CODING AND ENTRY OF FOOD INTAKES

Lenore Arab, Ph. D.
Klinisches Institute für Herzinfarktforschung
der Medizinischen Universityätsklinik
Bergheimer Str. 58
D-6900 Heidelberg, FRG

Those more than 250 of you attending this pre-conference either use or wish to use nutrient data bases to convert food intakes or recipes into nutrients. Such usage requires a nutrient composition source, access to a computer and also a way of communicating with the computer. That is, a way of telling her which foods you are interested in. This presentation focuses on this communication problem--how to code and enter foods for analysis with a computerized nutrient data base. There is no single way of doing this; the theory behind this, various approaches to coding and entry, and the considerations and complications which guide in the selection of a method will be presented. Finally, how these tasks are performed in Heidelberg will be discussed.

How did I become involved? I am working in the field of nutritional epidemiology in the FRG and conduct nutrition and health surveys quite like a German HANES--but with different emphasis, and different dietary methodology (1)--we take great pains to collect seven-day dietary records from our subjects--which requires entry and coding of massive amounts of dietary information, and is the most labor intensive part of our clinical examinations.

Coding is always an expensive, labor intensive part of nutriental analysis, and a large potential source of error, but also as well, a critically important part of nutrient analyses--and too often neglected. Coding can be simplified, but not eliminated, by modern technology. It is a process, and not a single step.

Coding is the conversion of the information on items eaten or to be eaten into machine readable form.

Data entering is forcing the machine to read the coded information. Together they achieve what is being attempted in Figure 1; communication between subject and computer about what has been eaten.

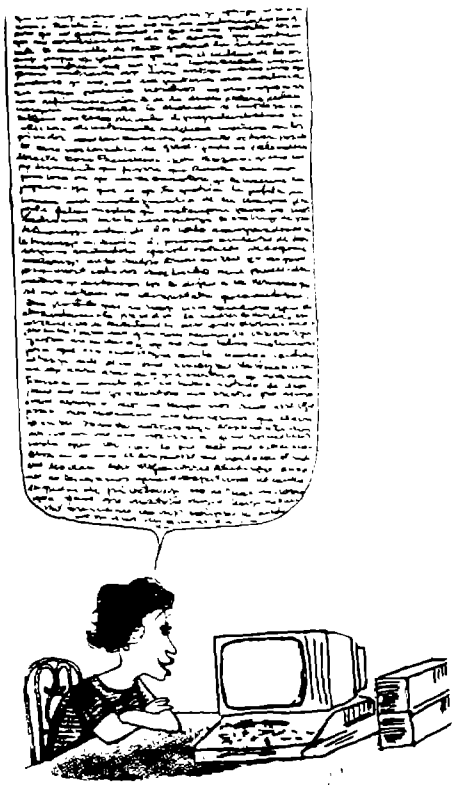


Figure 1.

Let's take a look at the steps involved in the calculation of a 24 hour recall to get a feel for what coding and entry entail. The steps with the orientation on the personnel normally required in this process include someone to interview; that is ask the subject exactly what and in which quantity was consumed yesterday; someone to look up and transcribe the codes for these foods and determine their amounts; someone to punch this information. This information must be fed into the machine by someone (leading to possible reading and typing errors); then the information in the computer should be checked for correctness against the originals; and the inevitable mistakes in such a system corrected, that is rewritten, typed, entered and checked. These steps and the dangers associated with each threatening the accuracy of the data are detailed in Table 1.

TABLE 1

STEPS INVOLVED IN 24 HOUR DIETARY RECALL ASSESSMENT

Steps	Potential Error Sources
1. Question person about what was eaten	Memory failure Influence of interviewer
2. Question about how much was eaten	Memory failure Estimation difficulties Non-edible portion (steak-bones, peach-pits) quantification
3. Code foods into numbers	Small code grouping of items with loss of information Reading errors Writing errors
4. Portion size conversion to gram amounts	Plate waste Non-standard portions Refuse deduction Portion calculations from standard recipe books
5. Entry of subject identification, date, meal, foods and amounts	Transcription errors
6. Checking entered data for correctness	Oversights Difficult forms No printout of food names
7. Correction of errors	Renewed typing errors
8. Recheck entered corrections	Oversight or elimination altogether
9. Calculation	Non-standard, non-debugged programs

That is an overview of the work involved. Now let's concentrate on the code itself.

APPROACHES TO CODING

The code is some specific and unique identification for each substance in the data base. The code may or may not contain information intrinsically and this choice of an informational or non-informational code helps determine the desired approach to coding.

The code itself may be numerical, alphabetical or mixed. One may have direct access without deliberate coding, that is, no need to enter a code and one may have no code at all. These options are illustrated in Table 2, along with some examples.

TABLE 2

APPROACHES TO CODING

1. Numerical code	123 18607258345
2. Alphabetical code	hamburger zygrhrjztsjr
3. Mixed code	b72
4. Direct access	touch it point to it
5. No code	lists pre-programmed questionning

Numeric codes can be as elegant and simple as 1-2-3, as in the first example.

They may, on the other hand, while trying to pack too much information into the code number (which is seldom necessary) result in a grossly awkward code along with proportionally more errors in transcription.

Alphabetical codes may apply easy-to-remember codes, such as simple food names or acronymes, or more cumbersome ones which for some reason can come out as with the second example; incorporating varied information such as that this is a mixed food, eaten in a restaurant, grilled, from frozen meat, using ground round, top choice pure beef, and certain spices and so on. This example is extreme, but I have seen some difficult to recognize and remember computer generated alphabetic codes. And I can imagine that mistakes are more frequent with such alphabetic codes than number codes because letters are not as "neutral"; one wants to associate foods with particular letters or sequences. Alphabetical codes of equal character length with numeric codes do require less storage space since 26 possibilities, rather than 10, are packed into each 2 bytes. An important drawback of alphabetical codes is that the coder must be able to type--that is to find the letters on the keyboard, rapidly and accurately; which is no minor consideration.

Mixed codes, a combination of letter and numbers, may improve coding--if, for example, the first letter is the food group (as in the example in Table 2, b for beverage, and then the number 72 for its specific identification. This can keep the numbers low, if designed sensibly.

Direct access incorporates newer technology such as touching the desired code on a screen, rather than searching for it in a book and transcribing it. Another approach to direct access, is, as we code in Heidelberg pointing to the desired food on the screen with an arrow--which will be described. The final option in Table 2 is no code--which sound very attractive; a system in which the coding is done automatically. This requires rather advanced technology such a pre-programmed 24 hour recall directly on the computer or list of foods in which the subject himself makes the match in searching for his appropriate choice. We have done the former with the 24 hour recall (2) and are quite pleased with the results.

NUMERIC CODING SYSTEM

Most people use numeric codes, and the numbering system can take various forms, as alluded to earlier. The code number may be random; absolutely random and totally non-informative. Or it can be an alphabetical code so that apple is the number one and zucchini is last (which is difficult to update), or sequentially numbered on the basis of entry into the data base, so that each new food in the data base--and they do keep expanding--gets the next number. With this one can easily tell when what came into the system, but one cannot easily find the codes.

The numbering system may be partially hierarchical; the first few digits for a food group, and then a sequential numbering within a food group for example, perhaps ordered after frequency of use. Another alternative, more difficult to develop, is a strictly hierarchical code in which every digit has a meaning, and the number of digits is determined by the required exactness of description; up to species, brand type of code with which we are operating. It is called the Bundeslebens-mittelschlussel (BLS). This is a "Federal Food Code", now contains approximately 5000 foods and 3000 mixed dishes, is 3 to 12 characters long, and strictly hierarchical in form, as illustrated in Table 3, along with an example (3).

TABLE 3

STRUCTURE OF THE BLS

Level		Example	
1-3	Basic food	2	Vegetable
4	Industrial processing	26	Root vegetable
5	Preparation method	263	Carrot
6,7	Modifications of basic foods	2636	Cooked carrot
8	Source, where purchased		
9	Packaging		
10-12	Country of origin		and so on.

The code is expandable and collapsable to the desired number of digits so that if one is not interested in positions 6-12, coding with 5 digits, as shown here, can be carried out. One may also drop earlier digits and code for example levels 1-3 and 12 as a 4 digit code (Rottka et al.).

A final approach to numeric codes incorporates "safety digits". The last digit of the code is related by complex mathematical equation to the first digits so that a one digit transcription mistake can be readily recognized, and an invalid code number results--the beginning and end do not match up properly. It is a method of error recognition but requires an increase in the number of digits in the code. In my opinion, simpler is better; as most people want to pack a maximal amount of information into a microcomputer; and other methods of entry without the risk of transcription error are available.

ALPHABETICAL CODING SYSTEMS

Another option is to use alphabetical codes. This presents problems, even if food names are used, as multiple names are possible for a single substance; such a frankfurter, hot dog, sausage, and bratwurst; and therefore duplicate listings are necessary, at the expense of storage space.

It is name descriptions in general which make the greatest demands on storage space. We allow 24 characters per name, which is like having a 24 digit code, or equivalent to sacrificing the storage of 12 nutrients with that food.

Also to be considered when using alphabetical name codes is the problem of syntax. Exact syntax is required by all small programs, and the acceptable syntax must be thoroughly learned by coders or multiple listings again incorporated, which can really tax the storage limits of the microsystems when you have a basic data base with a few thousand foods. Table 4 shows a few examples with milk.

TABLE 4

SYNTAX COMPLICATIONS

Milk, full fat
Full milk
Cow's milk, full fat
Full fat milk
Dairy products, milk...

Potential syntax problems can be tackled with a mixed system. The coder uses the basic, common food name or food group name and then chooses the specific or qualified foodstuff from a list of options.

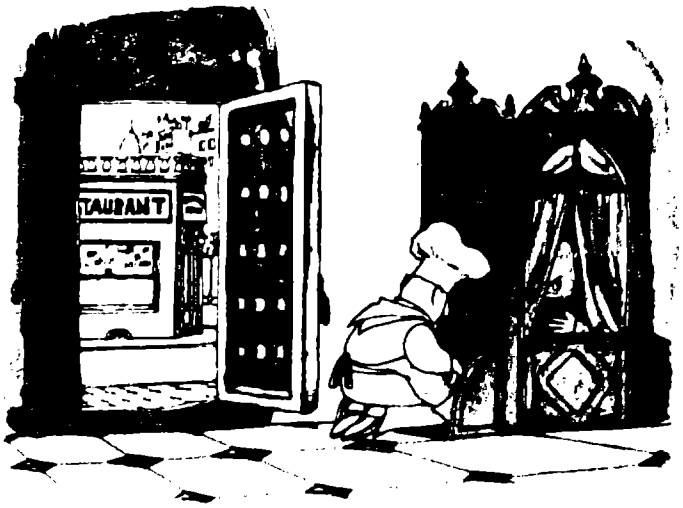
INVISIBLE CODING SYSTEMS

If not calculated by hand, all foods must be stored and identifiable through some system by the machine. There are methods in which the coder is not troubled with this process. For example, presentation by lists which can be scrolled up or down on a screen can be made to the coder. Another possibility is a logical questionning based on a decision tree with many possible options which automatically presents the foods. Or, a hierarchical code with level by level access to the desired food and preparation over food groups can be developed on the computers so that the coder need only point to the food group or specification of interest and thereby come closer and closer to the desired object. This is only programmable in a logically developed code and is the way in which coding is done at our institute (4). In all three methods the code number is invisible to the coders.

PROBLEMS IN CODING

The problems in coding mentioned include syntax variability, transcription errors, and multiple names for the same item. A greater problem is the use of the same name for two completely different items. An example of this, shared with me by Dr. Ken Samonds, is the recalling of Boston school children of eating multiple milk shakes each day. Closer questionning revealed that a candy bar named Milkshake was the real food consumed. One must be careful in the data collection stage to minimize such phenomena--one must be familiar with the population's behavior and its food choices. Fantasy names are also difficult to code--such as the name of one of my favorite Chinese dishes "ants climbing a tree"; or things like "sunshine salad".

Then there is the problem of mixed dishes and prepared dishes. Coding a simple food such as avocado is basically straightforward although the fat composition will differ between those from California and those from Florida--but it's the least of many problems. How does one code something like a salad? Numerous items and their proportions need to be identified and coded (unless your code has extensive mixed dish recipes), each item separately. Mixed dishes present other problems, especially if eaten away from home; one being that the consumer does not know exactly what he ate--only the chef does, as illustrated in Figure 2 (5).



Another problem is that of yield and retention factors. A cooked mixed dish, such as a casserole, cannot accurately be coded by its raw ingredients, that is not what is consumed. Yield and retention factors after cooking are relevant. An ideal code does this for you--has many standard recipes, brand names and options for coding various eaten forms of the basic food: peeled/unpeeled, cooked/raw, canned, dried, etc., so that a meal as in Table 5 can be coded with 6 codes rather than 26.

TABLE 5

SAMPLE MEAL WITH MIXED DISHES AS CODED WITH BLS

Amount	Food	Code
200 g	Greek salad	8301
1 slice	Mushroom pizza	8713
250 ml	Beer, Pils	0126
1 med.	Apple without peel	1182
2 scoops	Ice cream, vanilla	0345
3 Tbs.	Hot raspberries	1323

AMOUNT ESTIMATION

A huge problem which will not be elaborated on is that of amounts eaten. This is probably the greatest source of error in dietary assessment. Unfortunately amounts must also be entered, but the accuracy of estimated amounts is notoriously bad. Using ranges would probably be much more sensible. Measuring the recipe content of a meal is relatively simple compared to assess what people really consume. Depending upon the design and purpose of the exercise, different approaches can be taken. One can provide scales and ask people to weigh everything, but this is expensive, difficult for the subject, complicated to do properly in real life and probably has a great effect on what they eat. One can ask subjects to estimate amounts, which we have found to be unilaterally inaccurate. One can use portion size models--either realistic or abstract. Finally, many researchers depend on some normal serving size standards, which is presented as an option in our direct coding-on-computer system. When buying a nutrient analysis system it would be worth asking if household units and standard portion sizes are built in. What most people will want to enter into their data bases or calculations is person identification, meal food codes and amounts.

ENTERING PROCEDURES

Entry options for food codes and amounts include the old standby, rapidly fading away now, and for good reason, punch cards; optical readers using specially developed forms that need to be filled out specially in specific areas; and direct entry via computer terminal. In Heidelberg the latter is applied to these tasks.

As mentioned under the topic invisible codes, a computerized code book, based on a hierarchical code (the BLS) which requires no typing skills and no transcription of numbers in coding is used (4). Code and entry occur jointly with immediate checking and documentation. The program begins with asking the subject, date and meal--and if you don't know the code, one can opt to find the food by food group. Because the code is hierarchical one can find food within the 4-5 levels. 1) food group; 2) subgroup; 3) exact item; 4) how eaten; 5) how stored. The person at the computer need only use 4 keys on the keyboard to find a food. Those keys place an arrow up or down and to a higher or lower level of code. Then amounts are entered--next to the food name, and pressing return allows entry of the next food. A correction mode is possible at the end of any entry prior to storage or calculations. A final printout of foods and amounts in the same format as the original dietary records allows documentation of input and simplifies cross-checking.

Another approach which we apply is direct entry of foods, bypassing the coding stage, with the previously mentioned 24 hour recall on computer. In this system, the computer asks what was eaten for breakfast for example, and offers a choice of several food groups, selections here lead to a more detailed probing--and the desired description is stored by the computer without any steps of "pen to paper". This quantified information is sent to floppy disks and stored there until calculation or transfer to the data base. This questionning, coding, entry and storage occur simultaneously--eliminating many of the potential errors listed in Table 1.

CONCLUSION

Finally, to summarize, coding and entry are important issues in nutrient analyses, and there are numerous possible ways to do this. The best choice depends on the number of foods in the data base, the nutrients of interest, food groups of interest, ease of location of a food, ease and accuracy of entry, storage space constraints and access time.

Unfortunately, the best code for ease of use by coders is often not the most desirable to the programmer and may or may not meet the needs of research orientations. Coders want direct access, preferably no writing and little searching. If a code must be read and written, then it should be short and simple--acronym letters and lists for those who can type; numbers for those who can't. Programming is easiest and access fastest with hierarchical codes on a base 10 system. Researchers often want to be able to group foods, and therefore have some hierarchy in their code. They also want to know total amounts of basic foods consumed, and would therefore prefer not storing information as mixed dishes.* Priorities need to be set, and we set ours with a coder, who in the authorities opinion have the most tedious job, who are most responsible for data quality, and who spend the greatest amount of time with the task. An alternative option is to have two cross-referenced codes and a conversion program, or to use invisible codes with the type of direct entry systems we are applying.

My parting remark is that a standard food code available for use by all groups would improve comparability of results and facilitate recalculation of nutrient content as new data and new tapes arise. I believe it would prove to be well worth the effort.

*It is very difficult to add up the number of eggs used for example, when hidden in breads, noodles, casseroles, etc.

REFERENCES

1. Arab, L. Schellenberg, B., Schlierf, G. (1982): Nutrition and Health--a survey of young men and women in Heidelberg. In: Ann. Nutri. Metabl 25/S1/82. Karger, Basel

2. Arab, L., Bellin, O., Schlierf, G. (1983): Ein standardisiertes System für das 24-Stunden-Ernährungs und Activityätsprotokoll, das die Befrager-Variabilität ausschließt. XX. Wiss. DGE-Kongreß Gießen. Ernährungsumschau 30/7, S. 249.

3. Rotte, H., Arab, L., Polensky, W. (1983): Bundeslebensmittelschlüssel (BLS) für Verzehrserhebungen. XX. Wiss. DGE-Kongreß, Gießen. Ernährungsumschau 30/7, S. 250.

4. Arab, L., Pfannendörfer, H., Schlierf, G. (1983): Lebensmittel codieren ohne Nummern--ein Computer programm zur direkten Dateneingabe. XX. Wiss. DGE-Kongreß, Gießen. Ernährungsumschau 30/7, S. 249.

5. Quino: Cartoons. Deutscher Taschenbuch Verlag, München, 2. Aufl. 1980.