Chapter XXXIII
Bit-Wise Coding for Exploratory Analysis

Thomas O'Daniel
Monash University, Malaysia Campus, Malaysia

INTRODUCTION

Getting data to yield their insights can be hard work. This is especially true with survey data, which tends to be oriented toward the presence or absence of characteristics, or attitude relative to some arbitrary midpoint. A good example of the first comes from surveying the Web by looking at Web sites: Ho (1997) analysed the value-added features used by 1,800 commercial Web sites to form a profile of commercial use of the Web within various industries; the U.S. Federal Trade Commission (2000) examined the characteristics of the privacy policy on 426 Web sites; West (2004) looked at 1,935 national government Web sites for 198 nations around the world, evaluating the presence of various features dealing with information availability, service delivery, and public access. A good example of the second comes from any number of studies that use the "Likert scale."

These studies are characterised by (a) large sample sizes and (b) an analysis that must incorporate a large number of "indicator" and "categorical" variables. Coding and analysis should be considered at design time: not only "what information to collect" but also "how to store it" and "how to use it." This is particularly true with Web-based surveys using HTML forms, since the data can be stored automatically without the intermediate step of transcribing the answers from paper questionnaires into the computer. Getting a relevant graphical view of the data is often essential, since the human eye is a powerful analytical tool. The help files that come with statistical analysis applications explain particular techniques, but the importance of coding is often obscured by the description.

Most multivariate statistical methods are built on the foundation of linear transformations: a weighted combination of scores where each score is first multiplied by a constant and then the products are summed. This combines a number of scores into a single score. For example, in multiple regression analysis, linear transformations are used to find weights for several independent or predictor variables such that the regression line expresses the best prediction of the dependent or criterion variable.

In surveys like the ones mentioned, it is easy to have a large number of variables that merely indicate the presence or absence of a trait. The goal of the technique presented here, "bit-wise coding," is description rather than prediction, using groups of these variables. The essence of the method is to code each variable as zero or one, multiply by a power of two, and sum the products to achieve a single score. The key difference between this and a linear transformation is that the constants

(weights) are assigned sequentially, so the resulting score uniquely represents a combination of attributes.

The real power of this method lies in the fact that no data is lost: the score represents a "profile" of each observation, and a frequency histogram represents the "popularity" of each profile within the sample. *A priori*, peaks offer insight into *multicollinearity*, which occurs when two variables are highly correlated, or if one variable has a high multiple correlation with others. Many analysis techniques are sensitive to multicollinearity, and it is often not immediately apparent when many variables are involved. Valleys and missing values represent combinations that are infrequently (or never) present. Outliers may be the result of (a) observations that are truly not representative of the general population or (b) undersampling of actual group(s) in the population. In both cases, they will distort the analysis, so outliers must be examined and the observations deleted (cautiously) if deemed unrepresentative–at the expense of sample size.

In this entry, we will look at the essence of the technique using a group of simple "yes/no" categorical variables, move on to using dummy-coded variables to represent several states, and finally, discuss appropriate methods for using these scores. It is assumed that we are using some statistical software that represents the data spreadsheet style: each column as a variable and each row an observation. The definition section provides a short list of multivariate techniques and the appropriate types of data.

THE BASIC TECHNIQUE

To begin, assume the following variables have been used in a study similar to the ones mentioned, and that they have been coded as (one = has) (zero = does not have) the characteristic. Each variable is assigned a power of 2 as its weight, the weight is multiplied by zero or one, and added to the total.

16 Privacy Policy
8 Advertisements for Other Companies
4 Return Policy
2 Online Ordering
1 Online Payment

Thus, a final value of (say) 20 means that this Web site has a privacy policy and a return policy (16+4), but no other features; furthermore, *no other combination of features will yield a score of 20*. A site with all of these features will have a score of 31, and a site with none will have a score of zero. It is also important to note that since every score represents a unique combination of features, the arithmetic mean is meaningless: a score of 3 means the site has Online Ordering and Online Payment, which does not represent the midpoint between a site that only has Online Ordering (2) and a site that only has a Return Policy (4).

This technique can be called "bit-wise" because it is the technique used by the computer internally to deal with groups of binary digits (bits). The powers of two are assigned sequentially, so the order of the bits we are combining becomes important. Unfortunately, bit ordering has long been a subject of controversy in the computer world. The "Little-Endians" want binary numbers to look like positive integers, where the leftmost digit is the most significant: (001 < 010 < 100). The "Big-Endians" want to lay out the bits like the x-axis of a traditional two-dimensional graph, with the origin (left) as zero, thus (100 < 010 < 001). This is the sort of thing that gets people (even computer scientists) annoyed and possibly confused. For a rather detailed but still amusing explanation, see Cohen (1981).

The happy reality is that it does not matter as long as the order used is consistent. These examples have the "low" bits on the right and the "high" bits on the left: thus, 10100 would be the binary representation of 20 in this example.