the initial variable, meaning that now we no longer have the variable with these spaces. I could have created a new variable, which is what I'm going to do now just for you to see it and to compare things better.

So I'm going to rerun the whole code. A very easy way-- these are the advantages of R, that you can rerun things pretty easy. And now, instead of rearranging things on top of the variable that already existed, I'm going to create a new one that's going to be called New.

New-- I'm going to redo the process. Now, we have this new code, this new column here. And so when we have an N/A, it's always the same as MAT. But in the ones that we have N/A, we've used this one instead. And this is a very efficient way that we have to gap-fill data coming from map data.