PANELS AND ENTITIES: DATABASES AND COSTS

Halliman H. Winsborough

CDE Working Paper 87-1

Panels and Entities; Databases and Costs

Halliman H. Winsborough

Center for Demography
University of Wisconsin--Madison

This paper is a discussion of papers presented in the Data Base Management Issues for Panel Surveys session of the International Symposium on Panel Surveys, Washington D.C., November 20, 1986. It's preparation was carried out (in part) using the facilities of the Center for Demography and Ecology, University of Wisconsin, Madison, which receives core support for population research from the National Institute for Child Health and Human Development(Grant No. 05876).

PANELS AND ENTITIES; DATABASES AND COSTS.

Let me begin with a disclaimer. I'm off my turf. I'm a demographer-sociologist; these papers were written by survey researcher-economists. The authors are insiders in the panel survey business; I'm an outsider. My last data project was to shepherd the making of public use samples from the 1940 and 1950 Censuses. So I'm from the repeated cross-section part of our business.

I begin with this disclaimer because I'm not sure what these papers are for. It is clear that all the authors think that panel surveys are so unusual and complex that they need something really special in the way of a computerized data management scheme. Two of the papers sound as though they are working their way towards a proposal to support a database system design; the third is in the middle of using a fairly complex relational system. I sense an argument going on. It sounds like a family quarrel of long standing; one that you come in on unawares, say, over dinner. Since you don't know where the argument comes from, it is hard to figure out where it is going. Usually in that kind of situation an, outsider should shut up. But that's not the discussant's role. Besides, I'm puzzled by some of the assumptions that seem to underlie the argument.

My quandary centers around three issues. They are:

1. What's so special about panel studies that data management issues arise and are discussed exclusively in this context. Aren't these issues which arise for all public data?
2. Why would any group of social scientists even think about writing their own data management programs. There is so much to know about databases. So much skill and facility is available in packaged systems.

3. Whose responsibility is it to do what to the data? What should we expect from the collectors and what from the analysts?

1. DO PANEL DATA REQUIRE SPECIAL MANAGEMENT?

A first assumption that winds through all of these papers is that panel studies present special data management problems. Two reasons are given. First, they are large and complex. Second, they involve a large number of "entities" and there is a problem of tracing some of these entities through time.

Although the first of these reasons, size and complexity, should make one think seriously about data management issues, I don't think it is unique to panel studies. There are lots of large and complex data sets in regular use. A year's worth of CPS files is many times larger than the PSID. Even ignoring CPS's panel aspects, a year's worth is very complex, especially if you keep careful record of the design elements. Some of the "life history" type surveys are complex. Several surveys about to go in the field are, I think, similar in complexity, if not size, to SIPP. One example is the National Study of Families and Households. So, there are a lot of complex data, whether paneled or not, and more on the way.

That's not to say that size and complexity are bad reasons for panel analysts to worry about data management; only to note that data management issues may be of increasing importance in the social sciences at large. Perhaps we should be addressing the issue in general rather than just in the panel context.

The second reason given for the special data problems of panel studies is more interesting. It has to do with the number and kinds of entities that exist in the data and the problems associated with tracing these

entities through time. I think that the number of entities is, like size and complexity, a good reason to worry about data management but not unique to panel studies. I want to discuss this issue first and then remark briefly on issue of tracing entities through time.

Some kinds of entities loom large in the design of nearly any survey: PSU's, housing units, households, persons. Everyone worries about them to some extent, with PSU's and housing units probably getting less attention in database issues than is warranted by their importance in the design. Using a relational system, I like to set up a separate relation, i.e., a separate rectangular data file, about each of these entities. In each file the "variables" are information about that kind of entity. Thus, number of rooms goes in the housing unit file, number of cars in the household file and years of school completed in the person file. Household number is also a variable in the person file, so you can aggregate over members of the same household and "join" information from the household file to that from the person record at will. A modern database will do that "join" operation in trivially longer than it takes to read the record. You can "join" aggregated person information to a household file or you can "join" household characteristics to all the appropriate records in a person file. Since a relational system isn't hierarchical, it would be possible for persons to belong to more than one household as well as for many people to belong to one household.

Other kinds of entities are sought-for in the schedule. For example, race and ethnic membership questions may be asked to identify membership in these entities. Members of some groups may be asked special questions not asked of others. Indeed, nearly every filter question involves the designation of an entity that may need to be recognized in the data management scheme. In PUMS files, I like to divide the person file into

two parts, one pertaining to the questions asked of all people and another for those questions asked only of persons over age 15. That saves a lot of space and helps keep track of the appropriate universe for variables. Of course, it is easy to "join" some or all of the questions asked of everyone to some or all of the questions asked adults.

Other sought-for entities are formed, not by common response to a question, but by relationships to other entities. Household relationship questions get at families, subfamilies, secondary families, cohabiting couples and the like. Since these questions essentially ask for "pointers" to other records, they want special consideration in designing a management scheme for any survey in which they appear. The traditional way to deal with these problems is through the rather restrictive convention of a household reference person. Household members' relation to that reference person is assayed and from the several responses the network of relationships among household members are unraveled and new entities such as subfamilies and secondary families are created. The query language for most relational systems makes such work relatively easy. In addition relational systems permit more flexible ways of indicating relationships between records. A field for the record number of the spouse or mother or father of the person is possible for one-to-one or many-to-one relations. For many-to-many relations, for example, siblings, there are a number of ways to proceed and the choice among them depends on the use one is likely to make of the sibling relationship.

Other sought-for entities are found through rosters in the schedule: autos, job transitions, residence changes, pregnancies, transfer payments, marriages. In relational systems these new entities each get their own file wherein variables are strictly about that specific entity.

The foregoing kinds of entities are explicit in the schedule. If you don't ask the questions you don't get the entities. Lots of other entities are implicit rather than explicit in the design and the schedule. They can be made explicit by aggregation or disaggregation, e.g.; birth cohorts or person-months. Birth cohort is an obvious aggregation on the person attribute, birth-year. There are a lot of aggregations like birth cohort, they are usually easy to accomplish even if the data is in pretty rough form, and we don't have to worry about them too much.

Married couple is also an aggregation on person records using the relationship questions to form the new entity. It can be a bit harder to produce. If your database isn't relational, you'd be better off knowing you want this entity from the start. Otherwise you are likely to have to write a good deal of code to process the files.

Person-month, at its simplest, involves creating multiple records for each person in a "wave" with the multiple depending on the months in the wave interval and then "stacking" the records over persons and then over waves. Attributes for such entities would be things happening to that person in that month. If you didn't know you wanted it before you structured the data, you need quite a good system to easily retrieve information about person-months.

All these entities can exist in cross-sectional as well as panel studies. They all argue for taking the data management issue seriously in general. They aren't special to panels.

The thing about entities that seems unique to panel studies is persistence. Can you match entities over waves? Persons and housing units seem conceptually straight forward although, as both David and Doyle point out, they may be operationally troublesome. Households and families are a mess to trace overtime. That's because the survey household is a

statistical convention invented for cross-sectional surveys. It is a kind of bag to hold all of the relationships and entities that exist among people sharing living quarters without having to specify each kind of relationship or entity individually.

The family concept we use in survey work is little different in character. So far as I know the idea was invented for the 1940 census as a way of specifying some of those relationships and entities collected together in the household. Where we used to assume members of a household shared expenses, now we assume members of a family do. So the survey family idea is a bag within a bag. It should really be called the cohabiting family to make clear that there is a larger and temporally more persistent entity out there that is different from this conventional idea. Since these conventional ideas of household and family are really convenient cross-sectional aggregations of entities and relations, it doesn't make much sense to try to follow them over time. That's to reify a construct beyond its useful extention. What is usefully traceable over time is the specific entity or relationship rather than the bag it was in. Thus, one might wish to trace couples or mothers and their children or people who share income. Tracing each of those specific entities wants some thought but has the virtue of specificity.

From the foregoing I conclude that the puzzle about the persistence of entities in panel analysis isn't something deep about the panel design but rather something shallow about the household and cohabiting family constructs.

2. SHOULD WE MAKE OUR OWN DATABASE SYSTEM?

Several years ago I visited the Institute for Social Research at Michigan and met Monica Blumenthal, a psychiatrist turned survey researcher doing wonderful studies of attitudes about violence. She remarked that social scientists are quite odd in their distrust of people with other specialty areas. If she had a patient with a heart problem, she said, she and most other physicians would call in a cardiologist. A social scientist would try to read up on the matter and do surgery himself. I thought of that remark as I read the papers which seem like proposals for the design of a database system for panel surveys.

There is a lot to know about designing database systems. It is one of the areas of concentration for the PhD in Computer Science at U.W. Madison. A good deal of research has accumulated. Some of it is quite theoretical but much of it is applied. The fruit of this work is lots of systems for machines ranging in size from pc's to Sierras. Nearly every major machine vender has several. The older CODASYL, or network, systems are being phased out.IMS, IBM's standby for years, for example, is officially on the way out. In these older systems you must pre-specify all the "joins" you want to make. If you think of something new you want to do it is trouble. Setting these systems up is a big deal and changing them is heavy work. It sounds to me as though the directly structured file system for PSID is of this sort.

Various implementations of the relational model appear to be the coming thing. Among other virtues they support ad hoc queries. You don't have to know all the questions you are going to ask of the data, all the "joins" you will want to do, before you begin. IBM has two forms, SQL/DS and DB II. DEC has added its own relational system, RDB, to its longstanding CODASYL system. There are several systems not associated with

manufacturers, Ingres, Oracle, ADABAS. Of course systems for micro's abound. DBaseIII is spoken well of by the pros I know.

Many of the available, and affordable software systems are powerful and fancy. Good ones have query optimizers. These parts of the system analyze what you want done in terms of what's known about your specific database and which resources are scarce on your machine. They figure out the optimum way of satisfying your query given the constraints and environment. They are not perfect. Indeed, a lot of research is currently underway to try to make them better. But, as they stand, they do a better job of deciding than many database managers can achieve and they do it very rapidly.

Two specific reasons are given for not wanting to use one of the currently available systems. One reason seems to be feeling that relational systems are just for businesses that process a lot of transactions. Many implementations of the relational model work hard to make transaction processing go quickly. Other implementations optimize other things. The Omega system designed for the Crystal Project optimizes the speed of the "join" and would be fine for our sort of work. The point is that it is not the relational model which is "for" transaction processing but rather the implementation. In a number of systems you can tailor the implementation to make it work better for your specific kind of problem. You can choose the storage structure of the data and you can chose to create indexes. You can choose how much effort you wish to spend collecting "marginals" on the data for use by the optimizer. You can also choose to use or not use features. For example, the journaling David mentions in Ingres is quite resource intensive and I avoid its use whenever possible. There is also great security control in many relational systems. For example, you can permit a given user access to only specific relations

at specific times of the day from specific terminals

Certainly the current package database systems aren't perfect. But they have developed over a number of years of intensive work by first-class computer scientists. Even when they are less than optimum, they are awfully good compared to doing it yourself.

A second reason mentioned in the papers for wanting to do it yourself is the issue of size and cost. I don't understand this point. Let's take size first. The 16 waves of PSID that we have a Madison occupy about a half gigabyte of space on tape. I've just ordered an additional half gigabyte of disk storage for one of our Vaxes. It will cost about $10,000. The PSID would probably take up about half as much space if stored in a relational system, partly because of reduction in "padding" and partly through using integer format. But suppose it required a whole gigabyte to work effectively. That would be about $20,000 for the drives, another $2,000 for cabinets and say an additional $1,000 for installation. Add about $10,000 for the initial license for a database system at the University price. So far you have spent $33,000. The opportunity cost for that money is a middle level programer for about half a person year. You don't get very far on a complex problem in half a person-year. Even if you had to spend another $30,000 on a Microvax or a Sun to get more cycles, I suspect you would still be miles ahead.

Once machines were expensive and people comparatively cheap. Now machines are cheap compared to people. It used to be worth spending time writing super-efficient programs. The price of machines continues to fall. Disk storage especially seems likely to continue its dramatic decline. So an investment in efficiency has to pay off pretty quickly.

3. WHOSE RESPONSIBILITY IS IT TO DO WHAT TO THE DATA

This is the most implicit issue in these papers. The authors make little distinction between the responsibility of the analyst and that of the collector/disseminator. All the authors are really analysts at heart. As such, they are pretty sure they know what information "in" the data is important and pretty sure they know how to get it out. That's natural. Often analysts have invented the project and participated in "selling" it to others in order to get the money to do it. They know that their idea of what's important is widely held. They know that the design and content are focused on those issues. Thus it isn't surprising that analysts want to structure the data in ways to make answering "their" questions most efficient. For their own work, that is exactly what they should do.

People in the data-providing role, I think, would be wise to remain a bit more agnostic about what will be the important information in a given survey. They will know from experience that when the first round of questions are answered, there will be a second round. What's in round two depends on what was found out in round one and the most important round two question is likely to come from the round one answers which were most surprising; i.e., the least predictable.

There are other reasons for providers to stay agnostic about what's important. People use your data for most inventive things. Indeed, earlier in this volume Baylor proposes that an important use of SIPP could and should be a great deal more information about how the survey machine actually works.

How should an agnosticism about what's important influence the provider's solutions to data management issues? I think the first thing it says is that the provider needs to work quite hard to provide <u>all</u> the information available in the survey. That often means information about the place of an entity in the design and also about the things that go

wrong. Perhaps one reason there is less research on some methodological issues than would be desirable is that the necessary information isn't really on the publicly released file but in the provider's filing cabinets. The second thing an agnosticism about importance implies is that the data provider needs to be able to restructure the data set as new questions arise and as new entities become important. The relational model is the only one that provides that facility without a major re-tooling.

4. LET ME SUMMARIZE.

First, I think data management problems are acute for all parts of social research, not just for analysis of panel data. I also think the software and hardware environment is changing to provide helpful answers to our problems.

Second, I think custom programming or custom design of data management systems is likely not to be cost-effective except for the very largest agencies.

Third, I think analysts should do whatever it takes to the data to get good answers to the questions burning in their hearts and minds. But providers must conserve all the information in a survey and keep that information in a way that allows it to be re-formed facilely to answer all the questions it can address.

Mailing Address:

Center for Demography and Ecology
University of Wisconsin
1180 Observatory Drive
Madison, Wisconsin 53706-1393
U.S.A.