4. Harmonising Data Across Multiple Studies – Transcript

Introduction [00:00]

Welcome to the fourth video about the RECAP Preterm platform. In this video, I will be showing you how to harmonise data using the platform. And again, I will be mentioning some things that we did on previous vídeos, so please watch those first — if you haven't — before this one.

And as usual, if you'd like to follow along, you can find a link to this wiki that I'm using, on the description of this video.

So this time, we're going to be covering this third part of the tutorial, which is *Harmonising Data Across Multiple Studies*. Well, we're going to see how to harmonise data in one study and I'll explain then how it would work if multiple studies were involved.

So let's click let's click here, on Part C.

Using Views in the Data Repository [00:58]

So, harmonisation on the platform is done in the Data Repository. So remember, that was one of the layers — one of the components of a node — where we imported data to on the second video. So that's where the data is, so it makes sense that that's where the harmonisation takes place.

So how does it happen, then? Well, the starting point would be coming up with the harmonisation dictionary. Meaning that, if you're a researcher and, say, you have some research question you'd like to approach and you need some data in order to do that — you need to do some statistical analysis — first, typically, you would come up with a list of variables that would be relevant to answer that research question. And so what you would do here, is create a harmonisation dictionary, similar to the Excel dictionary file that we saw on the second video. So it's just a spreadsheet that lists the variables and describes them: the labels, the types and the categories and all of that. So, you would do that — you would come up with a dictionary — then send the harmonisation dictionary to the relevant cohorts (or at least the cohorts that will be participating in your harmonisation study), and then the idea is for each of those cohorts (they would have a RECAP node with data inside — data from the from their cohorts, of course) and so they would receive the harmonisation dictionary and then use their own Data Repositories to harmonise their own data, according to that harmonisation dictionary.

So the harmonisation happens in a distributed fashion on this platform, so each node does its own harmonisation. So that's why we have a harmonisation dictionary that we then can distribute to all the nodes. So in this image here, you see... so that would be the starting point, the harmonisation dictionary and we have node A and node B here. Say node A has some data table there, and what you would do is — so the harmonisation process is based on views, the concept of views. So we have a table, right? We know what a table is. We did that on our second vídeo... if you remember, inside of our Data Repository, we created a project and then imported data into it, which created a table to hold that data. So that would be this part right here, the table itself, with the data. And then we can create a view over that table. A view is sort of a virtual table that would extract data from some table to produce some other table — a view, it's not really a table, it's a view — and the way that happens here, so you create a view using the harmonisation dictionary. So you would feed the harmonisation dictionary to your Data Repository, creating a view over that original table. And once you have the view, you can create harmonisation scripts for each of the variables that you need to harmonise. And then the view, taking those harmonisation scripts and the data from Table A, transforms the data according to the harmonisation scripts, and then produces the final harmonised view.

And the final harmonised view would be similar to... if you go to the Data Repository, you see they're listed alongside the other tables regular tables, but it would be a view. And if you click on it, you would see the variables that would be there would be the ones that were defined here, in the harmonisation dictionary. And they would have some data inside and that data came from there, but was transformed using these scripts. So the data is transformed in order to comply with whatever was defined here, in the harmonisation dictionary. So the idea is that every node does this process, produces its own harmonised view and by the end, all the nodes would have a similar table. Similar in the sense that it would have the exact same structure, because it was based on the same harmonisation dictionary.

But of course, it will hold different data because the sources of data are different. And the harmonisation scripts will surely be different too. They will be whatever they need to be in order to transform the data to comply with whatever was defined in the harmonisation dictionary.

So in the end, all participating nodes would have a similar dataset that could then be used for analysis, which we'll see on the next video.

A Real Example of Harmonisation on the Platform [07:12]

So before we actually start, I just want to show you what this process (if it was undertaken by multiple nodes) would look like on the platform, after it's done. So currently, there's another ongoing demonstration project within the RECAP Preterm project, and we're doing something like this. We're involving multiple nodes, we created a harmonisation dictionary, we distributed the dictionary to all the relevant nodes — or cohorts, rather — and basically, they went into their Data Repositories, they used the dictionary that we gave them to create a view over their original data. They harmonised it using some scripts and then produced the harmonised view. And once all of them did that, we were able to do something like this. So this is the page for this particular demo project. This is on the central RECAP catalogue. You can see here... so on the left side here, we have the list of variables that were asked of the cohorts. So the initial harmonisation dictionary had all of these 22 variables. And then we have a column for each of the cohorts that have done this already (that have harmonised their data).

So you can see, for each variable, the harmonisation status on each of the nodes. So for example, on the first column here, we have EPICE-PT, which is the cohort that we are using as an example for this series of videos of the Summer School. You can see they were able to harmonise most of the variables. There's a couple of variables that they weren't able to harmonise. So, if I actually click on this little check mark here, I'll be able to see how they harmonised their data. So they probably had some variable — or variables — on their original data, that they have used to harmonise to this variable. So let's see. Let's click on here. And there you go, so these are the categories of this particular variable and this is the script that they used to harmonise their own data to this variable. And then if I scroll further down, we can see some summary statistics: just frequencies for this variable within the EPICE-PT cohort. So we can see here they had 974 registers. So, this is the harmonisation of this particular variable, for this particular cohort — EPICE-PT — but if we go back here, we can click... instead of clicking on this one, we could also click on all the other ones and see how they did the harmonisation. Then we would see individually, for each individual cohort, how they did that. But we could also click here and we'll actually get a global overview of this particular variable. And we'll see here... so these are global statistics across all of those cohorts, because they all harmonised their own data to this target harmonisation variable, which was “alive_at_bith”. So they all did that and so by the end of this harmonisation, we have about 15,000 records. So down here, we can see that all of them were able to harmonise this variable.

So this is what that process looks like — the harmonisation process. After it's done, we can create something, like a nice overview of all the statuses on all the cohorts. So keep in mind, by the end of this video, we'll have something like this. Or a really small subset of this, because we're just using a single cohort, and I'll just give you two or three examples of how to harmonise a few variables. And then we'll set, for each of those variables, we'll set the harmonisation status — saying if we were able to harmonise it or not — because we have a few different statuses that can happen here. So “complete”, “partial”, “impossible”, “undetermined”. So for example, EPICE-PT was able to harmonise this variable, let's see why. If we click on it, instead of seeing the harmonisation script, we see there's a comment here saying “EPICE-PT does not have this variable, nor any others that could be used to determine it”. So that's why they weren't able to harmonise it, they just don't have the data — they don't have data that corresponds to this variable. So they set the status on their Data Repository for this particular variable as “impossible”. And that's why we see a little red cross here.

So this is the end goal, it’s to build something like this, so you can document the harmonisation process. And then after all of this is done, you can hopefully use the harmonised data for your analysis. We'll see that in the next video. But keep this in mind. Everything we're going to do, for this video at least... the point is that we can get something like this by the end of it.

Uploading the Harmonisation Dictionary [13:52]

So let's go back here. Let's actually start. So we'll basically just go into the node — the test node that we've been using — and use the EPICE-PT pseudo data that we imported on the second vídeo... and I already have a harmonisation dictionary with just a few variables, just for this purpose. And we'll use that, we’ll create a table over that data that we have on our Data Repository and we'll do two or three harmonisation scripts and see what the resulting view looks like.

So let's start by downloading the harmonisation dictionary. So I’ve basically skipped the process of selecting variables that would be relevant to my research question. So we're just looking at the process of the harmonisation itself. So I created an example harmonisation dictionary that you can download too, if you're following this wiki, like I am. So I'm just going to download the dictionary. Let's just have a look. See what it looks like. So it should be pretty similar to the regular dictionary that we saw on our first video. So let's see. You can see there's just 10 variables and we're not going

to harmonise all of them — we'll just do a few. So basically, I selected this set of variables that would be relevant to my research question, for my study. And basically what I would do now as a researcher, I would take this dictionary, send it to the cohorts from whom I would need the data and then they would harmonise their own data — pick the variables from their own original data that match these ones, or maybe they don't have the exact variables that are here, but they can use the data that they have to determine the data for these variables.

So that's where the harmonisation comes in. So this is the dictionary. And again, it has two sheets like we've seen before. This is the variable's sheet, with the names, the labels, the types, the description... You can also see the categories. We have categories for the actual categorical variables like this one, or this one, but we also have categories for all the other ones. Say for example, this one, this is "year of birth". This is not a categorical variable, but it does have a specific code that represents missing values. So we can do that on the platform by just saying that that variable has this particular category, which represents a missing value. That's why there's a "1" here, otherwise, it'll just be a "0". So this is the harmonisation dictionary. Let's close it.

Let's go to our test node that we've been using. Open it here. Okay, so this is our test node. You can see here the things we've done before. We've imported data, we've imported 11 variables from EPICE-PT. So what we're going to do is take the harmonisation dictionary that we just saw, and harmonise this data to those variables that are in the harmonisation dictionary. So like I said, Harmonisation is done on the Data Repository. So let's go there. Let's click up here on "Applications", then "Data Repository".

I need to sign in using my own credentials. So I'm in the repository. I'll go to "Projects" and see the project that we created on the first vídeo, or second vídeo, rather. So I'll click on it. There's a little summary here. We have one table with 11 variables. Let's click on the "Tables" tab, which is the second one, right here. Okay, so here it is. This is our original data. So if this was a real node, this would have all the data for this particular collection event — for the perinatal collection event of EPICE-PT. So we would have many more variables and much more data. So what we want to do, again, is create a view using the harmonisation dictionary and that view will extract data from this table and harmonise it, according to the scripts that we're going to do.

And also, the original table remains intact. So the harmonisation process does not change the original data. It just uses it to create a view with a harmonised version of that data. So how do we create a view? So first, let's upload the harmonisation dictionary because we're going to need that. So

let's go here to “Files”, on this tab. So these are the files we already used on previous videos. So I'm just going to upload another one. Click there on “Upload”, “Choose files” and I'm going to select the harmonisation dictionary that we just downloaded. Click on “Open” and then “Upload”.

Okay, so there it is. So the harmonisation dictionary is there in my Data Repository.

Creating a View over the Original Table [20:18]

So now let's go back to “Tables”, here. Okay, so now we want to create a view, using that harmonisation dictionary. The way we do that is by going here to “Add table” and then “Add view”. Click on that. And the first thing we need is just a name for the view. So we are harmonising EPICE-PT perinatal data, so let's call it something like “EPICE-PT_Perinatal harmonised”.

Second thing is “Table references”. So remember, let me just show you this again. What I said a minute ago, is that we create a view over some table (or tables, actually) and view references tables. So this is what this means: “Table references”. So from which tables do we want to pull data? Well, in this case, we only have one: the “EPICE-PT_Perinatal” table, so let's select that and click on “Add”. So this is the table from which the data is going to be pulled, for the harmonisation. And then the third thing here is selecting the harmonisation dictionary. Click “Browse”. We should see the harmonisation dictionary, there it is. Select that, and now it's selected. And with that, I think we're good to proceed. So we named it we define the references (the tables from where the data is coming). And then the harmonisation dictionary. So let's click “Save”. So it created the view. We are already inside the view. We can see the dictionary of the view — this is the information that was in our harmonisation dictionary. You can see our variables here, the labels, the types, and the categories. So this is the information that we just saw on the Excel spreadsheet. So if I actually go back here, to the root of the project, we'll see the original table and now the view that we just created. You can see it has a slightly different icon there, just to indicate that this is not an actual table. It's a view.

Harmonising to a Categorical Variable [23:18]

So we already have the view now, we already have this part and it has a reference to that table but it doesn't know which variables from there correspond to the variables here. So that's what we have to do now, is do

this part, the harmonisation scripts. So those will link variables from there to variables from here and transform the data of those variables in any way that is necessary. So let's go back here. And let's start by clicking on the view. So this is where we were before. So let's see, let's do a simple one first. So there's basically two ways to harmonise variables here on the Data Repository. There's a graphical interface way, where you can just do everything here on the browser, you can just do it in a more graphical way, which is easier. And that's for if the target variable (when I say target variable, I mean the variables that were in the harmonisation dictionary, so these ones, the ones that we're going to harmonise to).So those are the target variables. And if I'm harmonising a particular target variable, that is categorical, we can just use the web interface here. It's very easy to do. And then the second way is if the target variable is continuous, we have to actually write a script. But we'll get there in a minute.

So let's do the easier one first. So let's harmonise, for example, this one, the sex variable, let's just click on it. And, like we saw on the Excel spreadsheet, we have three categories: “0” for “male”, “1” for “female” and “9” for “missing”. So these are the categories that we need to harmonise to. So this is our target variable with these categories. Now we need to go into the original EPICE-PT data and find a variable that we could use to harmonise to this one. So let's go back to the root of the project, and select the original table. And now let's find a variable for the sex of the baby. Let's see. There it is, “Sex of baby”. The name is “a8”. Let's just click on it and see what categories it has. So it's similar to what we need, but it has different codes: “1” for “male”, “2” for “female”, “3” for “undetermined” and “9” for “missing”. So it's quite similar, but it has different codes, and also an extra category that we don't have on our target variable. So what we need to do now is map these categories to the ones that we have on our view for the sex variable. So let's go back to the view and let's just remember that this is the variable that we need from the original table: “a8”. Let's go back to our view, which is this one. And we want to harmonise this variable, “sex_bin”, let’s click on it. And we need to harmonise those categories that we just saw in the “a8” variable of EPICE-PT, we need to map those to these three.

So the way we do that is by clicking here, “Derive”, and selecting this option “Categorise another variable to this”. This is what we want to do: take one of the original variables from the original table and categorise that variable to this one, “sex_bin”. Okay, so let's click “Categorise another variable to this”. And now we have to select the original variable from which we want to pull the data. And remember, it was this one, the “a8” variable, so I'm going to select that one and click “Next” down here. Okay, so now we get a summary of what the original categories are. And like we saw, it's “1”, “2”,

“3”, and “9”, corresponding to “male”, “female”, “undetermined” and “missing”. We also get some frequencies here. So now we need to say that, for example, this one, this category, from the original variable, is going to be mapped to a new value. So this column here is where we select which of the categories on our target variable correspond to the original ones. So the original variable had “male” with code “1”. And if you remember, on our target variable, we had just “0” for “male”, “1” for “female” and “9” for “missing”. So what we want now is the “0”, right? On our target variable, “male” is coded with “0”. So we need to map this one, this original category to “0” on our target variable. And you can see we have the three options from our new variable here. So we want to map “1” to “0”. We’ll do the same for the “female” category which, in the original variable, is coded with “2”, but now we want to code it with “1”. Now this one, we don't have this category on our target variable. We just have “male”, “female” and “missing”. So what we could do here is map this category to our “missing” category on our target variable, because we don't have this category on our target variable, so we'll just map it to “missing”. So with code “9”. And also, we should tag it as being missing (you'll see what this actually means in practical terms in a minute). And then also, the original variable also had an extra category for missing values. And we also have that one. So it's automatically mapped to the same category in our target variable, which is “9”. And it's also missing. Now, these last two lines, what these mean is... so the original variable is coded with “1”, “2”, “3” and “9”. But it could also have missing values. Missing in the sense that the data is just empty. It's not “9”, because if it's “9”, then it fits here in this category, but you could also just have “NAs” (i.e. not available). So we will map those... we can see here that actually, there are no “NAs” on the original data. But anyway, if there were, we would map them here, we can map them to our missing category, right? We'll just map those to “9” as well. And then also, this asterisk represents any other value. So any value other than “1”, “2”, “3” and “9”, or “NA”. So if, for example, there was some mistake on the original data, and you had, let's say, “6”, for example, which does not fit in any of these categories... so that was clearly a mistake. So that will fit here. So I would think that we would map that to “9” as well. So we've considered that to be a missing value, because it doesn't fit on any of the original categories. And mark that is “missing“ as well. So let's see. The mapping is done, so let's just click “Next”.

Now, before we click on “Finish”, when we're harmonising here on the platform, we can actually have a quick look at what the result will be after we click “Finish”. So we can click here and we'll get a summary of what the data will look like. So we will have 25 males, 22 females and 3 missing. So basically, you could use this intermediate step before you hit “Finish” to

check whether the mapping you did on the previous step is actually what you mean to happen. Because you can make some mistake here. So if you get to this part and realise you've made a mistake and need to go back, and then you will fix the mappings here, and then you could run this again and see if the frequencies look right to you.

You can also click here on "Values", you will actually see the values here. So we only have 50. I actually think this is just a subset of the whole data. So basically, when you do this harmonisation using the graphical interface like we did back here, what the Data Repository actually does is it creates (in the background), it creates a script that does this process of mapping the original categories to the new categories. And you can have a look at the script. You don't need to, because you can just do this in the graphical way. But if you want to look and see what the script looks like, it's something like this. So this is the language that is used in the background. So basically it just means: "take this variable from the original data and map using these rules". So "1s" on the original data will be mapped to "0s", "2s" will be mapped to "1s", "3s" will be mapped to "9s", "9s" to "9s", and then these last two are the ones that we saw here: "NAs" and then any other values will be also mapped to "9".

So, if we look at this summary and we think it looks good to us, then we can click "Finish". And basically that's done. This variable has been harmonised. If we go back here to our harmonised view, we can click on "Values" and we can see everything is "null" for now, because we haven't harmonised all the variables. The only one that we have harmonised is this one, "sex_bin". And you can see there's a bunch of zeros and ones. There's a nine right there. So because we did the harmonisation for this variable, in this view, what the repository has done is pull the data from the original table, use that script that we just saw, and transform the data. And now we can see the result of that. This is the harmonised version of that original variable from EPICE-PT. So that's one example of how to harmonise to categorical variables. If your target variable is categorical, you can just do this. It's very easy, right? It's just mapping one category to another. So that's the easy way for categorical variables.

Harmonising to a Continuous Variable [36:25]

But like I said, I'll just give you an example of how to do it if the target variable is not categorical and it's actually a continuous variable. So let's now look at one example of a continuous variable harmonisation. Let's do, for example, "age_discharge", let's click on that one. So this is a continuous

variable. Although it does have, like I mentioned before, it does have one category but that does not mean that it's a categorical variable. This is just to represent missing values for this variable. And as you can see here... so this is “age at discharge” in years, we can see the unit here. It’s coded in years. So again, like we did with the previous variable, we have to find out which of the variables from EPICE-PT we could use to harmonise to this one. So let's go to the original dataset. And let's try to find out what variable we could use for that target variable, which was “age at discharge”. So there's a variable here called “age_out”, and the label reads “Age at discharge home, or another place, in days”. So this is almost exactly what we want. However, this is coded in days, and our target variable is coded in years. So we can use this one (i.e. “age_out”), but we'll have to do some harmonisation. So let's do that. So let's remember the name: it’s “age_out”, the name of the original variable, and it's coded in days. Let's remember those two things.

Let's go back to our view here, and select the “age_discharge” variable. There it is. Let's click on that. And now we want to harmonise that other variable — the “age_out” variable — to this one. But because this variable is not categorical, we cannot do as we did with the sex variable. So this is the point where we have to do some manual scripting. And for that, we can just go to this tab right here: “Script”. Click on it and then we have a space here. If we click on “Edit”, this is where we will write our script. So by default, it's “null”, but we'll just get rid of that and then we have to write our script here. So the language that we have to use is called *MagmaJS*. Let me just go back here to the wiki and then if I go down to this part, which is what we're doing right now... so if the target variable is continuous, we have to use *MagmaJS* scripts. So this is a language that is based on JavaScript. And you don't have to know JavaScript in order to do this. You can just use the documentation here. I'll just click on that. And there's a few functions that you can use for your harmonisation. Let's see, for example, this one, the “map” function is the one that is automatically used in the background, like we saw, whenever you're harmonising categorical variables, the Data Repository itself just automatically creates a script and uses this function. And there's a lot of other functions that we can use. I think for this one, we will have to... Let's see, the original variable is coded in days. But our target variable is coded in years. So what we can do is take the original values (the original data), and just divide them by 365, for example, and we'll convert from days to years.

So there's a function here that we can use for that. I'll just search for it. There it is. So these are functions for numeric values. So we can use this “div” function. And we'll just click on it and see how we can use it. So it's something like that. So first, we have to reference the original variable. The

way we do that is just by using a dollar sign, and then just the name of the original variable, which in our case, if you remember, it was called “age_out”. So we'll have to do something like this. And just write “age_out” there. And then “div”, which stands for division. So we want to take the values from that variable and divide them by whatever we put in here. So in this case, it's dividing by some other variable, but in our case, it's simpler than that and it would actually be something like this. So here, they're using the variable “height” and dividing the data from that variable by 100. So this is similar to what we want. So let's go back here. So first thing, we need to reference the original variable by using that syntax that we just saw, which was a “dollar sign”, then parenthesis, and then we just write the name of the original variable, which was “age_out”.

If we just leave it like that, there's a button here called “Test” and then we get the same thing that we did before (with the other categorical variable), we get a summary of what it will look like once we save the script. Let's just actually look at “Values”. So these are basically just the values from the original data. Because we didn't really harmonise it, we just referenced the original variable. So what this script — the way it is right now — it’s just pulling the original data and doing nothing to it. So the values we're seeing right here are the original... or are the same as the original values, which were coded in days. So these represent days. What we need to do now is divide these values by 365, to convert them to years. Let's close that.

Let's use that function that we just saw, which was called “div”. And we want to divide by 365. So take the values from that variable on the original table and divide those values by 365, effectively converting them from days to years, which is what we want for this variable. So if we click “Test” now and look at the values, now we can see that those are the same values but just converted to years. We can click on “Summary” to see a graph of frequencies. And so we clearly have an outlier here, because most of the values are on that range there. And we have an outlier here. And that's because we forgot something. We forgot to deal with the missing values. I'll show you what I mean.

Let me just open up a new tab. Open the original table on a new tab here. So this is the original table. And I want to select the original variable, which is this one: “age_out”. And we can see, we don't even have to click on it, we can see we have the name, the label, the type and categories for missing values. So this means that if there's any record for which we don't have this variable, on only original data, that would be coded with this number. And that's probably what we're seeing on our graph here. That outlier is probably where that comes from. So we can see that what is probably happening is that it's taking those missing values from the original data and considering

them to the actual valid values, but they're not. So what we need to do is we need to tweak our script and we need to map those original “9999” on the original variable “which represent missing values), we need to map those to this. So this is the code, which is similar to what they use on EPICE-PT, but now we want to code missing values for this variable as “99”.

So all we need to do in our script is map any “9999” that are found on the original data, we need to map those to just “99”. So let's go back to our script. And instead of just doing that, we need to write a small condition here. So let's reference the original variable, again: “age_out”. So if the original value is equal to “9999”, we want to return just “99”, which is the code that we want on this target variable for missing values. So if we find that, we convert it to “99”. Else, we just do what we were already doing. So if this condition is true... So basically, the script is executed for each row on the original data. So this condition is going to be tested 50 times, which is, I think, the number of rows that we have on our original data. So if one of those rows has “9999”, it will convert that to just “99”. Otherwise, just do that for any other values. Just do what our original plan was: just divide it to convert it to years. So let's click “Test” now, and see if we actually got rid of that outlier. Yes, we did. So now that we mapped that original outlier — it was not an outlier on the original data, it was a missing value — but because we weren't treating it as a missing value, it ended up being an outlier on our harmonised view. But now we’ve mapped it to the correct category that represents missing values on our target variable. We got rid of it, it doesn't show up here anymore. So now that looks okay. This is useful, right? That we immediately see a summary of what the harmonised data will look like. It's an automatic feedback that you can look at and see if you maybe did something wrong on the script and then you go back to the script and fix it and look at it again. And once you're happy with it, and I think we are, we can close this, then just click “Save”. Okay, so this is our script for this variable.

And let's go back here to the view. This is our view. And now if we click on “Values” again, we will see... so these are “null” because we haven't harmonised them. We harmonise this one. And we haven't harmonised these. So let's keep going here. Age at discharge. And there you go. Those are the values that we harmonised. And you can see here, there was one missing value, at least one that we're seeing, on the original data that was mapped to “99”. And it's “99.0”, because this variable is... if we go here and look at the dictionary... this variable is a decimal. So although it's “99” there, any values will automatically be converted to decimal. So the Data Repository just added “.0”.

Harmonising a Continuous Variable to a Categorical Variable [51:40]

But basically that is it for harmonisation. So there's our harmonised data. So we harmonised a categorical variable using the web interface. That's the easier, easier way to do it. There's actually... let me just quickly show... because I basically split this process into two options. So if your target variable is categorical, you just use the web interface to remap the categories. And if it's continuous, you do a manual script.

There's actually a third option, which is... let's say if the target variable is categorical and the original variable is also categorical, you can do what we did here, which was mapping the original categories to the new categories.

But that's if the original variable is also categorical. What happens if it isn't? If the original variable is continuous and, let's say, we want to map it to a few categories. Let's say you're harmonising an "age" variable. So the original variable is a continuous variable that represents the age of the person. And then on the target variable, you have age groups, for example, 0 to 10, 10 to 20, something like that. So you need to map the original continuous variable to a target variable that is categorical. So you could also use the web interface to do that. So let me just select a random variable that I think would be continuous. So this one I think, is continuous. And you get something like this. So if you try to map a continuous variable to a categorical variable, you can do that via the web interface as well. You can just map continuous values to ranges of values. So that's the way you would harmonise continuous variables to categorical variables, you could also use the web interface for that. And any other type of harmonisation would have to be done manually using a script.

Publishing a Harmonisation Study [54:27]

There is just one last detail that we need to do. Now, if you remember, at the beginning of the video, I showed you this table and I told you that this was our end goal to have something like this — or a very small subset of this. So the thing that is missing... we've harmonised two of the variables, so we need to set the harmonisation status for those variables, saying that we were able to harmonise them.

So remember, the possible statuses are these four. And so before we can have something like this, we need to set the status. So let's go back here to our harmonised view. So the ones we did harmonise, were this one and also this one. So the way we set the status for those variables, is just selected

them, like I just did and then up here, we can click on “Apply attributes“, “Apply annotation”, and you can see it's already selected for us by default. But if it wasn't, you could just look at the drop down list and look for “Harmonisation” and then “Harmonisation Status”, and then you choose the status here. So, we were able to harmonise them, so let's set the status to “Complete” and then click “Save”. And just so we can see what it looks like, let's choose one that... I happen to know that this one, if we were to harmonise all of these variables, I know that it EPICE-PT does not have this variable, nor any others that we could use to determine this one. So this one would be impossible to harmonise. So let's set the status of that one to “Impossible”. So select “Apply attribute”, “Apply annotation” — it's the same process, but instead of choosing “Complete” now, I will just choose “Impossible” and save.

You can see here, this last column now has the status that we just set for those variables. So after having done this, there's an extra step that I'm not going to do. At least not in this video. I've done it previous to recording the vídeo, just to save a bit of time. But I will just quickly explain what that step would be. So this table that I showed you, this is on the catalogue. This is the catalogue of the central node, so the equivalent of this table for us would be on the catalogue of the node that we are currently using: our test node. And remember that every single information that appears on the catalogue of a node, that information was published in the underlying Study Manager. So what we would have to do is go into the Study Manager of our node and create this... Let me just go back to this part. So this was in the previous vídeo, where we created a study, a population, collection events and linked all of that to the data and the Data Repository. But we did not talk about these right here. So this is what was missing that I just mentioned. We would have to go into the Study Manager and create a harmonisation study, add a population to it and then create a harmonised data set and that — much like this one — would be the link to the harmonised view that would be here in the Data Repository.

So I've created all of this already. And I've linked it to the data that we just harmonised. So I've created this and linked this to our harmonised view and I've published it. So let's go into our test node. Let's go into the catalogue. And you can see there's two datasets. So one of them is the one that we created in the previous video, it's this one. Now that second one is this one that I've created offscreen. So let's click here: “Datasets”. You can see there's our dataset that we've created. This is the original dataset from EPICE-PT, and then I've created this one just to show you what that harmonisation overview will look like. Let's click on “Read more”. And you can see we only have one study, which is the one that we are using as an

example. If I scroll down, you will see the status of harmonisation. Those are our 10 variables from our harmonisation dictionary. And then we have just one column because we only added one cohort. And most of the statuses have not been set because we only set this one for the sex variable, and this one for the age at discharge, and we also set that one just to have one example of when the harmonisation is impossible. So we can also click on this one, for example, to see how that variable was harmonised. We can see the script that was automatically generated by the Data Repository when we mapped the categories, the original categories to the new categories on our target variable. We can also see some summary statistics. So this is our little example. But of course, in a more real setting would get something closer to this.

Closing Remarks [1:01:15]

So now, after having done this, what you would typically want to do is use the harmonised data in some way... run some statistical analysis on it. So that's what we're going to see in our next two videos. But before we get to the actual analysis, there's an intermediate step that we have to do, which is manage permissions. So the data is there. It's harmonised. Now who do we want to access those data and in what way? Because there's different levels of permissions that we can grant to users. And depending on the level of permission, they will be able to run different sorts of analysis. So before we get to the analysis, we need to quickly talk first about managing user accounts on a node and managing permissions... what those users are able and not able to do on your node. So that's what we will be talking about in the next video.