HRSA/MCHB 2007 FEDERAL/STATE PARTNERSHIP MEETING

Building Blocks for Promising Practice Models

October 14 - 17, 2007

Record Linkage 201: Vision for Data Integration to Action and Implementation

RUSSELL S. KIRBY: --identifying variable. It's not a good idea. I know one state that links hospital discharge records with vital statistics to create its birth defects registry, and if the hospital discharge record doesn't have a Social Security Number, it falls out of the analysis. And guess what? The data they get from their regional perinatal centers, none of the records have Social Security Numbers, and guess where most of the high-risk babies with birth defects go for a treatment.

So, again, don't rely on a single identifying variable. You've got to do something more comprehensive than that. And on the other hand, when you get finished, remember the *KMF, you can create a common identifier that you can put on each of the databases once you've linked the records, once you're sure that they really--each record in a pair of links, and then you can use that as a basis for doing analysis in the future. You don't have to store all of your linked data in an N-by-M matrix. You can use SQL and other relational database strategies to analyze the data, but you've got to have a linker that identifies that record I in database M has a common with record J in the other database. So, now, we'll go

back here.

So, these are just some examples on the birth certificate and, say, on a newborn screening record, looking at how the variables might be named differently. Now, in SAS, you have to do a rename so that the variables are identically named in the two data sets. I would recommend that you don't do a rename, but you create new variables that have the common names, so you always retain your original data so that you can go back to it without having to backtrack several steps later. So, mother's information, infant's information, again, some of the variables that might be in common.

Now, notice that this pair of variables here, these might not necessarily be exactly the same date. Newborn screen might be on the same day the baby was born, but it's probably going to be done shortly before the baby is discharged, which--even with the *year of early discharge, which it's probably going to be the next day or the--or two days later. Also, children's names can change between the birth certificate in the newborn screen, and in fact, if anything, they might be more accurate on the newborn screen, which is not regarded as a legal document in a way that the birth certificate is. And sometimes, the birth certificate file takes a while to get caught up.

And then, there could be related fields that you could use that might not specifically identify the individual, but can really enhance your ability to match.

And two that are particularly useful for that are the zip code of maternal residents and, potentially, the hospital or facility ID. Potentially, those can be very helpful in a lot of ways. Then, you need to look for missing data in the linkage variables. And then, of course, you have to decide what you're going to do when you find missing data. You don't want to throw the record out, but that particular record, if it has missing information--how many of you have--how many of you used SAS to do record linkage? Some. Okay. Well--so, one of the common things you do when you do a data step with a merger and update statement is that you'll get this error message followed by about 40 pages of printout, and it tells you that your master file has duplicate values for the bi-variable. Well, that's--when you have missing data, all of the records that have missing data are treated the same. They--that's--their value for that particular variable is missing. And so, you've got to think about what you want to do with this in order to clean up the data.

Also need to be very concerned about duplicate records, and specifically, records that share the same values for the whole vector of matching variables that you're going to use. And so, again, you have to think about what are you going to do when this happens because you don't--if you do that, you *outrun the potential of matching the same record in the master file to two different records that may actually be on different individuals, so you have to think about that. Then, how do you decide what variables you want to use for linkage? Which ones are the best? One of the issues you need to think about is how much missing data is there.

You have to think about what kind of information is contained. But really, the key thing to think about in terms of this, I don't have it on the slide, but the key thing you want to think about is parsimony. What you would like is you would like to identify which variables within your data set can uniquely identify all of the individual records in the data set, so that instead of having these 11 variables to uniquely identify all the records, maybe you can do it with four or five, or what-have-you, because your algorithms are going to work much better if you get down to the point where there are no duplicates within your by-groups, that's definitely important. Then, of course, and this is what I just said actually, use the most discriminating combination of variables, first, in your algorithm, and then loosen criterias as you go on.

So, in a deterministic approach, you are going to start out with some fairly strict criteria in your first step. The records that match in your first step, you might put in a separate database, and then the records that don't match, you go to the second step, and so on and so on. I've actually seen--the most complicated one I've seen, I think, had 242 iterations in the deterministic linkage. It was also the one that subset it out, all the ones that didn't have Social Security numbers. So, they were very assiduous in trying to match the records that did have Social Security numbers. But anyway, you can have a very complex algorithm.

It's important to create IDs in your data set that are not necessarily the case IDs that came with the data set, but IDs that enable you to track where your record

came from and what its position was in the original data set. This is--in SAS, this is pretty simple to do. It just shows you in a very short program how you can do that. And then, when you sort by your vector of linkage variables--well, okay, what would happen if we did this? Say, we have a file for birth certificates and Medicaid and we do data link and merge and run, what's going to happen? How many records are we going to get from that merge, and how much SAS log output with the 40 pages of error message--we're going to get a lot of records. We might potentially get N x M records in it, and we might actually crash the whole system because it uses up all of the available disk space, depending on how large your files are.

So, it's very important that you--you need to use bi-variables. And it's also a good idea to use additional enhancements on your merger update statement that identify which file each of the records came from and direct the files, the output, to different locations. So, you might--instead of having data linked, you might have a data linked, a data unlinked (inaudible), the data unlinked med, and so you can put the records back together and facilitate the next step.

‘

And so, when you--another thing that's important to do when you sort the files, you would like to identify the duplicate records and set them aside for separate processing, so that you--because the duplicate records are going to cause potential errors. Then, another thing that's interesting to think about is this: the potential of allowing full replacement prior to running each new iteration. Now, if

you remember that this is Record Linkage 201, but one day, you guys all take Statistics 101 and you'll learn about sampling theory and the idea that there's this theoretical population in the sky and we take repeated samples of and cases and we can build our sampling distribution from it. Well, that's sort of what this is about, too. There's the possibility that your particular record, that you matched on the first iteration and are no longer looking at anymore because you put it in the matched file, it might potentially match with additional records in your master file. And you might actually would like to know about that because we'd like to believe that all of our, particularly, vital statistics date are perfect and there's never a situation where there are duplicate birth certificates on the same individual. But in fact, it doesn't happen often, but it can. And in fact, it could even be that there could be peculiar things with the person assigned different genders and all kinds of crazy stuff, so you have to be careful.

It's also important to keep track of which level of the linkage it should be--if your pairs of records were matched. So, you keep track of that and evaluate how they actually got into your file. Don't discard the records that fail to match because you want to continue to work on them. Your goal is to try to match all the records. Okay? And then, you want to remerge, again, to get your unlinked data sets, and then pass them onto the next step, so that they have an opportunity to go through again. Even if you do it the thing with replacement, it gets a little bit messier at the end in terms of figuring out what had been, it's not impossible.

Okay. Then, when you're finished with this, you want to combine all the linked data sets. So, you've got to link one file, link two, up to link nth, from all your different *adorations, then you want to put them all together, you want to analyze the unlinked records and see if you can figure out what's going on.

When I did my project, linking the NICU records of the birth certificates, I only had 85 NICU records that I couldn't match. It turned out that 39 of them were subsequent NICU records where the--where the record actually matched with another NICU record, but these were all records that--where the baby had been transported to children's hospital, which didn't have any births, and so, there wasn't enough information to actually match those. But I was able, when I looked more closely at them to figure out what the original NICU record was, so that I was able to incorporate information from those second NICU hospitalizations with the original. But anyway, you want to be very thoughtful about how you do that. And then, of course, evaluating the quality of links. And you might find, when you're done with all that, that the way that you did the linkage really wasn't the best, and you can find some ways to enhance your process for your second time to.

So, just very briefly on probabilistic record linkage, this is a methodology that uses probabilities to determine whether a pair of records refer to the same individual, and we calculate weights that quantify the likelihood that each particular pair is a true match. It's computationally intensive because it really

does compare every record in data set N with every record in data set M. And the probabilistic weights that we applied can be either nonspecific or potentially value-specific, and we have the opportunity to set what those are. So, nonspecific weights might be--whether there's agreement or disagreement on a particular variable. So, we might--if we have--and we can give different variables stronger weights depending on their agreement.

So, if we have a match on gender, well, it's about--what is it right now? It's 105 or 104 males per 100 females, so it's about 50-50. So, that's--I mean, it's nice if they match, but there's a lot of chances for error based on that. Whereas with date of birth, there's still potential for error, but it's a much more specific variable. So, it might potentially give us a--give a higher weight for that. But on the other hand, if the records--if the pair has disagreement on gender, we would give that a higher penalty in terms of the weight because, again, it's--probably is wrong, probably isn't a match.

And then, value-specific weights, where they put the specific values for a variable. So, for example, using initials, where there are 26 initials, the letters of the alphabet, we could give value-specific weights *where agreeing on Z might get a higher weight than agreeing on S, or agreeing on E, for example, and then disagreement on that initial would get a higher penalty, and so on. So, the weights then objectively reflect our confidence in a match. And then, we can use individual choice in terms of what---where do we set the bar below which we say

it’s not a link. And again, if you use this particular process over time, you learn from it, and you have empirical data you can use to set weights in the future once you get started.

Okay. So, how do you get a probabilistic linkage application? Well, you can go to Wal-Mart, I guess, and find one. But it’s not something that most people who shop at Wal-Mart are gonna want. So, a lot of people write their own probabilistic code. There are some software packages available. Some are expensive. Some are difficult to use. Some are not all that expensive. Some are available as a freeware. And these numbers are old, and I don’t actually know what the current ones are. But AutoMatch is one program that a lot of people use, particularly in business, marketing applications, and it’s very expensive. But if you happen to be somewhere where they bought a copy when it didn’t cost that much, like in 1989, it’ll still run and you can use it, and it’s fairly robust. GIRLS is another program that’s fairly useful. This is a very high-end, runs in oracle but, again, very, very useful. LinkPro which is now, I believe, called LinkPro3--these are the numbers that I had a couple of years ago. I don’t know what the current prices are. This is the program that was developed by Andre Wajda, who developed the populous population of data warehouse where the province is in Manitoba in Canada, which is fairly good. It turns out there’s a program called Lynx, which is basically a set of integrated SAS macros that can be obtained for free. I was able to obtain it for free. I didn’t actually get a good answer on whether I can give it to anybody that I happened to run into, so I don’t know about that. But I could certainly tell

you how to contact the people if you're interested in it. It's a very rigid program from a programming point of view. Everything's got to be done exactly right, almost like a Fortran program where everything's got--the syntax has to be very specific.

Link King is another program that's available that you can download for free. LinkPlus is a program that's available on the CDC website. You don't need to memorize this. Just LinkPlus will take you right to it in Google just fine. And then, it's developed for cancer-record linkage, but it is easily used for other purposes. Febrl, if you like programming and you happen to already know how to use Phyton, which is a newer programming language, this is actually a very robust shareware but it's a freeware program from Australia National University, which has some possibilities as well.

Okay. Now, I just want to close for the few comments about evaluation. It's absolutely essential to evaluate the results of the linkage. And one of the advantages of probabilistic methods is that it's built right into the methodology and you can't avoid it because you have to--in order to decide how you're going to determine which records are matches and which aren't, you have to engage in this process, so it's very important in terms of that.

Now--document, document, document. Even if you plan to stay in your job for the next 30 years--and I know all of us do--document everything. Document your

programs. Document the output. Save the log files. Create data dictionaries, and don't put them on three-and-a-half-inch floppy disks and take them in a drawer. Make them somewhere where they're going to be accessible. Retain the details so that it's possible to keep track of what you did, what data you used, and so on because this is very important. Now, data warehouses, this is a trendy topic nowadays. And I'm sure that it's been a subject of discussion in pretty much every state. And it's a nice thing, I think. But we have to be careful about what exactly we're doing with the data warehouse or we might get something that doesn't do up what we wanted. And one of the issues is, again, when the data warehouse is based on linkages that are solely based on unique identifiers, that's going to be a problem. You always want to find out how the linkages were done and why. And, of course, it means different things to different people. Data warehouse for Wal-Mart means a very different thing and it's going to mean in a public health agency for one thing. But it might be that it's a perfect one to many or many to one linkage repository. It might be that's a library of databases that may or may not be linked at all. They're just stored in the same place, and that could be a warehouse. And I like to call the bigger challenge the (inaudible) data cube because you could have a situation where you have basically a warehouse in three dimensions, and some areas are fully populated and linked across different sources, others are holes. Others are link to data that are aggregated at different levels like households or census tracks or counties or whatever it might be. And then--and again, some areas might be completely empty.

Okay. So, again, evaluate before you analyze. Don't assume a linkage has been done correctly even if you did it yourself, especially if you did it yourself. Always have some self-doubt about that. And then you have to do linkage whether you're--you have to do evaluation whether you use deterministic or probabilistic algorithms. You need to compare the values on non-linkage variables as well as those that you use in the linkage. In fact, ideally, you should compare values on variables' comments if the data said that you didn't use in the linkage to evaluate. But usually, we don't have enough of those to do it. And then create ParaWise linkage scores and throw out those that don't meet your minimum criteria.

Then if you publish reports or submit manuscripts, make sure that you talk about how you did--how you evaluated the results and give the details about your linkage because you don't want to be in the situation here. It's almost Halloween again and that witches are saying, "What I really hate is knowing that I'm doing this exactly the way my mother did it." And you want to avoid that, but then again, we've always done it this way. So ask yourself, "Why do we do the linkages once a year or once every five years? And could we consider changing our processes from information systems to informatics, and building mechanisms for linking records as they're generated?" Think about those kinds of things. Could we think about ways that we could automate the process of creating longitudinal records across our databases? All those kinds of things we need to think about. So I--have I run out of time or can I tell them the 10 commandments of record linkage? You want to hear this?

UNKNOWN SPEAKER: Sure.

RUSSELL S. KIRBY: Okay. So I mentioned about the bible before, and there are other commandments besides the ones that we have in the Alabama Supreme Court a few years ago, and these are some that relate to record linkage.

So the 10th commandment, thou shalt not take the name of thine software in vain, particularly if you wrote it yourself. Number nine, thou shalt not covet thy neighbor's database nor shalt thou hoard it (inaudible), okay? Number eight, know thy purpose. This actually should have been one of the original 10 commandments. It really, really should have been. And maybe there are more than 10. Number seven, thou shalt not merge without thy variables. I mention that one but I have a colleague who was on a doctoral dissertation committee, and the student had all kinds of bizarre findings that didn't make any sense in terms of the previous literature. And when he'd asked her to bring her program and this is actually what she did, very important. Okay, number six, thou shalt checketh thine statistical software log before thou proceeded to thy next step or process. So remember that. Number five--and by the way, if you use SPSS, I've got some students who--and I've been using mostly SAS since 1981 but I've got some students who learned SPSS, and they claimed that they can't get their log files. Well, you can get your log files from any statistical software, so that's not an excuse. Okay, number five, thou shalt protect the privacy of those whose

information is recorded even as you use it to conduct the linkage analysis. But the fact that you're concerned about privacy shouldn't be a reason for not having access to that information to use as long as you're going to protect it appropriately. Okay. Thou shalt not bear false witness against the inconsistent values of variables common to the two data sets nor because thou fail it to evaluate thine linkage results. Then, of course, another important commandment, know thy data. And then, thou shalt not underestimate the complexity, time commitment, and staffing required to conduct the record linkage nor shalt thou over-estimate the time needed to conduct analysis. Usually, once you have the linkage done, it takes a day or two to do the analysis but it can take weeks to do the linkage. And then finally, show humility to others even those who doubted that the tasks that you've accomplished could be done. So Plato told us the life which is unexamined is not worth living, and that's certainly true. But we also have to remember that the database which is unexamined is not worth analyzing. So I'll leave you with that final thought. And I've actually used up most all of our time, but I would be happy--if any of you guys want to rush off for break, that's fine. But if you want to stay and ask questions, I'd be happy to answer them. Anybody? Okay.

UNKNOWN SPEAKER: And then you talked about that when you do the data linkage, especially you use the join, how about in arc view, do you have some two-table join in that line for that *long?

RUSSELL S. KIRBY: Well, that's--in arc view where you're--typically if you're doing a join on geo-codes, those are going to be fairly accurate and specific pieces of information, and that's probably going to be okay. But if you're working with--if you're using--just using a join on a Medicaid number between two different data sets or a Social Security Number and just using that alone, that I think would be a difficult thing. Yeah. Other questions or comments? Okay, I've--

UNKNOWN SPEAKER: Comment.

RUSSELL S. KIRBY: Yeah.

UNKNOWN SPEAKER: Number one, the interstate exchange is advance to the point that it does have a name. It's called STEVE, State and Territorial Exchange of Vital Events.

RUSSELL S. KIRBY: Yeah, I basically heard about STEVE. Yeah.

UNKNOWN SPEAKER: And our fellow colleague, Paul Johnson is a member of the National Committee.

RUSSELL S. KIRBY: Good.

UNKNOWN SPEAKER: And he has worked with both Russ and I.

RUSSELL S. KIRBY: Yeah.

UNKNOWN SPEAKER: And he carries on that work that Russ started, okay.

RUSSELL S. KIRBY: Yeah. But it's good. Yeah.

UNKNOWN SPEAKER: (Inaudible) comment is on your complexities of Medicaid matching, I recommend a forthcoming publication of your colleague Dan Brownstein.

RUSSELL S. KIRBY: Okay.

UNKNOWN SPEAKER: Who's been working with us and were actually in Medicaid Arkansas data. And the complexities of taking--trying to account for all pregnancies in the Medicaid data, how many and how hard that is. She really details it out very completely but it takes you a long time to work with those tables to figure it all out because it is a complex process.

RUSSELL S. KIRBY: Okay. Yeah. But it--I was sure there were some people in the room who were working on that issue. Yeah. But the Medicaid--just because Medicaid data are complex and messy, however, is not a reason not to try to work with them because it's--and for that matter also, the State Child Health

Insurance Program data is another potential source that we really ought to be thinking about how to use to look at some child health issues. Anybody else? Okay, you got my email address there. Now, there was a notepad that was going around, and we need that to come back to Scott or me so that--what--one way or another, we’re going to email the PowerPoint file from this presentation to everybody who put their name on the list. Okay.