6. Performing Analysis - Transcript

Introduction [00:00]

Hello, and welcome to the sixth and final video about the RECAP Preterm data platform. In this video, I will be showing you how data analysis works on the platform.

And again, I advise you to watch our previous videos before this one, if you have not. And I will again be following our wiki here. And you can find a link to it in the description of this video.

Analysis Using Individual Patient Data (IPD) [00:23]

So in this last video, we're going to go over this last part here: “Performing Analysis” on the platform. And there's basically two ways of doing this. And they depend on what type of access you have to the data that you're using on your analysis.

So the first one assumes that you have access to the individual patient data. So someone has granted you access to the individual level data on their node. And if you have access to the actual data, you can just go to the node and export that data into your computer and then just use it as you normally would. So let's see how we could do that, I'm going again to our test node that we're using, I'm going to the Data Repository. And I'm just going to show you how you could export data here. So we'll select our previous project and go to the tables. You can see our tables that we've created here. And so I'm an administrator here, so I can see all the tables. But we would, like we did on the previous video, we granted, we created a user and granted that user access to this table.

And if the level of access was access to individual-level values on this table, or this view, rather. So what that user would be able to do is just select the view here, and then just click Export. And then if you click Export, you'll see you'll have multiple formats, you can select CSV, SPSS, SAS, Stata, CSV, Excel. So you can choose whatever format you want, and then just download the data into your machine. And as long as you have the data, you could do this on all of the nodes that have granted you access to those data, you can download all the data into your machine and combine the data and just run the analysis as you normally would. So that's not very interesting, from the point of view of the platform, because you're just using it to download the data, and then you just run the analysis as you normally would.

But the thing is that, as I'm sure you know, there's multiple issues with being granted access to individual-level data. So that would presumably require some sort of agreement. Usually, it's a Data Transfer Agreement between institutions. And if you're including multiple cohorts in your study, you'll have to do DTAs (Data Transfer Agreements) with all of those institutions. That could take months or more. So in order to try and circumvent those legal and ethical restrictions, there's a second way of doing analysis on the platform, let me just go back here.

Analysis Using Non-disclosive, Aggregated Data (with DataSHIELD) [04:01]

So what we just saw is just... individual-level access to the data, if you have that, you can just do regular analysis, download the data, do your analysis, that's the first way of doing it. The second way... let's say you don't have access to the individual-level data, or even you're not interested in having access to the individual-level data, because you know, the legal and ethical restrictions around that, and all the bureaucracy you'd have to go through in order to get access to those data. So you could try instead, the second option, which is doing analysis using non-disclosive, aggregated data. So you can do this on the platform using something called DataSHIELD. And so let's see what DataSHIELD is. I have the website open here, you have this link on the description of the video too.

And so, what is DataSHIELD? You can see here, DataSHIELD is an infrastructure and series of R packages that enables the remote and non-disclosive analysis of sensitive research data. So several things here. So the first thing is... so, R packages. If you don't know what R is, R is a programming language mostly used for statistical analysis, and DataSHIELD is built using the R language. So DataSHIELD is a series of R packages. So that's the way it works in R, you have multiple packages that you can install and then use different packages in your analysis. So the way it works is you don't have access to the individual-level data. But you can be granted access to aggregated data like we saw, just like we saw on the previous video. I can go back here just to remind you, remember, we went into this view, the harmonised view and then we went here to permissions and this is the user we had created in the Authentication Server and then we granted that user this level of permission via dictionary, and value is actually this is, okay. So having this permission that the user will be able to see the values. But in order to use DataSHIELD, you don't need to see the values. That's the whole point. So let's edit that, and switch to the lowest level of permission: “View dictionaries and summaries”. This is all that is needed in order to use DataSHIELD. Let's just save that.

Okay, so this is the account that we will use to test the DataSHIELD access. So basically, let's say you have requested this level of access in multiple nodes. We're just using one right now, we're using one of our test nodes. But let's say you've got a more real setting and other cohorts that have other nodes, you've asked them to harmonise the data. So they did, just as we did on the previous videos, they harmonise the data, they have something similar to this on their notes. And so you ask all of them to, after they've harmonised the data, you ask them to grant you access to their harmonised data, but again, not the actual data, you just ask for this level of access. And for them to grant this level of access that doesn't require data transfer agreements, because data is not transferred here. Okay, we'll see how that works in a minute. But basically, this is all you need. You don't need access to the individual-level data. And what will happen when using DataSHIELD, is you'll be able to connect to all of those nodes that have granted you just the lowest level of access. And you'll be able to run some statistical analysis using their data, but not having actual access to the data.

So the way it works is, DataSHIELD will... you'll have to install the DataSHIELD client, and I'll show you how that works. And you can then run functions, DataSHIELD functions, that will ask the nodes to compute some sort of analysis, and then what they will do is run those locally, in each of the nodes, and then each of the nodes will return just aggregated data that resulted from the execution of those functions. And then you receive all of that aggregated data and then you combine it on your machine, on your computer. So effectively, what you've done, is you've run a statistical function on multiple distributed data. But you didn't have actual access, you didn't need to have access to the actual data. But I'll run a bit of code in a few minutes just to show you how it works. But basically, the first thing we will need before we can install DataSHIELD is usually... well there's more than one, but really the standard for running R code is RStudio. I have it open here as well. There's also a link to this page on the description of the video. So you need to go here and then you can just click here to download RStudio Desktop and then install it. Okay, I have already installed it. I have it open right here. So this is RStudio.

And I have a script here that I've prepared as an example to show you, but a few points to make first. So the first one, is the idea for this kind of analysis is that it is a distributed analysis, right? So the analysis is computed in every node separately, and then they just return the aggregated results. But so far, we've only used a single node on our previous videos. So what I've done, is I replicated everything we've done so far on that test node, I've replicated that on another one of our test nodes, just so we can run some analysis using two different nodes. So you can see that it actually works.

When you're using different nodes or different servers, they're in different places. So, you'll be able to see how that works. And also the data that I used, when I replicated all the things that we did so far on the other node I used, it's basically, it's the same data, but because remember, the original data that we were using was pseudo data. So I just regenerated another set of pseudo data for the other node. So I'm going to show you some analysis here, some DataSHIELD analysis, and it's very basic analysis. But just keep in mind that this is all pseudo data. So don't pay attention to the actual results, because they might not make sense, because it's not based on real data. So this is just for me to show you how it works. Okay, so if the results don't really make sense, please pay no attention to that.

Okay, so I have a basic DataSHIELD script here, and I will explain all of these code blocks, somewhat. So this is a very basic script, just to show you the basic functionalities, and how it works. But in one of the videos of the last module of the summer school, which is the ECR module, there's a video where Andrei Morgan will be executing R code to run a DataSHIELD analysis that's a bit more complex, and it uses actual real data. So it's probably more interesting to watch that. So but this is just a beginner's, let's say, a beginner's script, if you will, so I'll put a link in the description to that video if you're interested after having watched this one.

So the first thing that we'll need to do is install the DataSHIELD client. That's what we're going to do, that's the package that we're going to use to communicate with the nodes. So this is the name of the package: "dsBaseClient". And this is the way you install packages in R. We also have to install this package, because it's a dependency of this one. So we have to install that one first. So let's run these two lines. And by the way, up here is the actual script that I wrote. And then down here, once I execute those lines of code, the output of those lines of code will show up, down here. Okay. So let's run those lines and install the packages. It might take a minute or two. So I'll probably just fast forward a couple of minutes. Okay, so that's a minute, minute and a half, probably. So the packages are installed. So now that we have the packages, we have to actually load those packages. So I will run these two lines here.

Right, packages are loaded. And now we can actually start running our code. So before we can run analysis on the nodes that we're going to use, we have to set up an object here, I've called it "login_builder". Basically, this object will hold the login information for each of those nodes. So let's create the object first. This is all in the documentation, by the way, on the DataSHIELD documentation, so I'll just run that. Okay, so we have our object, it's still empty. And now we're going to populate that object with login information for our two nodes. So our first node is the one that we've

been using throughout all of these videos. And we have to use this function here, the “append” function. So we're taking the “login_builder” object that we've created. And then we're adding, we're just adding information about one of the nodes that we're using. And we have to specify a name for the server. This can be whatever we want, but I've called it “EPICE-PT” because that's the data that we're interested in getting from this node, and then the URL of the node, which is this one we've been using. And then a username and the password. I'm using this account that we've created in the previous video, and then also, we have to specify the table that we want to access and the table... we actually have to specify the full path to the table. So that includes the name of the project, where the table is. So the name of the project, if you remember, was “EPICE-PT”. And then the name of the table was “EPICE-PT_Perinatal harmonised”. So it should be the “name of the project”, dot, “name of the table”. We can go back here just to make sure that we've got it right. So this is the project EPICE-PT. So we've got that right,and then we want this table, right? So “EPICE-PT”, underscore, “Perinatal”, then a space, then “harmonised”. And I think that is exactly what we have. So this is our table that we want from this node. And then this is just a default parameter. I think, even I, we can omit that. So this is the information for the DataSHIELD client package, for the package to know how to log in this node and retrieve information from this table.

So now, we were doing the exact same thing. But on a different node. Remember, I said a minute ago that I replicated everything we did on this node, I replicated that on another node, just with slightly different data. So basically, I've just called it “EPICE-PT2”. And then I'm using another one of our test nodes. So this is a different node. Okay, you can see from the URL, these are different servers, and then I also, on this node, I also created the same account, and granted that account access to this table, which is the same, it's the same harmonised view that we created here, it will just have different data, because I changed the original data a bit in this node. So these are our two nodes that we're going to use. And remember, we are doing... The idea is to execute functions in a parallel fashion, right? So, you execute a command here, as the client, and then that command goes to each of the nodes, the same command, and each of the nodes computes that command and then returns the aggregated result, never returns data. So in order for that to work, in order for you to be able to then take the results, the aggregated results from both nodes and then assemble that, combine those results, the tables have to be the same, the same in the sense that they have to have the same structure. That's why we harmonised them, right? it's just for the tables to have the same variables in order for this to work. Okay, so let's run these two blocks here. So we're adding the

login information to our login object. Okay, that's done. Now I'm going to scroll down a bit here.

And this is what actually builds an object with the login data. So all the information that we just put in that R object will just be assembled into this new object called “logindata”. Let's run that. And now we have... so this is the object that currently holds the login data for our two nodes. So now we only have to login. So let's do that. And that we do that by using this function here, “datashield.login”. And then we need to pass that object as a parameter here. And there's a few extra parameters, this symbol “D”, what this means is that once we log in... so basically the nodes, both nodes will create an R session for us on the nodes, that R session will have an object called “D”. And “D” will be an object that holds the different tables. So in this node, there will be a “D” object that holds the data from this table and then on this node, there will also be a “D” object that holds data from this table. So, we can refer to both tables by the same name, which is “D”. But there are different tables and there are different nodes. Okay, so, let's log in then, let's run this line. So you can see down here it's logging into the servers, and... there was an error here. So it's saying it's forbidden... Yes, it is forbidden because I forgot to do something.

Well, what I forgot to do is... so basically, I'm using this account, right? To log into the node. And that's fine. I can log in and I have access to the table. Because we've added that permission here. So that account has access to this table, but there's an extra permission, that I forgot, to grant this account. And that is permission to actually use the DataSHIELD. And the way we do that, is by going to administration here. And then we go here to DataSHIELD, click on that. And then if we scroll all the way down, there's a section here “Permissions”. And so basically, I have to say that I allow that user to use DataSHIELD on this node, it's not enough to give the permission on the specific table, you also have to, for the user to be able to use DataSHIELD, you also have to specifically say that I allow that user to use DataSHIELD, so we “Add user permission” here. Then we just add the name of the user and we can select “Use”. So we're allowing this user to use DataSHIELD on this node. So I'll click save on that. Alright, there it is, the permission to use DataSHIELD for that user.

So let's go back now to our script. And let's run that command again. So this permission that we just set, was on this node. It's the node that we've been using on previous videos. And then on the second node that I'm just using for this demonstration. I've already done that. Okay? I just forgot to do it on that first one. So let's second node should be okay, in terms of permissions, so, let's try running this again. Try logging in again. And it's done. Okay, so this time it worked. You can see here, “Logged in all servers”, “No variables

have been specified". This is because you can specify here, on this login function, you can specify, if you want, if you're not interested in all of the variables from the harmonised views, if you just want a couple of variables, then you can specify that here, you can set a list of variables that you're interested in. If you don't specify those variables, it just assumes that you want all of the variables. So it will assign all of those variables to that "D" object that we talked about. So "Assigning table data" and then "Assigned all tables", okay.

So at this point, there are two sessions, R sessions, one in each of these nodes. And those R sessions both have an R object called "D" that holds the data from the respective harmonised tables. So what we can do now is just start executing DataSHIELD functions. So for example, this is a basic function. This is the "dim" function. So it's just to check the dimensions of the data. So what we're saying here is that we want to know the dimension of the dataset that is kept in this "D" object. So let's try executing that and see what we get down here. And just clean the terminal first. Now let's execute that. Okay, so as you can see here, we get the dimensions of the table on this node, which is 50 rows, 10 variables. And then the same thing for the second node. It's also 50 rows and 10 variables. And then you get the combined dimensions, so would be 100 rows, and still 10 variables. So that's the "dim" function. And then also, you can just check the names of the variables that are currently on this "D" object. So if we just run that, this is the "colnames" function. You can see on this node, these are the names. So these are the names of the variables that we harmonised. We didn't harmonise all of them, we just harmonised two, I think. And then on the second node, it's the same thing, right? It makes sense because both nodes have harmonised to the same harmonisation dictionary, so they will both have the same variables.

Then there's something we can do here, for example, we can specify one of the columns of the variables in this "D" object by using the dollar sign, let's say we want to check this variable, we can run this "class" function, it will just tell us the type of that variable. So that's a categorical variable. So in R that's called a factor. So in this node, it's a factor and in that node, it's also a factor, which makes sense. They're the same variable. We can also use this "table" function, let's use the same variable, sex variable. Let's run that. And you can see we get... let's see what we get here. So we get, for example, percentages. So down here, it's just counts, right? Just frequencies. So in EPICE-PT there's 25 zeros, 22 ones. Same thing for EPICE-PT2, different numbers. This is the same thing, but just with the percentages. And then down here, we should have the combined totals. There you go. So if you combine those two, the data from the two nodes,

you get 46 zeros, and 47 ones, and then 7 missing values. So that's the “table” function.

And then let's say we want the other variable... you'll notice that I'm only using the two variables that we actually did harmonise in the previous videos, because those are the only two variables that actually have the data harmonised. So those are the ones that we must use for this. So I'm just using, for example, the “age_discharge” variable. Now I'm just calculating the mean of that variable. So let's run that. Alright, so we can see on the first node, it's 0.17. And then on the second one, it's 0.16. And then we also be well, we also have missing values and invalid values and the totals. And we know that it's 50 on each node. So for this particular function, you get the separate means in each node. And if you want to combine the mean, you can use the same function but use an extra parameter called “type”, and then just just specify “combined”. So if you run this same function, but now we'll get combined results, the combined mean. You can see here, “studiesCombined”, and this is the average age at discharge across the two nodes. So this is the global mean across the two nodes. You can see... so, I hope you're starting to see how this works. So basically, we've asked each node to calculate the mean for that variable. They return just the mean. And then we just combined the means. And then this is the combined mean. So this is how DataSHIELD works. Basically, this is a good function to illustrate how DataSHIELD works, it's easy enough to understand.

And then you can also do some graphs, let's do a histogram. Again, let's use the “age_discharge” variable, and we should get two separate histograms. So it just splits the data into different ranges, you can also specify the ranges if you'd like, this is just a default behaviour. So you get a histogram for the first node, and then the same histogram for the second node. And you can also, this function also accepts the “type” parameter to combine if you want to combine. So let's combine the histograms. And now we should get a single histogram, there you go. That has the frequency for both datasets combined. And all of this, we get all of this information and we don't have access to the individual-level data, we're just using DataSHIELD. So we're just using aggregated data. So you can get a sense of what the possibilities are here.

And then lastly, let's just try to do regression, simple linear regression. And I'm just using these two variables as the dependent and independent variables. Again, nevermind if it makes sense. I'm just showing you how it works. Okay, so I'm doing a binomial regression. Let's run that line. So now, it will just make multiple iterations. But let's go back, there's a lot of output from this function. But let's go back to the beginning. Okay, so this is where we started. And then basically, DataSHIELD is just going to the nodes

multiple times. And it's doing multiple iterations until it eventually converges, we can see “Convergence criteria TRUE”, so it has converged. And now it starts outputting the results of the regression, you have the usual values that you would get in a regression, and down here, you can see the coefficients of other regression. So basically, you can see that you are able to get... if you were to... if you had access to the individual-level data of these two nodes, and then you just downloaded that data and merged the two datasets, and just ran linear regression on this variable, you will get the same results, okay? And basically, we were able to do that linear regression, and we don't have access to the individual-level data.

So I hope that illustrates some of the possibilities of analysis here, of using DataSHIELD. Okay, so just a bit of housekeeping, just after you after you're done, you should just log out. Because remember that there currently are two sessions in each of the nodes. So if we're not going to use them anymore, we should just log out in order to just clean up after ourselves. So let's just log outs on both nodes. We’re logged out and we're all done.

Closing Remarks [35:30]

So, this was a very basic example of DataSHIELD. This is a really very simple example. There's many more things you can do with DataSHIELD. And you can see that if we actually are able to still do analysis, and yet not have access to the individual-level data, you can see how much easier it makes our lives to avoid bureaucracies of data transfer agreements and those sorts of things. So again, If you're interested in this, there's a link in the description to that ECR video I mentioned. There'll be some more complex analysis there. So if you're interested, you can watch that video as well.

And so that is it for me and for this module of the RECAP Summer School, I hope you enjoyed it. And I hope you've learned something, and I hope you enjoy the rest of the Summer School.