Teaching Database Concepts

Don McNeil - Sydney, Australia

1. Introduction

Statistics has been traditionally concerned with the development of methods for the analysis of data, and statistical packages in current use reflect this objective. In a broader definition of statistics the objective is information analysis, where the information includes data structure as well as data distribution. Database systems - rather than statistical packages - are designed to handle data structure information. Thus a knowledge of database concepts is essential to any student of statistics who desires a broad education. At Macquarie University database methods are taught as part of a second-year statistics course. Our approach to teaching this topic is described, and some implications for the future undergraduate curriculum in statistics are discussed.

2. Information and data

Consider a broad definition of the discipline of statistics: "Statistics is the discipline that is concerned with the analysis of information in data". This definition embraces parts of other disciplines, particularly computer science. But statistics is the discipline that is *most* concerned with the analysis of information in data, and this object is *more central* to statistics than it is to other disciplines.

Let us examine some consequences of the definition.

What do we mean by the information in a set of data? There is information in the data themselves - their frequency distribution, tendency to separate into different groups, or to fit defined models containing specified parameters. The analysis of data information of this kind is the basis for traditional statistical theory and methods, including regression, categorical data analysis, analysis of variance, factor analysis, and other multivariate methods. Much of this is concerned with statistical inference, which involves making statements about the populations using the data as samples.

Data information is not the only kind of information of interest to statisticians. Other information includes assumptions (such as independence, additivity, linearity,

homoskedasticity, normality) about the population from which the data are sampled. If the data have arisen from an experiment, the experimental design (e.g. matched or non-matched experimental units) is informative. There may be further information in associated data (such as predictor variables in regression or other determinants) that are not subject to sampling error. Non-data information could even include information about the subject matter which may be relevant. For example, certain models may be excluded because they do not correspond to interpretable results. Information of this kind may be called structural information.

In a narrow statistical view, the data are the pieces of information collected in the course of a scientific experiment which are presented to the statistician for analysis towards the end of the investigation.

A broader view, argued by Marquardt (1987), involves the statistician in the additional roles of study design and data management as well as statistical data analysis. The statistician then becomes interested in structural information as well as data information. In traditional statistics teaching, structural information concepts are presented in courses in statistical design and statistical theory, largely ignoring the practical implementation of these concepts. It is perhaps for this reason that experimental design is usually only taught towards the end of the university statistics programme. The practical implementation of these concepts, which is necessarily based on data management and data retrieval prior to formal statistical analysis, involves the use of a computer to set up a statistical database. Landwehr (1985) also proposed the use of computer simulation to introduce factorial designs in elementary statistics courses.

3. Limitations of statistical packages

Statistical packages are the tools used by statisticians to process data information. They use procedures which take data arrays as inputs and create outputs in the form of graphs, tables, and other data arrays. These procedures have qualifiers which control the type of analysis and the type of output.

Statistical packages can also deal with structural information, but require this information to be programmed in the commands which specify the procedures and their qualifiers. Thus, for example, the distributional assumption of normality is specified by using a procedure that requires this assumption. Design information may be included in a similar way through qualifiers specifying the stratification variables in the data set.

There is a major disadvantage in the way statistical packages handle structural information. Since the structural information is not included as such with the data, there is no guarantee that the statistical analysis will be appropriate for the given data. One solution to this problem is to store the data in an information system, or database, which includes both data information and structural information. Besides reducing the likelihood of an inappropriate statistical analysis, this approach has a further benefit: problems of data management such as inconsistencies, errors, non-adherence to study protocols, and incomplete data, can be overcome prior to statistical analysis.

Although routinely included in computer science undergraduate programmes at universities, database concepts have not been considered a particularly important part of the statistician's education, even by persons arguing for more attention to computing in statistics. They are not mentioned at all in a recent paper by Makuch, Hahn and Tucker

(1990) which proposes a detailed syllabus for a dual degree at graduate level in statistics and computer science aimed to meet industrial needs. However, there are some compelling reasons for including database concepts in *every* modern statistics programme.

First, concepts in database appeal to students who are more interested in the practical applications of statistics. There are greater opportunities in the workforce for graduates who have these skills than there are for graduates whose skills are narrowly theoretical. As it is, many statistics graduates find it necessary to learn database concepts after they have graduated. Second, although many textbooks on database systems develop the subject in a way that is strongly linked with computer architecture or a computer language, database methods do not require prerequisites in computing. Perhaps most important, database concepts offer an opportunity for the development of effective statistical expert systems, and thus provide an important avenue for research.

Database concepts are taught as part (about one-third) of a second year undergraduate course in statistics at Macquarie University, with the other two-thirds covering topics in computer simulation. The prerequisites are elementary courses in statistics and computing which themselves require no prerequisites and are thus open to all students.

4. Teaching database concepts to statistics students

Date (1986) gives the following reasons for using a database:

(i) redundancy can be reduced;
(ii) inconsistency is avoided;
(iii) data are shareable;
(iv) standards can be enforced;
(v) security restrictions can be applied;
(vi) integrity can be maintained;
(vii) conflicting requirements can be balanced.

Database techniques are primarily concerned with the definition of models which contain structural information. The relational database model comprises, (a) a set of rectangular array structures (tables) with uniquely identifiable rows (records) whose columns (fields) have certain attributes, and (b) procedures for modifying and manipulating the data, including the retrieval of subsets of data satisfying specified conditions.

Well-chosen examples are needed to introduce these ideas to students whose computing background is minimal. In their first lecture the students are requested to write their name on a piece of paper together with, (a) the names of any students also doing the course whom they already know, and (b) some of their "likes" and "dislikes". The information thus obtained demonstrates the need for setting up a relational database, since it is not obvious how to structure the information with a view to answering simple queries such as "Who knows Jim?" and "Who doesn't like cats?".

A pictorial representation of the information about who the students know is shown in Figure 1.

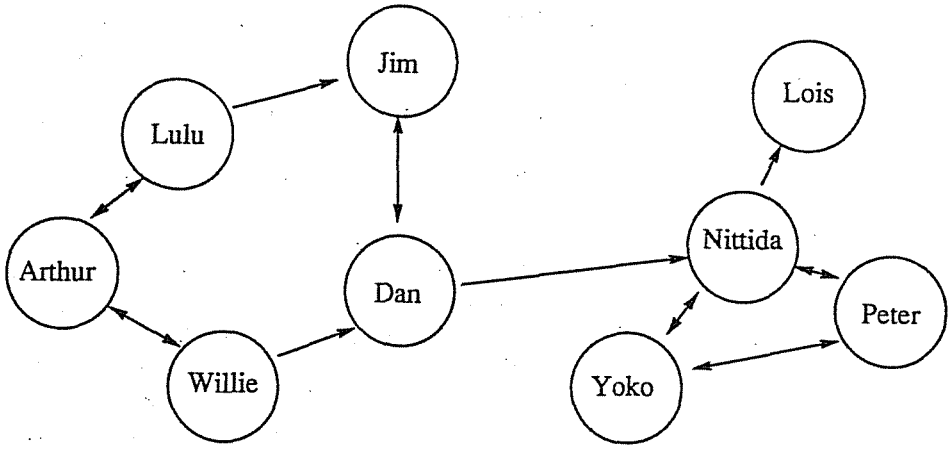

FIGURE 1
Graph showing who knows who

It is clear that the type of graph illustrated in Figure 1 is not a very useful device for storing the data in a computer with a view to further queries and statistical analysis. A table recording the students and who they know, such as that shown below, provides a better basis.

Jim - Dan
Lulu - Jim, Arthur
Lois
Arthur - Lulu, Willie
Dan - Jim, Nittida
Nittida - Lois, Yoko, Peter
Willie - Arthur, Dan
Yoko - Nittida, Peter
Peter - Nittida, Yoko

These data may now be stored in a rectangular array with four columns suitable for analysis using a statistical package (with missing values in records corresponding to persons who know fewer than three other persons). However, retrievals such as "Who knows Jim?" are still not easily answered, and data modification is likely to give rise to inconsistencies. Some examples: (1) a spelling correction to a person's name needs to be done for every occurrence in the table; (2) confusion will result if there are two persons with the same name; (3) if a person who already knows three persons makes a new acquaintance, a new variable will need to be created; (4) when new persons are added, it is necessary to check that all the persons they know are also included as separate records. Thus, many of Date's database requirements are not satisfied here.

The next step involves seeing how database concepts may be used to build a database for which data modification and data retrieval are straightforward. By introducing the concept of a *primary key*, the redundancy of the repeated names is avoided, and the possible confusion of having two persons with the same name is also eliminated. Table 1 shows a representation of the data table incorporating a primary key