

PN-ACH-800

# **DESIGNING A DATA ENTRY AND VERIFICATION SYSTEM**

**Peter A. Tatian**

**MICROCOMPUTERS IN POLICY RESEARCH 1**

**INTERNATIONAL FOOD POLICY RESEARCH INSTITUTE**

1

Copyright 1992 International Food Policy  
Research Institute

All rights reserved. Sections of this report may be reproduced  
without the express permission of but with acknowledgment to the  
International Food Policy Research Institute.

Library of Congress Cataloging-in-Publication Data

Tatian, Peter A.

Designing a data entry and verification system / Peter A. Tatian.

p. cm. -- (Methods and applications for using microcomputers  
in public policy research ; 1)

Includes bibliographical references.

ISBN 0-89629-324-6

1. Agriculture--Economic aspects--Developing countries--Data  
processing--Planning. 2. Agriculture--Economic aspects--Niger--Data  
processing--Planning--Case studies. 3. Agriculture--Economic  
aspects--Senegal--Data processing--Planning--Case studies.  
4. International Food Policy Research Institute. I. Title.

II. Series.

HD1417.T38 1992  
338.1'091724'0285--dc20

92-15309  
CIP

C  
11

# CONTENTS

Foreword

Preface

1. Introduction .....	1
2. Questionnaire Design .....	7
3. Data Files .....	15
4. Data Entry .....	26
5. Data Cleaning .....	36
6. Using Previously Collected Data .....	49
7. Data File Management .....	52
Appendix: Software Considerations .....	57
Glossary .....	62
Bibliography .....	65

# FIGURES

1.	Crop purchase questionnaire . . . . .	9
2.	Sample codebook entry . . . . .	11
3.	Sample rectangular file . . . . .	15
4.	Method 1 of organizing census data files . . . . .	17
5.	Method 2 of organizing census data files . . . . .	19
6.	Method 3 of organizing census data files . . . . .	20
7.	Sample schematic of a database . . . . .	21
8.	Sample data files for census questionnaire . . . . .	23
9.	Results of first match of census data files . . . . .	24
10.	Results of aggregating FILE_B.SYS to calculate average age . . . . .	25
11.	Results of second match of census data files . . . . .	25
12.	Flow chart of data entry and verification system . . . . .	27
13.	Batch and questionnaire numbering . . . . .	28
14.	Data entry and verification batch control form . . . . .	29
15.	Logbook entry for recording the receipt of questionnaires . . . . .	31
16.	Questionnaire preparation form . . . . .	32
17.	Sample crop purchase data . . . . .	40
18.	Output from out-of-sequence record program . . . . .	41
19.	Sample crop purchase data . . . . .	42
20.	Output from double record program . . . . .	43
21.	Sample household census questionnaire data . . . . .	43
22.	Output from comparison of files on households and persons . . . . .	45
23.	Control form for supplementary verifications . . . . .	48
24.	Sample file of consumption quantities . . . . .	49
25.	Output from the program listing consumption unit measures (CONUNITS.LIS) . . . . .	50
26.	Directory tree structure . . . . .	53
27.	Data file documentation format . . . . .	54
28.	Summary of features for data entry packages . . . . .	58

# FOREWORD

The International Food Policy Research Institute (IFPRI) was established to identify and analyze alternative national and international strategies for meeting food needs in low-income countries, with particular emphasis on relatively poor groups within these countries. IFPRI's research on policy issues related to food production, food consumption, food marketing, and international trade is disseminated through IFPRI and other publications, through seminars and workshops held around the world, and through one-on-one interaction with developing-country collaborators and policymakers.

Important by-products of IFPRI research are the tools and techniques developed during the research process. These can be considered IFPRI's social science "germplasm" and the "seeds" for developing-country self-sufficiency in policy research and analysis.

This series, *Microcomputers in Policy Research*, aims to make IFPRI's tools and techniques available to individuals and institutions responsible for undertaking food policy research and analysis in the developing world. This series supplements IFPRI's ongoing efforts to contribute to the stock of human and physical capital available for policy research in developing countries, which IFPRI currently accomplishes through its collaborative research efforts, the supervision of developing-country Masters' and Ph.D. dissertations, and training workshops.

It is our hope that this series will provide, for the first time, documents that effectively communicate the "dos," "don'ts," and "how-tos" of policy research in ways that can be understood and implemented. We rely on our clients—you, the ultimate users—for guidance on how to improve the series, both in terms of content and presentation.

Just Faaland

# PREFACE

Over the past decade, the increasing power and reliability of microcomputers and the development of sophisticated software designed specifically for use with them has led to significant changes in the way that socioeconomic data are collected and analyzed. The venue of the computations has shifted from offsite mainframes, dependent on highly trained operators and significant capital investment in supporting equipment, to the desktop and even to the laptop, dependent only on the occasional availability of electricity. This means (1) it is now feasible to transfer quickly new statistical software to IFPRI collaborators for use in developing countries, (2) data manipulation costs of policy analysis have been substantially reduced, and (3) a new level of complexity and accuracy is now possible in the collection and analysis of household survey data in developing countries.

As with any new technology, however, there are substantial costs in time and money involved in learning the most efficient ways of using this new technology and then transmitting these lessons to others. This series on *Microcomputers in Policy Analysis* represents IFPRI's collective ongoing experience in adapting microcomputer technology for use in food policy analysis in developing countries. The papers in the series are primarily for the purpose of sharing these lessons with potential users in developing countries, although persons and institutions in developed countries may also find them useful. The series is designed to provide hands-on methods for resolving statistical and data collection problems encountered in food policy research. In our opinion, examples provide the best and clearest form of instruction, so examples will be used extensively throughout this series. Actual software code will be provided wherever relevant.

The first paper in the series, *Designing a Data Entry and Verification System*, by Peter Tatian, is a manual outlining how to manage and verify the collection of household survey data. It is based primarily on IFPRI/ICRISAT experiences in Niger and Senegal but draws on IFPRI experiences in other countries as well. Tatian addresses a number of issues relating to questionnaire design, the structure of data files, the entry and cleaning of data, and the management of data files. Illustrations are provided throughout, using SPSS/PC+ code and output.

Howarth Bouis, Lawrence Haddad, and Stephen Vosti  
Editors

## **ACKNOWLEDGMENTS**

This document is the culmination of three years of work that began in 1989 when I first traveled to West Africa to develop data entry systems for household surveys being carried out by the International Food Policy Research Institute (IFPRI). I am extremely grateful to IFPRI researchers Tom Reardon, Jane Hopkins, and Valerie Kelly for giving me the opportunity to try out my ideas in the field, using their research as "guinea pigs." I am also appreciative of the support, encouragement, and guidance they have given me in developing these materials.

I would like to express my gratitude to IFPRI for encouraging me in this work and for supporting its completion. I would also like to acknowledge the University of Laval, Quebec, Canada, for funding the initial development of many of the ideas on data entry and verification systems presented here.

Most importantly, I wish to thank the many colleagues who took the time to review and criticize this paper; it is a greatly improved work because of their valuable input. In particular, I would like to express my appreciation to Ruth Meinzen-Dick, Nancy Walczak, and Julie Witcover of IFPRI and Maris Mikelsons of the Urban Institute for having the tenacity to persevere through several iterations of this document. I am also grateful to Clemen Gonzales, Lisa McNeilly, and Dipa Nag-Chowdhury for their helpful comments.

# 1 INTRODUCTION

The past few years have witnessed a steady increase in the availability of personal computers (PCs) in developing countries. Although this increase presents many opportunities for improving both the quality and quantity of survey data collection, there have been few resources available to assist researchers in applying this new technology to field survey work. It is the author's hope that this document will help fill this gap by presenting some practical guidelines and techniques for using PCs for the entry and verification of field survey data.

The principal goal of this document is to provide a guide to researchers who use PCs for entering, verifying, and processing survey data—bridging the gap between survey research theory and practice and the technical issues involved in data entry and analysis. The ideas presented here are based on the author's experiences with data collection projects undertaken by the International Food Policy Research Institute (IFPRI) in Niger and Senegal in West Africa. These studies used a number of different data instruments administered on a continuous basis to a fixed sample of rural households over a period of two-to-three years. While many of the suggestions are directed toward this type of long-term, multicomponent study, the author has nevertheless tried to present some general principles that may be applied to a broad range of surveys and to a variety of data collection circumstances.

A key point that will be made throughout this document is that data entry and processing requirements must be considered at all stages of survey planning and implementation. Consequently, a variety of different issues relating to the collection of survey data will be addressed. The remainder of this chapter discusses issues confronted in incorporating data entry and processing requirements into survey planning. Chapter 2 considers various problems in questionnaire design and the impact these can have on data entry. Chapter 3 explains the structure of data files and illustrates how some file designs may be easier to use than others. (Although this paper is directed primarily to researchers who engage in data collection, readers who work with secondary data will also benefit from the information in Chapter 3.)

A complete data entry and verification system is presented in Chapters 4 and 5. Chapter 4 outlines procedures for organizing and entering questionnaires, while Chapter 5 describes various types of data verification. In Chapter 6, an example is given of how data entered early on in a survey can be used to guide later data collection. Finally, Chapter 7 discusses several issues relating to proper management and organization of data files.

Throughout this document, it is assumed that the reader has some previous computer experience and a basic understanding of computer concepts. At IFPRI, SPSS/PC+ and SPSS Data Entry<sup>1</sup> are widely used for data processing and analysis, and programming examples from these packages are used to illustrate the ideas presented. This is not meant to imply that these particular packages should be used, however, as the procedures presented in this document can be adapted to most of the database and data entry software currently available. A comparison of the different types of software suitable for data entry appears in the Appendix.

Before beginning, a warning about the use of computers in survey work is in order. It is natural for people to assume that computers will make the task of collecting data easier. Computers are indeed powerful tools that can perform many mechanical data checking tasks very rapidly and accurately, but this increase in power comes at a cost. As will be seen, computers often cause systems to become more complicated and place additional demands on the researcher's managerial abilities. The use of computers to verify data collection is not a panacea, and it certainly should not be considered a substitute for thorough training and supervision of survey personnel.

If computers are to be used in the field, they should be introduced in a systematic and incremental manner. *It is best to start off slowly, introducing more sophisticated and intricate steps only after the staff become comfortable with the basic procedures.* The level of experience and training of the staff should always be the most important factor in deciding how complex a system to implement. The destructive capacity of a computer in untrained hands should not be underestimated: months of work can be wiped out in seconds. To help prevent this, the researcher should make sure that each person using the computer is well trained in the PC's operating system and the software that he or she will be using.

It is best to have someone in the office (full-time if possible) who is thoroughly familiar with the data entry and analysis software and who will write programs and handle any data processing problems that arise. This will free the researcher from these responsibilities and allow more attention to be devoted to the field work. If no one is available locally with these skills, the researcher may want to bring in a programming "expert" to leave in place a system that can be operated by the office staff. There is a danger in doing this, however, in that the expert may create programs that are beyond the understanding of the permanent staff. Since all programs require maintenance and modifications, the researcher should be certain that any programs written by a consultant are not completely beyond anyone's ability to decipher and revise.

---

<sup>1</sup> Companies producing the computer software mentioned in this manual are listed in the notes to Figure 28, p. 58.

## **SURVEY PLANNING**

Because survey planning has important ramifications for data entry, it will be discussed briefly here. The first step in planning any survey is to articulate the basic questions to be answered by the research. For example, What is the importance of nonagricultural activities in the income of rural households? What constraints exist to expanding agricultural production? How is food security affected by cash cropping? From these general questions, the researcher forms a more detailed set of questions and hypothetical answers and then determines the types of data that are needed to validate these hypotheses.

It is this last issue that is often given too little thought. If the goals of a survey are poorly articulated, the result will be that important data elements may be omitted. Alternatively, some researchers attempt to collect data that will answer every conceivable question, causing resources to become overextended and resulting in data of poor quality. It is far better to collect fewer, more specialized data elements and be insured of the quality than to collect a large quantity of data of questionable validity.

In order to collect reliable, accurate, and comprehensive data, it is necessary to specify in advance the uses to which these data will be put. Variable definitions, recall periods, and collection methods all depend on the analysis to be done. The researcher must therefore try to spell out all desired uses for the data well before the survey starts. Summary tables should be laid out as they will appear in their final form. All calculations and models should be written out to be certain that the data being collected are adequate to carry out the required computations. In this way, missing data elements and flaws in the structure of data files will be exposed *before* data collection begins.

## **PRETESTING**

Most surveys include a *pretest* of questionnaires on a small subsample to verify their suitability and to develop lists of coded responses for each question. For example, a food consumption questionnaire pretest may provide a list of the different types of household unit measures used in meal preparation. It is strongly recommended that data from the pretest questionnaires be entered in order to check data entry and cleaning procedures. Doing this can reveal problems in file structures and data entry and cleaning procedures.

## **ESTIMATING DATA ENTRY REQUIREMENTS**

One of the keys to managing a successful survey is accurately estimating the time needed to perform various tasks. When budgeting time and resources, it is important to include the time needed to enter and process the data. If insufficient attention is paid to data entry requirements, the data may be underutilized or analyzed too late to be useful. As Casley and Lury (1987, 126-127) point out, the most common cause of survey failure is failure to plan the processing.

The following formula produces a rough estimate of the amount of time necessary for data entry:

$$PERSDAYS = \frac{NUMQUEST}{QUESTHR \times HRSDAY} ,$$

where

*PERSDAYS* = Total person-days required per month to enter data from a particular type of questionnaire,  
*NUMQUEST* = Number of questionnaires collected per month,  
*QUESTHR* = Average number of questionnaires one person can accurately enter per hour, and  
*HRSDAY* = Number of hours in a working day.

The formula computes the total number of person-days needed to enter all questionnaires of a given type collected during a one month period. This calculation should be done separately for each survey instrument because the number of questionnaires collected per month and the average number of questionnaires that can be entered per hour will be different for different types of questionnaires.<sup>2</sup>

For example, suppose that in a study of 300 households there is a crop purchase questionnaire that is administered every two weeks. Each month of data collection produces approximately 180 completed questionnaires. If a data entry operator is able to enter 12 crop purchase questionnaires per hour and if there are 8 hours in a working day, then calculating by the formula above, 1.9 person-days will be needed to enter all crop purchase questionnaires collected in a month (MO):

$$\frac{180 \text{ QUEST/MO}}{12 \text{ QUEST/PERSHRS} \times 8 \text{ HRSDAY}} = 1.9 \text{ PERSDAYS /MO.}$$

Similar formulas could also be developed to estimate time needed for questionnaire preparation, to run verification programs, to clean the data files, and other processing tasks.

## **CONCURRENT DATA ENTRY**

Using PCs in the field makes it possible to accelerate the reporting of survey results by performing data entry and processing tasks concurrently with data collection. Before PCs were widely available, one had to wait until the survey was completed before being able to enter and work with the data. Now, however, it is fairly common to find PCs in developing countries, and battery-powered portable computers can even be brought directly into the field. The approach of entering questionnaires while the data collection is proceeding will be referred to as *concurrent data entry*.

<sup>2</sup> This formula does not take into account the "start-up" time required to familiarize the data entry operators with the software and questionnaires. It should be recognized that data entry will proceed slowly at first, increasing in speed as the operators move further along the learning curve.

The key advantage of concurrent data entry is that feedback can be received on the quality of the data before the study is completed. Problems thus uncovered may be correctable during subsequent survey rounds. Entering the data immediately forces a careful examination of the completed questionnaires, a task that might otherwise be neglected but which can reveal many data problems. Special tests can be implemented with the computer to detect outlier observations and to check for inconsistencies in the data. For example, per hectare yields can be calculated from production data and compared with documented yields from outside sources. Finally, preliminary analysis may expose omitted data that might be impossible to obtain after the survey has ended.

Another advantage of concurrent data entry is that one can use previously collected data to guide subsequent data collection. An example of this technique is given in Chapter 6, which describes how consumption questionnaire data can be used to produce a list of household unit measures. This list can be brought into the field to indicate which unit measures need to be weighed for each household, greatly facilitating the collecting of these data.

In spite of the advantages, there are also some difficulties with the concurrent data entry approach. For one, this system demands more resources and supervision. It is necessary to oversee work being carried out both in the field and in the office, and often the researcher is overwhelmed simply trying to keep up with activities in the field. Managing both tasks requires careful organization and more thorough training of survey and data entry personnel.

In the IFPRI studies in Niger and Senegal, for instance, it became necessary to scale back the complexity of the data verifications because the level of experience and expertise of the office staff needed to be improved before more involved procedures could be implemented. It took almost the entire first year of the study for the staff to become habituated to the data entry system to the point where more complicated procedures could be introduced. There were relatively fewer difficulties in Senegal, however, because of the better training and computer literacy of the local staff, which illustrates the need to take into account the skills of the personnel available when designing data entry procedures.

Another practical problem the IFPRI data collection effort had to overcome was how to receive quick feedback from the field enumerators. Ideally, enumerators should respond rapidly to problems uncovered during data entry, but this capability was severely hampered by transportation and communication difficulties. The size and coverage of the survey sample were two additional factors that affected response time. Communication with enumerators was somewhat easier in Niger, for instance, where the six sample villages were close to the capital of Niamey. This was in contrast to the situation in Senegal, where the 30 sample villages were scattered throughout the country and where it often took one to two months to receive a response from the field regarding a data problem.

**SUMMARY**

Proper planning is essential to a successful survey effort, and that planning must include not only data collection but also data entry and processing. Beginning with the basic research questions, the researcher must design questionnaires that will produce appropriate data files for the analysis to be undertaken. Care should be taken to avoid collecting unnecessary data elements and overlooking crucial items. Pretesting of questionnaires and data entry procedures will help expose problems before real data collection begins.

In deciding whether to adopt the concurrent data entry approach, the researcher must balance the costs with the benefits. Concurrent data entry can improve the quality of the data through rapid detection of errors and can speed the reporting of results. Nevertheless, it takes time to get such a system working properly and to develop the necessary coordination of activities between the field and the survey office. Both the office and field staff may require extensive training to make the concurrent data entry system work properly. Large or geographically dispersed samples also pose problems for concurrent data entry, since communication between the survey office and field enumerators may be more difficult.

## 2 QUESTIONNAIRE DESIGN

There are many elements of questionnaire design that can have an effect on data entry. Unfortunately, researchers do not always give adequate consideration to data processing requirements when developing their questionnaires. As a result, when it comes time to enter the data, it is discovered that the questionnaire format is not appropriate for the types of data files that need to be created. If data entry requirements are considered when the questionnaires are being designed, fewer entry errors will be made and time will not be wasted correcting survey instrument design during data entry.

It would be best to define some terminology before continuing. Some studies may use several different survey instruments to collect different categories of data. For example, there might be one questionnaire for collecting data on purchases of crops made by households, another questionnaire for collecting information on meals consumed by households, and so forth. In other studies, however, there may be a single survey instrument designed to collect a wide variety of information.

Regardless of the amount or types of information collected on a survey instrument, in this document the term *questionnaire* will denote a data collection form designed to be administered as a single unit. An individual questionnaire may consist of one or several pages. In order to distinguish between a single copy of a questionnaire and the different kinds of survey instruments used in a survey, the term *questionnaire type* will be used to refer to the latter. A survey may therefore consist of one or several different questionnaire types.

### HEADERS

Most questionnaires can be divided into two parts: the *header* and the *body*. The header normally appears at the top of the page and contains identifying information that applies to the entire questionnaire. Most of the information identifying "who," "what," "where," and "when" will be included in the header: Who was interviewed? What was the interview about? Where and when did the interview take place? The body contains the bulk of the data being collected by the questionnaire. On more complicated questionnaires, the body may be divided into several subsections, each subsection having its own header.

A uniform header should be placed on all the survey questionnaires as this will facilitate both data collection and data entry. For multi-page questionnaires, the header information should be repeated at the top of every page so that it will be possible to identify the data if the pages become separated. All questionnaires

should have a place for recording the date of the interview, which may appear either in the header or body of the questionnaire depending on whether the questionnaire is designed for only one interview or for several interviews. A single interview questionnaire will have the interview date in the header, while a multi-interview questionnaire will have the date in the body.

Figure 1 shows a typical questionnaire from the IFPRI studies that illustrates these principles. The questionnaire was applied approximately every two weeks to the head of household, who was asked to recall all crop purchases or gifts of crops received since the previous interview. The crop purchase questionnaire header gives the questionnaire title and number and has spaces for recording the village and household identification numbers (IDs), the name of the household head, and the village name. Since the village and household IDs are the only header information to be entered, boxes are placed around them in order to help the data entry operator find these items more easily. The data concerning specific crop transactions are recorded in the body of the questionnaire. Since this questionnaire is used for multiple interviews, the interview date is included in the body (column 1). (If this were a single interview questionnaire, a box for the date would have been placed in the header.)

## CODES

While most data analysis software will accept data in nonnumeric or "string" form, in practice it is difficult to use this type of information. Text information is often imprecise—different people may spell or say things differently. For example, "kilo," "kilogram," and "kg." all have the same meaning, but they are quite dissimilar to the computer, which only equates two text strings if they are *identical*. Moreover, the imprecision of written responses makes it difficult to use text in data analysis. It is therefore preferable to have sets of numeric *codes* for all qualitative responses. Instead of entering the word "kilogram," for example, the numeric code 1 would be entered.

Numeric codes serve two purposes—they standardize responses to questions so that ambiguity and misinterpretation are reduced, and they simplify the manipulation and analysis of the data. It is important to be consistent when developing code sets. For instance, if millet is coded as 101 in the agricultural production questionnaire, the same code should be used in the crop purchase questionnaire. This will simplify the work of the enumerators and data entry operators, as well as making it easier to compare information from different types of questionnaires.

The questionnaire should provide spaces for entering the coded response for each item. If the number of possible responses for a given question is rather limited, such as Yes/No or Male/Female, the enumerator can code these answers directly on the questionnaire: 1=Yes and 2=No, for instance. For items that have a large number of responses, however, it is better to leave a space for a brief written description of the answer and an adjacent space for the numerical code.

Figure 1—Crop purchase questionnaire

QUESTIONNAIRE #07 -- CROP PURCHASES AND GIFTS RECEIVED

VILLAGE 1 HH ID 2 HH head \_\_\_\_\_ Village \_\_\_\_\_ Recall period: since last interview

Inter-view Date	Days Since Last Inter-view	Trans-action No.	Crop Received		F O R M	Pur-chase (=1) or Gift (=2)	Quantity	Unit		Cash Payments		In-kind Payments					Reason for Purchase		
			Name	Code				Name	Code	Unit Price	Total	Type	Code	Quan-tity	Unit	Code	Form	Description	Code
7-1-89	15	1	Rice	106	2	1	2.0	kilo	1	0	0	Millet	101	0.5	S.Sack	10	2	HH cons.	1
7-1-89	15	2	Peanuts	108	3	1	1.0	packet	25	75	75							HH cons.	1
7-16-89	15	1	None	900															
7-31-89	15	1	Rice	106	2	1	2.5	kilo	1	150	375							HH cons.	1
7-31-89	15	2	Cowpea	109	4	1	1.0	packet	25	M	M							HH cons.	1
7-31-89	15	3	Peanuts	108	3	2	1.0	packet	25										
8-12-89	12	1	Missing*	800															

\* 8-12-89: Respondent not found.

For example, on the crop purchase questionnaire (Figure 1) the fourth and fifth columns are used for the name of the crop received and its code. The enumerator can record the written response and leave the coding until later, thus avoiding wasting time by searching through code lists during the interview. Furthermore, although only the code is actually entered into the computer, having both the written and coded response on the questionnaire makes it possible to verify visually that the correct code was used.

Another option is to use check boxes to record responses. For example:

Do you plant rice?	<input type="checkbox"/> Yes (1)	<input type="checkbox"/> No (2)
--------------------	----------------------------------	---------------------------------

In this case, the enumerator simply checks the appropriate response to the question. Note that the numeric codes of 1 and 2 are included with the responses of "Yes" and "No." This will facilitate subsequent processing of the questionnaires as no coding of these responses will have to be done prior to data entry.

In most surveys, code lists are developed during the pretesting of questionnaires. Even with thorough pretesting, however, it may become necessary to expand or change the code lists during the course of the survey. Some responses may have been overlooked in the original code set, or perhaps some responses would be better combined or eliminated. In fact, one of the advantages of concurrent data entry is that it can expose inadequacies in the coding scheme. *It is important to exercise caution when modifying code sets, however, as the potential confusion can be disastrous.*

Changes to code sets should only be made *between* survey rounds, not during them, and the decision to make coding changes should be made exclusively by the researcher. Enumerators should never be allowed to create new codes in the field as these codes would not be standardized across the survey. Changes in the code list should be clearly documented and distributed to all enumerators simultaneously along with precise instructions as to when the changes take effect (for example, "starting with round 7," or "as of 1 September"). If an enumerator thinks the current set of responses for a question is inadequate, he or she should note the response in writing on the questionnaire but leave the code blank.

Generally speaking, adding new codes does not pose a serious problem. If the researcher anticipates many additions to a code list during the survey, the codes could initially be numbered using odd numbers, leaving the even numbers for the insertion of new responses. This can be useful if, say, the responses are to be coded in alphabetical order—new responses could then be added without disrupting the order.

Changing the definitions of existing codes should be avoided if at all possible because of the problems that can result. *An existing code should never be given a new meaning, but rather a new code number should be assigned to each new response.* This is important not only to avoid confusion for the enumerators and data entry

operators, but also to prevent subsequent data analysis difficulties. Although information from different survey rounds may initially be kept in separate data files, these files will most likely need to be combined for analysis. Once merged together, it will become difficult if not impossible to distinguish between different meanings for the same numeric code.

The importance of not changing the meaning of codes during the survey is well illustrated by the following example. In the IFPRI studies, a unique identification number was assigned to each household in a village. A household's ID was not changed from one interview to the next, even though households were added and dropped throughout the survey. If, for instance, a household was removed from the sample, no other household was assigned its ID number. If, on the other hand, a new household was added to the sample, it was given an ID number that had not been previously used by another household. In this way, individual households could be consistently tracked throughout the entire survey.

A list of codes along with copies of all the questionnaires should be kept in a *codebook*, which is updated as needed. A sample codebook entry for the variable REASON from the crop purchase questionnaire is shown in Figure 2. This variable indicates the reason for a household's crop purchase. In the codebook, the name

Figure 2—Sample codebook entry

Codes for Crop Purchase Questionnaire (#07)			
VARIABLE	DESCRIPTION AND CODES		
REASON	Reason for crop purchase		
	<u>1988</u>	<u>1989-90</u>	
	1	1	Home consumption
	2	-	To give as gift
	3	3	To sell (commercial transaction)
	4	4	Other reason
	-	5	Gift to family member
	-	6	Gift to nonfamily member in village
	-	7	Gift to nonfamily member outside of village

of the variable and its description is given along with a complete list of coded responses. Note that there were two different code sets used for this variable—one for 1988 and one for 1989-90. For 1989-90, reason code 2 ("To give as a gift") was discontinued and codes 5, 6, and 7 were added to differentiate between gifts to family and nonfamily members. Because new codes were created and code 2 was *not* redefined, it is possible to combine data from 1988 with data from 1989-90 without any confusion of code meanings and without any recoding of responses.<sup>3</sup>

### **MISSING AND ZERO RESPONSES**

An important issue in designing code sets involves distinguishing between missing and zero (0) responses. The *missing response* signifies a complete absence of information (a nonresponse), while the *zero response* indicates that it is known that a particular activity did not take place. These two responses represent completely different situations, and it is important that the coding system be able to differentiate between them. Situations where an entire interview is zero or missing, as well as those where a specific data element is zero or missing, also need to be accommodated.

Being able to distinguish between missing and zero responses is crucial for proper data analysis. If a zero response was entered for an observation, then this can be interpreted as a true zero in computations (when calculating averages, for instance). It is incorrect to interpret a missing response as a zero, however, because one does not know whether or not the activity took place. These cases either have to be discarded or else some method needs to be used to determine probable responses for the missing information.

### **MISSING AND ZERO INTERVIEWS**

To illustrate these principles, return to the crop purchase questionnaire in Figure 1. If the respondent had neither bought nor received any crops during the recall period, then this would be a *zero interview*, that is, the household received no crops. If, on the other hand, there was no information whatsoever for this household for the recall period (because, for instance, the head of household could not be interviewed or could not remember whether any transactions took place), then this would be a *missing interview*, that is, there is no way of telling whether or not the household received any crops.

How are these cases handled on the questionnaire? *An entry should be made for every interview—even for those that are missing or zero.* In the example of the crop purchase questionnaire, the date of the interview (or attempted interview, if none took place) and the number of days since the last interview are entered on the questionnaire in the first and second columns. There would be no entry for Crop Received, however, because there is no information on specific crops transacted. Since the legitimate crop codes range from 101 through 599, the special codes of 800 or 900 are used to denote a missing or zero interview, respectively. These lines are entered

---

<sup>3</sup> The format for this codebook entry was adapted from Hadden and Léger (1990).

into the data file along with the rest of the transactions. In Figure 1, the interviews for 7-16-89 and 8-12-89 illustrate the method for recording zero and missing interviews.

Entering missing and zero interviews serves three purposes. First, it allows the researcher to monitor when all interviews (or attempted interviews) took place. Second, it indicates those households that were interviewed but simply did not make any transactions (zeros). Third, it identifies observations that need to be excluded or estimated for analysis because of missing data.

## MISSING AND ZERO DATA ITEMS

Besides situations where information for an entire interview is missing, there may be cases where only specific data items for a particular transaction are missing. For example, the respondent may recall that he or she purchased some millet but may not be able to remember the price paid. In these cases, all available information should be entered on the questionnaire as usual, but the enumerator needs to indicate the missing items as well.

The manner in which missing data items are coded will depend upon the software used to enter and analyze the data. Some packages have a special value set aside to indicate a missing response. In SPSS/PC+, this value, called the *system missing value*, is indicated by a period (.). These missing values are automatically excluded from statistical computations by SPSS. Simply entering a period in SPSS Data Entry signifies that the value is missing for a given variable. On the questionnaire, however, the missing values should be indicated by a question mark (?), an M, or some other symbol.

Other software packages, like dBase III, do not have a special value for missing responses. Therefore, a numeric code needs to be designated as the missing value for each variable. Standard practice is to use a series of repeated nines to create a code that is greater than the largest legitimate value for the variable. For example, if the non-missing values for a variable range from 01 through 12, the missing value code for this variable would be 99.<sup>4</sup> This convention applies to both qualitative and quantitative (continuous) variables. These missing value codes would be entered both on the questionnaire and in the data file.

Returning to Figure 1, transaction 2 for the interview of 7-31-89 illustrates a case where the price paid for a packet of cowpeas was not recalled by the respondent. All other information is entered, but an "M" is written in payment boxes to indicate that the information is missing. This example assumes that there is a special nonnumeric missing value for the software, as with SPSS/PC+. For programs without a system missing value, numeric codes denoting the missing values would be entered. If the normal values for unit price and total payment range from 0 through 10,000, then the code 99,999 could be used to indicate the missing values for these items.

---

<sup>4</sup> SPSS/PC+ also allows otherwise nonmissing numeric values to be designated as *user missing values*. Once specified, user missing values behave similarly to the system missing value in computations.

Note that in transaction 1 for the interview of 7-1-89, only an in-kind payment was made for the purchase of rice. Since there was no cash payment, zeros are entered for the unit price and total cash payment to indicate unambiguously that there was no cash paid. This illustrates why it is generally not a good idea to use zero as a missing value; zero can be a legitimate data value in some situations.

It should also be noted that a missing value can sometimes be interpreted as "not applicable." Transaction 3 for the interview of 7-31-89 records a gift of peanuts received by the household. Since there are no payments made for a gift, a line is drawn through the blocks for the payment information to indicate that they do not apply to this transaction. In the data file, the payment data would be entered using a missing value. There is no problem in interpreting the missing values for these variables because the transaction is designated as a gift in column 7. The researcher must verify, however, that for each item there is no ambiguity between missing and not applicable responses. If there is the possibility of confusion, a separate code should be created for "not applicable."

## **REDUNDANT INFORMATION**

Another consideration in the design of questionnaires is the inclusion of otherwise redundant information. This allows the verification of both the data collection and the data entry. For example, on the crop purchase questionnaire the enumerator records the quantity of units purchased, the price per unit, and the total amount paid for each cash transaction. It is therefore possible to verify that the quantity times the price equals the total payment ( $\text{QUANTITY} \times \text{PRICE} = \text{TOT\_PMT}$ ).

Tests relying on redundant information are easily implemented on a computer. If the quantity, price, and payment do not agree, then either the enumerator or the data entry operator has made an error. To determine which was at fault, one should first verify that the data were correctly entered into the computer from the questionnaire. If the error is not at the data entry level, the enumerator should be consulted to discover the source of the problem.

## **SUMMARY**

The design of questionnaires and code sets has important implications for data entry. Consistent headers should appear on every questionnaire to permit proper data identification. Where appropriate, the questionnaire should contain spaces for recording both written and coded responses. Lists of coded responses should be developed in advance and kept in a codebook. Changes to these codes should be made with care and be fully documented.

All questionnaire coding systems should be able to differentiate between missing and zero interviews and have appropriate missing codes for individual data items. Some software packages, such as SPSS/PC+, have a special value used to indicate missing data. For packages that do not have such values, numeric codes will have to be designated as missing values for each variable. Finally, otherwise redundant information can be included on the questionnaire to permit verification of the data.

### 3 DATA FILES

All data collected on a questionnaire must be transcribed into the format of a data file so that it can be processed by the computer. This chapter begins with a brief description of the parts of a data file. It then discusses the concept of file levels and evaluates some different ways of creating data files from a questionnaire. It will be shown that the structure of the data file has important consequences for the ease with which the data can be used.

It is strongly recommended that the data files be set up before any data collection begins. Doing this can uncover problems in transferring the data from questionnaires into files, which may help minimize the amount of file manipulation that needs to take place before the data can be analyzed. It may also reveal potential problems in coding or interpreting responses to questionnaire items. Specifying the structure of the data files in advance can also force the researcher to articulate the exact meaning of individual questions and the types of responses that are expected.

#### ELEMENTS OF A DATA FILE

A *data file* is a collection of related information stored together in a form accessible by a computer. A file is identified by its *name*, which in DOS (the operating system of IBM compatible PCs) consists of up to eight letters or numbers followed by an optional three-character extension. For example, HELLO, FILE4.DAT, PETER.2 are all valid DOS file names. (Operating systems other than DOS may have different file naming conventions.)

There are many ways of organizing the information in a data file. The most common (and generally most useful for analysis) is known as the *rectangular file*. The rectangular file (sometimes called a flat file) can be visualized as a two-dimensional table (Figure 3):

Figure 3—Sample rectangular file

Record number	NAME	Variable name		
		AGE	WEIGHT	SEX
1	Moudou	50	54	M
2	Fatou	46	44	F
3	Marieme	37	46	F
4	Abdoulaye	17	51	M
5	Oulimata	16	36	F

File contents

In the rectangular file structure, columns represent *variables*—the basic types of information that are contained in the file. A person's name or the price of a product are examples of information that can be entered into variables. Like files, variables are also referred to by their names, which in SPSS/PC+ files consist of a single word of eight or fewer characters. Variables may be one of two types: *numeric variables*, which can contain only numbers, and *string (or alpha-numeric) variables*, which can include letters as well as numbers. NAME and SEX in the file shown above are examples of string variables, whereas AGE and WEIGHT are examples of numeric variables.

In naming variables, it is best to use names that are easy to understand. It is much easier to remember the meaning of variables such as PRODUCT, PRICE, and VILLAGE, for instance, than variables named X and Y. Certain programs (SPSS/PC+ included) permit the use of the characters "." or "\_" in variable names, which can make names easier to read (QNT.PUR and PRICE\_KG instead of QNTPUR and PRICEKG).

The rows of a rectangular file represent particular sets of values for each variable. In computer terminology, a file row is called a *record*.<sup>5</sup> Sometimes a record is described as a specific "occurrence" of a set of variables (Martin 1977, 12-14). The primary characteristic of the rectangular file is that each record has the same structure, that is, the exact same set of variables are represented in every record.

SPSS/PC+ and most other data analysis programs require that data be organized into rectangular files. Even given this restriction, there are often several different ways of arranging data from a questionnaire into rectangular files. Since the file structure affects the ease with which the data can be manipulated and analyzed, care must be taken to design data files correctly. This problem can best be illustrated by presenting three different methods of creating data files from a simple household census questionnaire and examining the advantages and disadvantages of each design. One method clearly emerges as preferable.<sup>6</sup>

## HOUSEHOLD CENSUS

Figure 4 shows a simple household census questionnaire that could be used to collect data on the characteristics of households in a survey sample. The questionnaire records information that is specific to the household—whether the household head is the village chief, the use of animal traction by the household, the distance from the compound to the main road, and the household's principal ethnic group. It also collects data on each household member—name, sex, age, and the person's relationship to the household head.

<sup>5</sup> Sometimes a record is referred to as a "case." The problem with this terminology, however, is that "case" can also carry the connotation of "observation." As will be seen, a file record and an observation are not necessarily equivalent. In order to avoid confusion, the rows in a rectangular file will be referred to exclusively as records.

<sup>6</sup> The illustration of the concept of file levels that appears in the remainder of this chapter has been adapted from Crawford et al. (1988, B3-16).

There are several ways of transferring the information from this questionnaire into a data file. In Figure 4, the data file variable names have been written on their corresponding fields in the questionnaire and the complete file structure is shown below the questionnaire. Notice that all the data have been put into a single file and that *there is exactly one record in the file for every household in the sample*. In order to accomplish this, a parallel set of variables containing the data on each household member are used, that is, the name of the first household member is NAME1, the second NAME2, and so forth.

There are several problems with this approach. To begin with, the data on the household members will be difficult to analyze. Most statistical software packages have straightforward commands for calculating basic descriptive statistics on a variable. For example, the SPSS/PC+ DESCRIPTIVES command will calculate the mean, standard deviation, and minimum and maximum values of individual variables in a data file. Calculating descriptive statistics

Figure 4—Method 1 of organizing census data files

QUESTIONNAIRE #01 -- HOUSEHOLD CENSUS

Village  HH ID  Household Head \_\_\_\_\_

Date of Interview:

Village chief? (Y/N)  Animal Traction? (Y/N)

Distance from main road:  meters

Principal ethnic group: \_\_\_\_\_ Code:

No. Person	Name	Sex (M/F)	Age (Yrs)	Relation to HH Head	Code
1	NAME1	SEX1	AGE1		REL1
2	NAME2	SEX2	AGE2		REL2
3	NAME3	SEX3	AGE3		REL3

File: VIL HH MO DY YR CH AT DISTANCE ETHN NAME1 SEX1 AGE1 REL1 NAME2 SEX2 AGE2 REL2 NAME3 ...

on the variable DISTANCE, for instance, is simply a matter of giving the command, DESCRIPTIVES DISTANCE.

Because the data on household members are in several different variables, however, these simple commands will not work. Instead, in order to calculate the average age of the members of a household, the following formula must be used:

$$\text{Average Age} = \frac{AGE1 + AGE2 + \dots}{N}$$

Note that  $N$ , the number of household members, is not included in the file and would have to be determined separately. Although this average formula is not complicated, consider the formula for the variance:

$$\text{Variance Age} = \frac{1}{N-1} \frac{(AGE1 - AVGAGE)^2 + (AGE2 - AVGAGE)^2 + \dots}{AGE1^2 + AGE2^2 + \dots}$$

One would first have to determine the average age (AVGAGE) before computing the variance. It is clear that this file would not be easy to work with.

An additional problem is that the data file in Figure 4 does not use space very efficiently. Each record in the file must have the maximum possible number of household member variables, even if all of these variables will not be needed for every household. If, for instance, the largest household in the sample has 32 people, the file must have 32 sets of variables for the names, sexes, ages, and relationships for every household (that is, NAME1 through NAME32, SEX1 through SEX32, and so forth). Each record will therefore need space for 32 people, or  $32 \times 4 = 128$  variables. Since it is unlikely that many households will have 32 members, the unused variables would be wasted space in the file. If, on the other hand, one later comes across a household with 33 members, the original file structure would have to be modified to accommodate the additional person.

To avoid the inflexibility inherent in the first data file, each file record could represent a *household member*, rather than an entire household. This layout is presented in Figure 5. Note that the variable NOPER, a sequential numbering of persons in the household, has been added to the file to identify the household member associated with each record. The household data (MO, DY, YR, CH, AT, DISTANCE, and ETHN) would be repeated for each household member, so that every person in a given household has exactly the same set of values for these variables.

With this structure, the maximum possible household size does not have to be predicted. The file can readily accept households with any number of members by simply adding more records (adding records to a file is generally easier than adding variables). Unfortunately, this structure also has some drawbacks. It is not possible to calculate statistics on the household variables directly from this file because each record no longer represents a single household.

Figure 5—Method 2 of organizing census data files

QUESTIONNAIRE #01 -- HOUSEHOLD CENSUS

Village            HH ID            Household Head \_\_\_\_\_  
           

Date of Interview:   

Village chief? (Y/N)                Animal Traction? (Y/N)   

Distance from main road:     meters

Principal ethnic group: \_\_\_\_\_ Code:   

No. Person	Name	Sex (M/F)	Age (Yrs)	Relation to HH Head	Code
NOPER	NAME	SEX	AGE		REL

File: VIL HH MO DY YR CH AT DISTANCE ETHN NOPER NAME SEX AGE REL

In addition, the new structure still wastes space because of all the repeated household information. Duplicating the household data also adds to data entry and editing time. If any corrections need to be made to the household data, these changes must now be made to multiple records for each household. Not only is this inefficient, but mistakes are more likely in entering and correcting multiple entries of the same information, resulting in different values of household variables for the same household.

The preferred method for placing the data from this questionnaire into files is shown in Figure 6. In this case, the data have been separated into two files: file A contains all of the household data and file B the household member data. In file A there is one record for every household in the sample, and in file B there is one record for every person in the sample.

This arrangement uses the minimum amount of space for storing the data and eliminates all of the duplicate information. These data are also easy to analyze. The SPSS DESCRIPTIVES command can now be used on all variables because in both files *each record represents a single observation*. With this structure there is a direct

Figure 6—Method 3 of organizing census data files

QUESTIONNAIRE #01 -- HOUSEHOLD CENSUS

Village  HH ID  Household Head \_\_\_\_\_

Date of Interview:

Village chief? (Y/N)  Animal Traction? (Y/N)

Distance from main road:  meters

Principal ethnic group: \_\_\_\_\_ Code:

No. Person	Name	Sex (M/F)	Age (Yrs)	Relation to HH Head	Code
<u>NOPER</u>	<u>NAME</u>	<u>SEX</u>	<u>AGE</u>		<u>REL</u>

File A: VIL HH MO DY YR CH AT DISTANCE ETHN

File B: VIL HH NOPER NAME SEX AGE REL

Note: Key variables are underlined.

correspondence between the statistical observation and the data file records.

## LEVELS OF DATA AND KEY VARIABLES

The file structure in Figure 6 is easier to use than the previous two methods because the two different levels of the census data have been put into separate files. The *level* of a data file is the way in which file records are classified in that file. That is, the level is the combination of elements that can uniquely identify one record of data within the entire file. Conceptually, the level of the file is equivalent to the "unit of observation" of the data in that file (Crawford, et al. 1988, B9-10).

In the household census, information is being collected on each household as well as on each household member. The household-member data are said to be at the person level, or, more precisely, at the *village/household/person* level.<sup>7</sup> In other words, there will

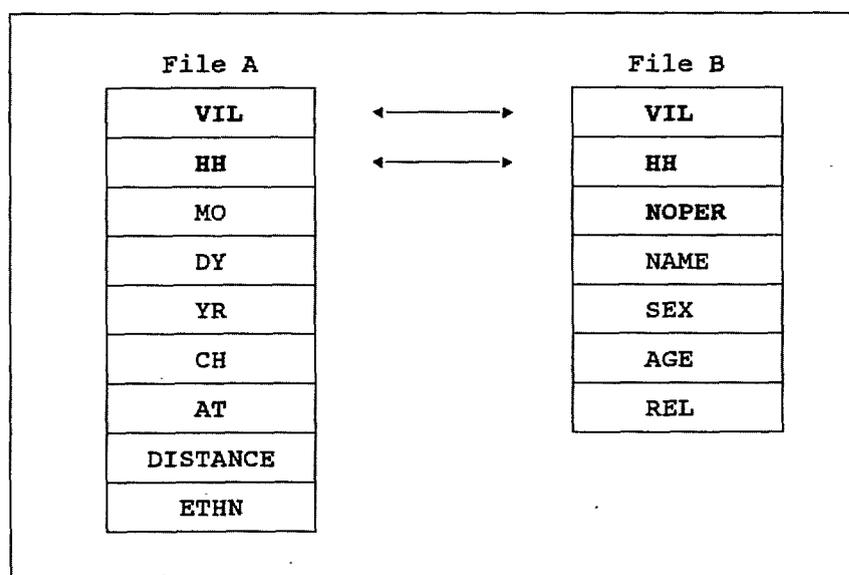
<sup>7</sup> This notation is adapted from Crawford et al., 1988, B10.

be a single record in the data file for each household member in every village. All three of these elements are necessary to identify uniquely each individual in the census. Similarly, the household data are at the *village/household* level because the elements village and household uniquely identify each household. (In this example, it is assumed that the census is performed only once during the study. If, however, it were administered annually, the year would also need to be included in the level description, as in *year/village/household*, so that records from different years could be distinguished from one another.)

In order to define the level of the data in a file, the file must include certain *key variables*. Key variables identify uniquely each data record among all the other records. Uniqueness is an important property for key variables—no two records should have the same combination of values for the key variables. In the example of the household census, the variables VILLAGE and HH are the key variables for file A, while VILLAGE, HH, and NOPER are the key variables for file B. These variables uniquely identify each household and person in their respective files. Variables that are not key variables are called *attribute variables* (Crawford et al. 1988, B8-9; Martin 1977, 206-9).

Taken together, all of the data files for all the different questionnaire types comprise a *database*. The individual rectangular files in the database are related to each other in ways described by the key variables. For example, the relationship between the two census files is given by the following schematic, with the key variables in bold (see Figure 7).

**Figure 7—Sample schematic of a database**



## USING DATA IN MULTIPLE RECTANGULAR FILES

It is helpful to use consistent variable names for key variables across different data files. For instance, village should be indicated by the variable VIL in every file, household by HH, and so forth. This makes the relationships between individual files clearer and makes combining data from different questionnaires easier.

The method of organizing data into rectangular files based on levels and relating them to each other through key variables is known as the *relational database model*. Most database software packages (dBase, for instance) operate according to this model.

The levels of data on a questionnaire should also be taken into account when the questionnaires are being designed. Many data entry programs cannot enter data to multiple files, so the data for each level will have to be entered in separate passes. It is therefore better to group together on the questionnaire all of the items at the same level so that the data entry operator will be able to find the data for each file more easily.

As was pointed out earlier, organizing data by levels into multiple rectangular files allows one to calculate statistics on variables within each file in a straightforward way. It will also be necessary, however, to combine data from different files. This section gives two examples of the merging of data files at the household and person levels for the census questionnaire.

Figure 8 shows a sample of data for files A and B of the census questionnaire presented in Figure 6. In this first example, the household-level variables AT and DISTANCE are added to the person-level data. That is, to each person in FILE\_B.SYS, the household characteristics of animal traction use and distance to the main road are to be added from FILE\_A.SYS. These characteristics should be repeated for each member in a given household.

To accomplish this in SPSS/PC+, use the JOIN MATCH command:

```
JOIN MATCH
  /TABLE 'FILE_A.SYS'
        /KEEP VIL HH AT DISTANCE
  /FILE 'FILE_B.SYS'
  /BY VIL HH.
```

The use of the /TABLE option with FILE\_A.SYS causes the values for AT and DISTANCE to be repeated for each person in the same household. The resulting file is shown in Figure 9. (In this example it is assumed that both FILE\_A.SYS and FILE\_B.SYS are sorted by VIL and HH. This is necessary for the JOIN MATCH command to work properly.)

As a second example, information from the person-level data (FILE\_B.SYS) is added to the household-level data (FILE\_A.SYS). In order to do this, the person-level data must first be *aggregated* to the household level. Aggregation is the process of combining into a single record all the records in a file with the same values for a subset of key variables. Normally, this process involves simultaneously calculating statistics on one or more attribute variables.

Figure 8—Sample data files for census questionnaire

File: FILE\_A.SYS

VIL	HH	MO	DY	YR	CH	AT	DISTANCE	ETHN
1	1	9	12	88	Y	Y	20	1
1	2	9	12	88	N	Y	150	1
1	3	9	14	88	N	N	200	2
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

File: FILE\_B.SYS

VIL	HH	NOPER	NAME	SEX	AGE	REL
1	1	1	Adamu	M	50	1
1	1	2	Kedibo	F	40	2
1	1	3	Moru	M	26	5
1	2	1	Daouda	M	31	1
1	2	2	Djebo	F	18	2
1	3	1	Oumarou	M	42	1
1	3	2	Meretou	F	30	2
1	3	3	Jitu	F	19	3
1	3	4	Idrissa	M	12	5
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Figure 9—Results of first match of census data files

VIL	HH	NOPER	NAME	SEX	AGE	REL	AT	DISTANCE
1	1	1	Adamu	M	50	1	Y	20
1	1	2	Kedibo	F	40	2	Y	20
1	1	3	Moru	M	26	5	Y	20
1	2	1	Daouda	M	31	1	Y	150
1	2	2	Djebo	F	18	2	Y	150
1	3	1	Oumarou	M	42	1	N	200
1	3	2	Meretou	F	30	2	N	200
1	3	3	Jitu	F	19	3	N	200
1	3	4	Idrissa	M	12	5	N	200
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

For example, suppose one wishes to add the average age of the household to the household-level data. First, the person-level file must be aggregated by VIL and HH, taking the mean of AGE to create a new variable, AVG\_AGE. This is done using the SPSS/PC+ AGGREGATE command:

```
GET FILE 'FILE_B.SYS'.

AGGREGATE OUTFILE *
  /BREAK VIL HH
  /AVG_AGE = MEAN( AGE ).
```

The resulting file from the AGGREGATE command is shown in Figure 10.

This aggregated file can then be merged with FILE\_A.SYS using the JOIN MATCH command:

```
JOIN MATCH
  /TABLE *
  /FILE 'FILE_A.SYS'
  /BY VIL HH.
```

The combined data file is given in Figure 11. The asterisk (\*) after the /TABLE option is used to indicate the current active file, which was created by the AGGREGATE command. (Again, it is assumed in this example that FILE\_A.SYS is sorted by VIL and HH.)

**Figure 10—Results of aggregating FILE\_B.SYS to calculate average age**

VIL	HH	AVG_AGE
1	1	38.67
1	2	24.50
1	3	25.75
.	.	.
.	.	.
.	.	.

**Figure 11—Results of second match of census data files**

VIL	HH	MO	DY	YR	CH	AT	DISTANCE	ETHN	AVG_AGE
1	1	9	12	88	Y	Y	20	1	38.67
1	2	9	12	88	N	Y	150	1	24.50
1	3	9	14	88	N	N	200	2	25.75
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

## SUMMARY

As was noted at the beginning of this section, the final form of the data should be considered when the questionnaires are being designed. It is highly recommended that the structure of all data files be set up before the data collection even begins. In this way, one can see all the different data files that will be produced from the different types of questionnaires. If these files are not appropriate for the analysis to be performed, the design of the questionnaires can be modified and less time will be wasted trying to rearrange data files after the data have been entered.

The most common data file format is the rectangular file. Questionnaire data should be separated by levels when being organized into rectangular files, which may mean separating the data from a single questionnaire into two or more files. The level of a file is defined by key variables that uniquely identify each file record. Data in this form will be easier to analyze because each record will be equivalent to a single observation. Through the use of commands such as JOIN MATCH and AGGREGATE, these files can be combined in any way desired.

## 4 DATA ENTRY

The process of data entry involves not only typing information into the computer, but also implementing a whole set of procedures for preparing and organizing the questionnaires and processing and verifying the data. Although most people focus on the computer aspect of data entry, many of the problems involved in successfully organizing a data entry system are related to management rather than to the computer.

This chapter describes a data entry and verification system similar to the one used for the IFPRI household studies in Niger and Senegal. In this system, data entry is carried out concurrently with data collection. A flow chart of the complete system begins when the questionnaires are first received in the office and continues through several levels of data verification (see Figure 12). This chapter covers the steps from questionnaire reception through data entry, and the next chapter discusses the data verifications.

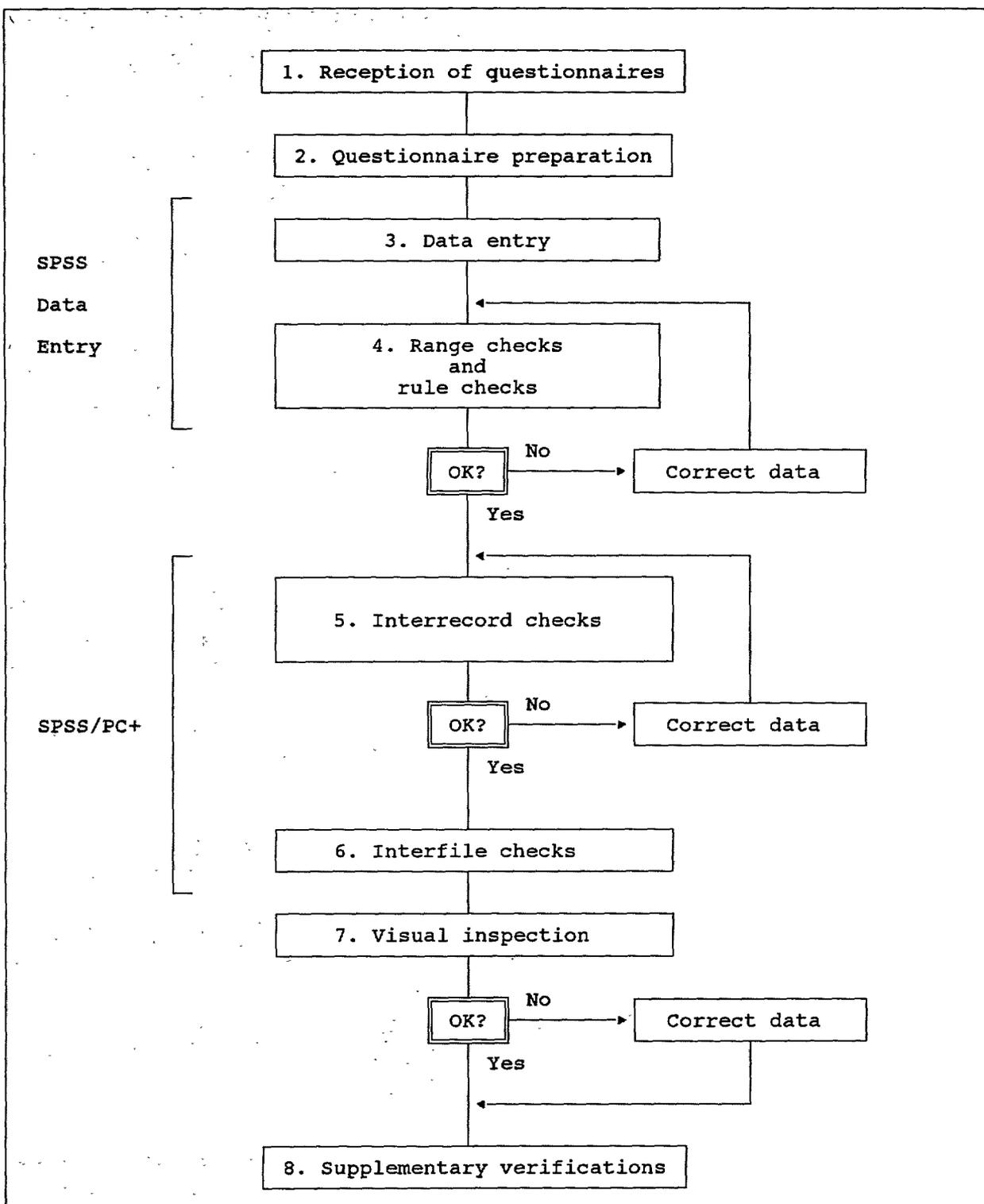
It is beneficial to include a list of instructions covering all data entry and verification procedures in a *procedures manual*. This manual should be in the form of a loose-leaf notebook, so that it can be augmented and modified as procedures are revised. It might include specific instructions for correcting certain types of errors or key-stroke-by-key-stroke descriptions of how to use the data entry and analysis software. The manual would serve as a reference to all staff and help ensure that everyone is following the same procedures.

### **STEP 1: RECEPTION OF QUESTION- NAIRES**

It is important to have established procedures for collecting questionnaires from the field and for handling them once they have been received. During a long survey, the questionnaires should be collected periodically from the enumerators so that they may be entered into computer files. Upon receiving the questionnaires, they should be separated into *batches* by type. That is, all the census questionnaires should be put in one batch, all the crop transaction questionnaires in another, and so forth. If the batches are quite large, they may be further separated by region or village.

Once the questionnaires have been separated into batches, they are sorted by their key variables and numbered sequentially. In the case of the household census, the questionnaires would be sorted by village, household, and the ID number of the household member. The questionnaire numbering is very important because it keeps the questionnaires in order and allows one to see quickly if a questionnaire is missing. The questionnaire numbers can also be entered into the data file, making it easier to find the questionnaire corresponding to a particular record. For multiple page question-

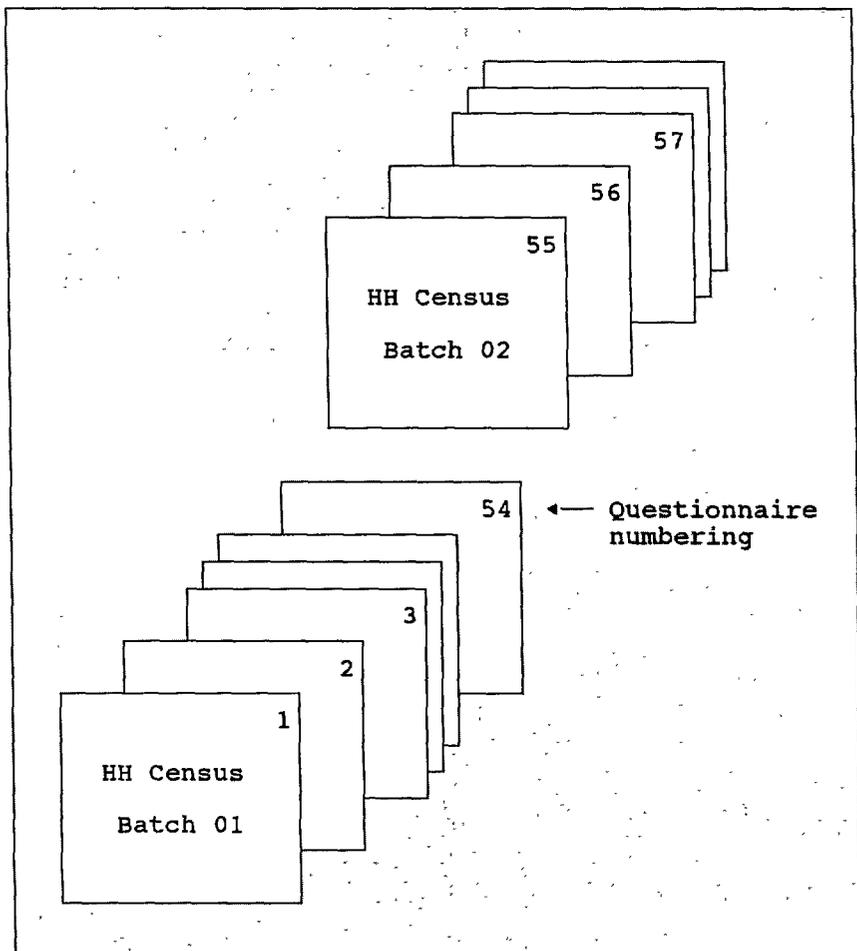
Figure 12—Flow chart of data entry and verification system



naires, each sheet within a particular questionnaire should have the same number.<sup>8</sup>

The questionnaire numbering should continue from one batch to the next for the same type of questionnaire. For example, if the last page number in the first batch of household census questionnaires was 54, the numbering of the next batch of census questionnaires should start with 55. In this way, no two questionnaires of a given type will have the same number. The batches themselves are also numbered, so that the first batch of census questionnaires will be number 01, the second 02, and so forth (see Figure 13).

**Figure 13—Batch and questionnaire numbering**



<sup>8</sup> Alternatively, the questionnaires could be numbered *before* they are sent to the field. This would make it possible to reassemble pages from a questionnaire that became detached while being transported from the field to the survey office.

Once the questionnaires have been sorted and numbered, a *Data Entry and Verification Control Form* (Figure 14) is attached to each batch. This form is used to record the completion of the different steps in the data entry and verification process. The following information is entered in the form header: the date when the batch was received, the questionnaire ID number (which identifies the questionnaire type), and the batch number. The summary box below the header has a check list of all the major steps and allows one to see immediately the status of the questionnaire.

Figure 14—Data entry and verification batch control form

DATA ENTRY AND VERIFICATION CONTROL FORM																																																									
<div style="border: 1px solid black; padding: 5px; margin: 0 auto; width: 80%;">           Date Received _____ Questionnaire ID No. _____ Batch No. _____         </div>																																																									
<div style="border: 1px solid black; padding: 5px; margin: 0 auto; width: 80%;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Preparations _____</td> <td style="width: 50%;">Interfile checks _____</td> </tr> <tr> <td>Data entry _____</td> <td>Visual inspection _____</td> </tr> <tr> <td>Ranges/rules _____</td> <td>Approved _____</td> </tr> <tr> <td>Interrecord checks _____</td> <td>Forward to supervisor _____</td> </tr> </table> </div>				Preparations _____	Interfile checks _____	Data entry _____	Visual inspection _____	Ranges/rules _____	Approved _____	Interrecord checks _____	Forward to supervisor _____																																														
Preparations _____	Interfile checks _____																																																								
Data entry _____	Visual inspection _____																																																								
Ranges/rules _____	Approved _____																																																								
Interrecord checks _____	Forward to supervisor _____																																																								
QUESTIONNAIRE PREPARATION																																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Vil</th> <th style="width: 25%;">Questionnaire No.</th> <th style="width: 20%;">Date</th> <th style="width: 50%;">By</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	Vil	Questionnaire No.	Date	By																									<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Vil</th> <th style="width: 25%;">Questionnaire No.</th> <th style="width: 20%;">Date</th> <th style="width: 50%;">By</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	Vil	Questionnaire No.	Date	By																								
Vil	Questionnaire No.	Date	By																																																						
Vil	Questionnaire No.	Date	By																																																						
DATA ENTRY																																																									
Level 1			Level 2																																																						
File: _____			File: _____																																																						
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Vil</th> <th style="width: 25%;">Questionnaire No.</th> <th style="width: 20%;">Date</th> <th style="width: 50%;">By</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	Vil	Questionnaire No.	Date	By																									<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Vil</th> <th style="width: 25%;">Questionnaire No.</th> <th style="width: 20%;">Date</th> <th style="width: 50%;">By</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	Vil	Questionnaire No.	Date	By																								
Vil	Questionnaire No.	Date	By																																																						
Vil	Questionnaire No.	Date	By																																																						
(continued)																																																									

Figure 14—continued

**SPSS DATA ENTRY CHECKS (RANGES/RULES)**

<b>Level 1</b>				<b>Level 2</b>			
1st Report:	Date _____	By _____		1st Report:	Date _____	By _____	
Final Report:	Date _____	By _____		Final Report:	Date _____	By _____	

**INTERRECORD CHECKS**

1st Report:	Date _____	By _____		1st Report:	Date _____	By _____	
Final Report:	Date _____	By _____		Final Report:	Date _____	By _____	

**INTERFILE CHECKS**

1st Report:	Date _____	By _____		1st Report:	Date _____	By _____	
Final Report:	Date _____	By _____		Final Report:	Date _____	By _____	

**VISUAL INSPECTION**

<b>Level 1</b>				<b>Level 2</b>				
Vil	Questionnaire No.	Date	By		Vil	Questionnaire No.	Date	By

Approved: Date \_\_\_\_\_ Forward to supervisor: Date \_\_\_\_\_

The receipt of the questionnaires is recorded in a logbook as shown in Figure 15. The logbook is divided into separate sections for each questionnaire type. Each time questionnaires are delivered to the office, the batch number and the date the batch was received are entered in the logbook. In addition, each village included in the batch is listed along with the number of questionnaires and the dates of the first and last interviews. This allows one to keep track of which questionnaires have been brought in for data entry and which are still in the field. The number of questionnaires is totaled at the end of each entry.

Figure 15—Logbook entry for recording the receipt of questionnaires

CROP PURCHASE QUESTIONNAIRE					
<u>Batch Number</u>	<u>Date Received</u>	<u>Village</u>	<u>Number of Questionnaires</u>	<u>First Interview</u>	<u>Last Interview</u>
01	12/04/88	Sagatta	17	10/19/88	11/02/88
		Khelcome Peulh	14	10/19/88	11/12/88
		Darou Cissé	11	10/17/88	11/27/88
		TOTAL	42		
02	12/20/88	Touba Toul	18	10/30/88	12/01/88
		Thylla Boubou	16	10/30/88	12/01/88
		Niakhar	18	11/05/88	12/01/88
		TOTAL	52		

## STEP 2: QUESTIONNAIRE PREPARATION

Before the questionnaires can be entered, they must be prepared, which involves a visual inspection of each questionnaire for missing or inconsistent information. (Of course, the questionnaires should also be inspected in the field by the enumerator or the field supervisor before they are brought to the office.)

Questionnaire preparation is an extremely important step that is often omitted for the sake of expediency. Experience has shown, however, that this step can reveal many data collection problems and that more time is wasted by allowing such problems to go undetected until the later stages. In the quest to find data errors, the motto must be "the sooner, the better."

The person preparing the questionnaires must be quite familiar with them so that he or she will be able to interpret correctly the information on them. The preparer writes in codes for any uncoded responses and searches for missing responses, inconsistent information, or entries that simply "don't look right." This requires someone with a careful eye and a good "feel" for the data (Casley and Lury 1987).

Some of the problems uncovered during the preparation may be resolved quickly, while others will require consulting the enumerator for clarification. The *Questionnaire Preparation Form* (Figure 16) is used to record any problems that need further explanation. The person performing the preparations enters the following information in the header of this form: his or her name, the questionnaire ID number, and the batch number. For each problem needing



Since it will take some time to receive a response from the field, it may not be efficient to postpone the data entry until all the outstanding problems are resolved. Unless the problems are too many or too serious, the data entry should proceed once the preparations have been completed. The remaining corrections can be made at a later point in the process.

When making corrections to the questionnaire, the original entries made by the enumerator should never be obliterated or made illegible. It is often extremely useful to examine the entry originally made by the enumerator, it may be necessary, for instance, to undo the correction and restore the original response. The proper procedure for correcting a questionnaire is to draw a single line through the incorrect response and write the new entry above it.

### **STEP 3: ENTERING THE DATA INTO THE COMPUTER**

Once the questionnaire preparation is completed, the data are entered into files. Each batch should be entered into a separate set of data files. This is necessary because different batches of questionnaires will be at different stages of data entry and verification, and it would be too confusing to have both cleaned and uncleaned records in the same file.

The data file name should include both the name of the questionnaire and the batch number. One possibility is a three or four letter specification for the questionnaire followed by the batch number. For example, CPUR01.SYS would be the first batch of the crop purchase questionnaire. (The .SYS extension is used to denote an SPSS/PC+ data file.)

If, however, the questionnaire has more than one level of data, the batch must be broken down further into separate data files for each level. A single letter may be added to the file name to indicate the level. For example, "H" could indicate the household-level file and "P" the person-level file of the census data. So, the first batch of the census questionnaire would be entered into two files, CEN01H.SYS and CEN01P.SYS.

A danger with the file naming system described above is that, because the file names are so similar, it is very easy for someone to make a mistake when typing a file name, entering "04" instead of "05," for instance. This could result in data from a previously entered batch being overwritten. The best insurance against the accidental loss of data is to make regular backups of the data files. Doing this also provides protection against hard disk crashes or other computer hardware failures. (Chapter 7 will discuss backups in more detail.)

Note that the DATA ENTRY section of the control form (Figure 14) contains two separate columns to accommodate two different levels of data in a questionnaire. (This form would not be adequate for questionnaires with more than two levels.) Since many data entry programs (SPSS Data Entry included) do not allow one to enter data to multiple files simultaneously, the data entry operator should enter all the data for one level before proceeding to the next level. For example, all the data would be entered for CEN01H.SYS

## SPECIAL FEATURES OF SPSS DATA ENTRY

before entering the data for CEN01P.SYS. The operator records the progress of the data entry for each file on the control form, and the summary box is checked once both files have been entered.

The data entry component of SPSS/PC+ has a number of features designed to make data entry easier and more efficient. Two of these, data entry forms and skip and fill rules, will be discussed here.

With SPSS Data Entry, custom data entry forms can be designed that resemble the printed questionnaire. This makes it easier for the data entry operator to find the variable corresponding to a given question. Creating a data entry form is a simple matter of positioning labels, lines, and data entry fields on the screen using the arrow keys. The disadvantage of the customized forms is that only one record of data is visible at a time. The operator can quickly switch between the custom form and a more conventional spreadsheet format, however, which displays 20 records at once (although it may not be able to fit all variables on the screen at the same time).

Skip and fill rules allow the user to control the order in which variables are entered and to assign values to variables automatically. Each skip and fill rule is associated with a specific variable in the file, and it is executed immediately after a value has been entered for that variable.

One use of skip and fill rules is to skip over variables that are not applicable to a particular record. For example, if a transaction entered on the crop purchase questionnaire (Figure 1) was a gift, the operator may want to jump directly to the next record in the data file, as the remaining variables (payments and reason for purchase) do not apply to nonpurchases. The following skip and fill rule for the variable UNIT will accomplish this:

```
IF (PUR_GIFT = 2) NEXTCASE;
```

Once the variable UNIT has been entered, the value of PUR\_GIFT is examined. If PUR\_GIFT is 2, then the transaction was a gift and SPSS Data Entry proceeds directly to the next record. If PUR\_GIFT is not 2, then the transaction was a purchase and the program continues (by default) with the next variable in the file (unit price paid).

A skip and fill rule can also be defined to skip over the remaining variables describing an in-kind payment if nothing is entered for the type of good used as payment. The following rule defined for TYP\_INKD (type of good for in-kind payment) will skip to the variable REASON (reason for purchase) if TYP\_INKD is missing:

```
IF (TYP_INKD = .) -> REASON;
```

A second use of skip and fill rules is to assign values to variables. For example, given the quantity purchased and the price of a crop, one might want to calculate the total payment with the following rule:

```
TOT_PMT = QUANTITY * PRICE;
```

This rule would have to be associated with both QUANTITY and PRICE, since changing either of them would require recalculating TOT\_PMT. (Note that one should not implement this rule if TOT\_PMT is being entered as redundant information in order to check that the total payment, quantity, and price agree.)

Skip and fill rules can also be used to assign default values to variables. Suppose, for instance, that transactions on the crop purchase questionnaire most frequently have "kilogram" as a unit of measure. Rather than having to type the code for kilogram (1) each time, a skip and fill rule associated with the variable QUANTITY can enter a default value:

```
IF (UNIT = .) UNIT = 1;
```

The above rule will cause the value of 1 to appear in the UNIT variable field immediately after the variable QUANTITY is entered. If the unit entered on the questionnaire is a kilogram, the operator can simply press the ENTER key and proceed to the next variable. If it is some other unit, however, the operator enters the code for that unit before pressing ENTER.

Note that this skip and fill rule first checks to make sure that UNIT is missing before assigning the default value of 1. This prevents a previously entered value for UNIT from being unintentionally replaced. When designing skip and fill rules, the programmer should make sure that entered values will not be replaced unintentionally. SPSS Data Entry allows one to deactivate all skip and fill rules temporarily; this may be helpful once all the data have been entered and corrections are being made to the file. The skip and fill rules should not be deactivated if they are being used to assign values to nonentered variables, however.

## SUMMARY

This chapter described the first three steps of the data entry and verification system. Step 1 involves procedures carried out when the questionnaires are received in the office—organizing questionnaires into batches, sorting and numbering them, and recording their receipt. Step 2 is questionnaire preparation, which requires the inspection of each questionnaire for missing or inconsistent information. Problems uncovered during preparation are recorded on a special form that is sent to the field for resolution.

Step 3 is the entering of each batch of questionnaires into a separate set of data files. Multiple files are required for each batch if the questionnaire contains more than one level of data. Two features of SPSS Data Entry, custom data entry forms and skip and fill rules, can greatly facilitate the data entry process.

## 5 DATA CLEANING

Once the questionnaire information is entered into data files, the quality of the data must be verified, a process often referred to as *data cleaning*. One of the main advantages of concurrent data entry is that various computer checks can be used to uncover data errors early on in the survey. Computers perform repeated tasks very rapidly and with great accuracy, an ability that makes them ideally suited to detecting certain kinds of data problems. Nevertheless, some errors can only be found manually; although computer verifications can be extremely valuable in improving the quality of data, they are not a substitute for careful training and supervision of enumerators and data entry operators.

Several levels of verification are addressed in this chapter. The first, *ranges*, verifies the values of individual variables. The second, *rules*, allows the comparison of different variables within the same record. Finally, interrecord checks test the relationships between different records in a data file, and interfile checks compare data from different data files.

Before describing the various data verifications, however, one must first distinguish between two different types of data errors. The first are *data entry errors*—mistakes made while entering the data from the questionnaire into the computer. These could involve mistyping a code (entering "7" instead of "1", for example), omitting a record of data, or entering the same record twice.

The second type of error involves *inconsistent or missing data*. In this case, the information was entered correctly from the questionnaire but some of the data are missing or "do not make sense" when compared with other information. Detecting this type of error often involves inspecting different variables in the data file or verifying the data against other sources. For example, an error would be indicated if a person were listed as the wife of the head of household but with sex designated as "M" for male. Or, the quantity of crops sold according to the transaction data may not agree with the quantity of crops produced in the crop production data.

The distinction between these two types of errors is important because each calls for a different method of correction. Data entry errors can be fixed quite simply by changing the data file so that it agrees with the questionnaire. Problems involving inconsistent or missing data, however, can only be resolved by consulting other sources, such as the enumerator or the respondent. This raises the important issue of who should be responsible for deciding how to correct each error type. In the IFPRI studies, it was decided that the data entry operators were to correct only data entry errors—they were not to make revisions to the questionnaires.

Inconsistent or missing data problems frequently involved judgments that could only be made by the researcher. This division of responsibilities needs to be made clear from the very beginning to everyone working with the questionnaires.

When correcting inconsistent or missing data, *one should correct not only the data file but the questionnaire as well.* It is very important that the questionnaire and the data file always agree, otherwise it will become difficult to remember which contains the correct information. The control form (Figure 14 in Chapter 4) is used to record the completion of the different verification steps described below.

#### **STEP 4: RANGE CHECKS AND RULE CHECKS**

SPSS Data Entry allows one to specify two categories of data checks: ranges and rules. Examples of each of these are presented below, using the crop purchase questionnaire to illustrate (see Figure 1 in Chapter 2). *Ranges* are simply a list of all values that are valid for a particular variable. For example:

<u>Variable</u>	<u>Range</u>
VIL	1,2,3,4,5,6
PRICE	50 THRU 300 \$SYSMIS

The operator THRU indicates that the values 50 through 300 inclusive are acceptable for PRICE. Given these range specifications, an error would be reported if the value of VIL was, say, 11 or if PRICE was 640. Ranges should be defined for all numeric variables in the data files. For discrete (that is, coded) variables, such as VIL, the list of valid codes is specified in the range. For continuous variables, such as quantity or price, a range of values is indicated using THRU. Note that the system missing value (\$SYSMIS) is included in the range for PRICE because it is legitimate for the purchase price to be missing if the transaction was a gift.

Ranges are verified as soon as a variable is entered. If the value falls outside the variable's range, the operator is immediately notified. This can save time by catching errors at the point of data entry.

*Rules* specify relationships between different variables within the same data record. They cannot compare values from different records, however. Rules are useful for identifying three types of problems: *missing data*, *mathematical relationships*, and *conditional relationships*. Examples of each of these are given below.

As was discussed in the section on missing responses in Chapter 3, certain variables may have missing values in some circumstances. Rules can be used to verify that variables only have missing values when they are supposed to. The following rule illustrates this function:

```
(PUR_GIFT = 1 IMPLIES PRICE <> . & TOT_PMT <> .) &
```

```
(PUR_GIFT = 2 IMPLIES PRICE = . & TOT_PMT = .)
```

The first part of this rule states that if the transaction was a purchase (PUR\_GIFT = 1), then the cash payment variables should not be missing (PRICE <> . means PRICE is not missing). The second part of the rule states that if the transaction was a gift (PUR\_GIFT = 2), then the payment variables *should* be missing. An error will occur if either of these conditions is violated.

Rules can also be used to verify *mathematical relationships*:

```
QUANTITY <> . & PRICE <> . & TOT_PMT <> . IMPLIES
```

```
TOT_PMT = QUANTITY * PRICE
```

This rule confirms that the total payment made for a purchase (TOT\_PMT) is equal to the quantity of units purchased (QUANTITY) times the price per unit (PRICE). The rule is conditional on all three variables having no missing values, since the statement TOT\_PMT = QUANTITY \* PRICE would produce an error if any one of the variables was missing. Since the first rule shown above already checks for missing values in the payment variables, it would be redundant and confusing to have another rule reporting the same error.

A third type of problem that can be tested with a rule is a *conditional relationship*. This is when the legitimate values for a particular variable depend upon the values of other variables. The rule below compares the crop purchased with its form:

```
(CROP IN 101 THRU 104 IMPLIES FORM IN 1,2) &
```

```
(CROP IN 106,108,109,112 IMPLIES FORM IN 3,4) &
```

```
(CROP IN 105,111,123 IMPLIES FORM IN 1,2) &
```

```
(CROP IN 113 THRU 122 IMPLIES FORM IN 5,6,7,9) &
```

```
(CROP IN 124 THRU 199 IMPLIES FORM IN 5,6,7,9)
```

For each value of CROP, there are corresponding possible values for FORM. For example, millet (CROP = 101) may be either unthreshed (FORM = 1) or threshed (FORM = 2), but peanuts (CROP = 108) are either in the shell (FORM = 3) or unshelled (FORM = 4).

SPSS Data Entry can produce a report listing all the records that violate each range and rule. If no errors are reported, the next set of verifications, interrecord checks (described below), are carried out. If there are errors reported, however, they must be identified as either data entry errors or inconsistent or missing data. If the entered data do not match what is on the questionnaire, then it is a data entry error and may be corrected immediately by making the file agree with the questionnaire. If, however, the data were entered correctly from the questionnaire, then it will be necessary for the researcher to decide whether a solution to the problem can

be inferred, or whether more information is required before a correction can be made.

Once all the data entry errors are corrected, a new range/rule report is generated in order to verify that the corrections have been made and that no new errors have been introduced in the correction process. This procedure is repeated until all data entry errors and other immediately resolvable errors have been corrected. A log similar to the preparation form (Figure 16 in Chapter 4) should be kept to record any outstanding problems that were awaiting resolution. As with the preparations, however, the entire verification process does not need to be halted while these problems are being investigated. The additional revisions can be made at a later date when the corrections become available.

## STEP 5: INTERRECORD CHECKS

After the data entry errors discovered by the range/rule checks have been corrected, the next category of verifications, the interrecord checks, are performed on the data. *Interrecord checks* involve comparisons between values from different records but *within* the same data file. Since SPSS Data Entry does not have the ability to compare records, these checks must be implemented with an SPSS/PC+ program.

Two examples of interrecord checks are presented below, again using the crop purchase questionnaire to illustrate. As with the range and rule checks, the interrecord checks are repeated until all data entry errors are corrected. Any remaining problems are recorded in the log (Figure 16) to be resolved later.

## OUT-OF-SEQUENCE RECORDS

Recall that when the questionnaires are organized into batches they are sorted by their key variables. The data file records should therefore also be in order by these variables. An out-of-sequence record could indicate that an observation is missing or that a value for a key variable was entered incorrectly.

The crop purchase data records in Figure 17 should be in order by village, household, the interview date, and the transaction number. The program below lists any records that are out of sequence according to these variables.

```
*****
**
** This program checks crop purchase data **
** for out of sequence records.          **
**                                         **
*****.
```

```
*****
** Convert date to single number **
** using YRMODA() function        **
*****.
```

```
COMPUTE DATE = YRMODA(YR,MO,DY) .
```

```

*****
** Flag records that are out of sequence **
**   by VIL, HH, DATE, or TRANNO.   **
*****

COMPUTE FLAG = 0.

IF (VIL = LAG(VIL) & HH < LAG(HH)) FLAG = 1.

IF (VIL = LAG(VIL) & HH = LAG(HH) & DATE < LAG(DATE))
FLAG = 1.

IF (VIL = LAG(VIL) & HH = LAG(HH) & DATE = LAG(DATE) &
    TRANNO <= LAG(TRANNO)) FLAG = 1.

*****
** Print out of sequence lines **
*****

TITLE "OUT OF SEQUENCE RECORDS".

PROCESS IF (FLAG = 1).

LIST VIL HH MO DY YR TRANNO CROP.

TITLE.

```

Figure 17—Sample crop purchase data

VIL	HH	MO	DY	YR	TRANNO	CROP	. . .
1	4	6	5	89	1	101	. . .
1	4	6	5	89	2	107	. . .
1	4	6	5	89	3	102	. . .
1	3	2	5	89	1	101	. . .
1	5	2	5	89	2	106	. . .
.	.	.	.	.	.	.	. . .
.	.	.	.	.	.	.	. . .
.	.	.	.	.	.	.	. . .

In the program output (Figure 18), the fourth record in the sample data (Figure 17) is listed because its household ID (HH=3) is less than the household ID of the previous record (HH=4). At first glance, it would appear that this is a data entry error and that the actual household ID for this record should be 5. This would have to be verified by examining the questionnaire, however.

Figure 18—Output from out-of-sequence record program

7/30/92		OUT OF SEQUENCE RECORDS					Page 1
VIL	HH	MO	DY	YR	TRANNO	CROP	
1	3	2	5	89	1	101	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	

Notice the use of the SPSS/PC+ YRMODA function in the program to convert the interview date to a single number. Given a year, month, and day, the YRMODA function returns the number of days between this date and October 15, 1582 (the first day of the Gregorian calendar). Therefore, YRMODA(90,12,4) yields the value 149,070, the number of days between December 4, 1990, and October 15, 1582. The YRMODA function makes it easier to compare two dates and to calculate the number of days between them. For example, if DATE1=YRMODA(90,1,17) and DATE2=YRMODA(89,7,8), then DATE1 will be greater than DATE2, since 1-17-90 is later than 7-8-89, and DATE1 minus DATE2 will be the number of days between these two dates.

## DOUBLE RECORDS

Another example of an interrecord check is a search for *double records*, which occur when the same record on a questionnaire is entered twice. Since key variables uniquely identify each record, it is sufficient to examine the key variables to detect double records. The following SPSS/PC+ program compares the key variables in a crop purchase data file (Figure 19) and lists any records that have the same set of values for these variables:

```
*****
**                                     **
** This program checks crop purchase data **
** for double records.                 **
**                                     **
*****.

*****
** Convert date to single number **
** using YRMODA() function         **
*****.

COMPUTE DATE = YRMODA(YR,MO,DY).

*****
** Sort by key variables **
*****.
```

```

SORT BY VIL HH DATE TRANNO.

*****
** Set FLAG = 1 for double records **
*****

COMPUTE FLAG = 0.

IF (VIL = LAG(VIL) & HH = LAG(HH) & DATE = LAG(DATE) &
    TRANNO = LAG(TRANNO))
    FLAG = 1.

*****
** List double records **
*****

TITLE "DOUBLE RECORDS".

PROCESS IF (FLAG = 1).

LIST VIL HH MO DY YR TRANNO CROP.

TITLE.

```

Figure 19—Sample crop purchase data

VIL	HH	MO	DY	YR	TRANNO	CROP	. . .
4	10	8	12	89	1	101	. . .
4	10	8	12	89	2	102	. . .
4	10	8	12	89	3	109	. . .
4	11	9	12	89	1	109	. . .
4	11	9	12	89	2	101	. . .
4	11	9	12	89	2	102	. . .
.	.	.	.	.	.	.	. . .
.	.	.	.	.	.	.	. . .
.	.	.	.	.	.	.	. . .

The double record program first sorts the file by the key variables. This places duplicate records adjacent to each other even if they were originally entered in different places in the file. In the example, records 5 and 6 have the same key variable values and so are listed by the program (Figure 20).

Figure 20—Output from double record program

7/30/92		DOUBLE RECORDS					Page 1
VIL	HH	MO	DY	YR	TRANNO	CROP	
4	11	9	12	89	2	102	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	

### STEP 6: INTERFILE CHECKS

This category of verification involves comparing information from different data files. For example, in the household census data (Figure 21), the same set of households should be represented in the files for both households and persons.

Figure 21—Sample household census questionnaire data

File: CEN01H.SYS (Household level)

VIL	HH	MO	DY	YR	CH	. . .
1	2	7	15	88	N	. . .
1	3	7	15	88	N	. . .
1	4	7	16	88	N	. . .
.	.	.	.	.	.	. . .
.	.	.	.	.	.	. . .
.	.	.	.	.	.	. . .

File: CEN01P.SYS (Person level)

VIL	HH	NOPER	NAME	SEX	. . .
1	1	1	Djibi	M	. . .
1	1	2	Daba	F	. . .
1	2	1	Abdoulaye	M	. . .
1	2	2	Keita	F	. . .
1	2	3	Astou	F	. . .
1	4	3	Ousmane	M	. . .
.	.	.	.	.	. . .
.	.	.	.	.	. . .
.	.	.	.	.	. . .

The program given below compares these files for the same batch of census questionnaires to make sure that the information is entered on both parts.

```

*****
**
** This program compares household and person **
** level files for the household census data and **
** lists households where one level is missing. **
**
*****.

*****
** Files must first be aggregated so that there **
** is a single record for each interview. **
*****.

GET FILE 'CEN01H.SYS'.

SORT BY VIL HH.

COMPUTE FILE_A = 1.

SAVE FILE 'TEMPA.SYS' /KEEP VIL HH FILE_A.

GET FILE 'CEN01P.SYS'.

AGGREGATE OUTFILE *
  /BREAK VIL HH
  /N = NU.

COMPUTE FILE_B = 1.

SAVE FILE 'TEMPB.SYS' /KEEP VIL HH FILE_B.

*****
** Join together files A and B **
*****.

JOIN MATCH
  /FILE 'TEMPA.SYS'
  /FILE 'TEMPB.SYS'
  /BY VIL HH.

FORMAT FILE_A FILE_B (F1.0).

*****
** Variable FLAG indicates type of error **
*****.

COMPUTE FLAG = 0.
IF (SYSMIS(FILE_A)) FLAG = 1.
IF (SYSMIS(FILE_B)) FLAG = 2.

VALUE LABEL FLAG
  1 'HH LEVEL DATA MISSING'
  2 'PERSON LEVEL DATA MISSING'.

```

```
*****
** Print results **
*****.
```

```
TITLE "Compare Household & Person Level Data".
```

```
PROCESS IF (FLAG > 0).
```

```
REPORT
```

```
  FORMAT AUTOMATIC LIST
  /VAR VIL HH FLAG (LABEL).
```

```
TITLE.
```

If a household is missing data on persons (as is household #3) or if there are data on persons but no corresponding household-level data (as for household #1), then the household is listed in the output (Figure 22).

**Figure 22—Output from comparison of files on households and persons**

8/2/92 Compare Household & Person Level Data Page 1		
<u>VIL</u>	<u>HH</u>	<u>FLAG</u>
1	1	HH LEVEL DATA MISSING
1	3	PERSON LEVEL DATA MISSING
.	.	.
.	.	.
.	.	.

## STEP 7: VISUAL INSPECTION OF ENTERED DATA

With all the computer checks, one might think there would be no need to check the data manually. Unfortunately, there are some data entry errors that are not easy for the computer to detect. For instance, the computer is not able to tell, in general, if an entire record of data was not entered. Also, although the computer can verify that a given variable has a value that is within an acceptable range, that does not mean that the value was entered correctly. Therefore, a *visual inspection*, a record-by-record comparison of the data entered into the file with the data on the questionnaires, should be carried out after the computer checks are completed.

To perform the visual inspection, the data in the file are printed double-spaced so that there is room to write in corrections. The following program, using the SPSS/PC+ REPORT command, asks for a file to be printed double-spaced.

```

*****
**
** This program prints the household-level records **
** of the census questionnaire data for the visual **
** verification step. **
**
** The LIST(1) specification in the REPORT command **
** causes the listing to be double spaced. **
**
*****.

SET PRINTER ON /EJECT ON /LENGTH 62.

REPORT
  FORMAT AUTOMATIC LIST(1)
  /STRING
    DATE (MO '/' DY '/' YR)
  /VARIABLES
    VIL HH DATE CH AT DISTANCE ETHN
  /BREAK (NOBREAK)
  /CTITLE '**** DATA LISTING: QUEST. #01 (H) ****'
  /LTITLE ')DATE'
  /RTITLE 'PAGE )PAGE'
.

SET PRINTER OFF /EJECT OFF /LENGTH 24.

```

The person checking the data compares each data record on the printout with the corresponding entries on the questionnaires. Corrections are made first on the listing. Once all the records have been verified against the questionnaires, the revisions are then made to the data file.

An alternative to visual inspection is a procedure called *double entry*, which basically involves entering all data files twice. In SPSS Data Entry this can be accomplished by setting the display mode of the attribute (the nonkey) variables in a file to "verify" once all of the data have been entered. (The display mode is set in the Dictionary Branch of SPSS Data Entry.) When the operator returns to the data entry form, the values of the attribute variables will be hidden. The operator reenters all of the data, using the key variables as a guide. If a value entered during this second pass does not match the original value, the program signals the operator. A window showing both the original and the new entries appears, allowing the operator to choose the correct one.

Since the double entry method does not involve printing out the entire data file, it may be practical in situations where paper is expensive or scarce. If availability of computers is a constraint, however, the visual inspection may be preferred as one person can be entering data at the computer while another is doing a visual inspection. Regardless of the method used, it is best to have someone other than the person who originally entered the data perform the visual inspection or the double entry.

The visual inspection or double entry should only be done after all corrections have been made to the questionnaires. That is, once all problems uncovered during the questionnaire preparation and computer verifications have been resolved and the questionnaires

## **STEP 8: SUPPLEMENTARY VERIFICATIONS**

are corrected. Otherwise, the procedure will have to be repeated if the questionnaire data are modified later.

All cleaning reports, the results of verification programs, and data file listings should be stored with the batch of questionnaires. Once the verifications have been completed through the visual inspection, the batch should be examined by the researcher (or someone that the researcher approves) to judge the quality of the work. If he or she is not satisfied with the quality of the data cleaning, new verifications should be carried out.

The new verifications could involve repeating all the checks or only some of them. Because this stage of the verification process is not as structured as the first round of verifications, a different control form (Figure 23) is used to keep track of the work done on the batch. Any new verifications performed or any corrections made to the questionnaires or the data files are recorded on this form, along with the date and the initials of the person who carried out the task.

When all pending problems with the batch have been resolved, and when the researcher is satisfied with the quality of the data, the files may be certified as ready for use. A special backup copy should be made of the "clean" data files, and the batch should be filed away separately from the questionnaires that are still being entered and cleaned.

## **SUMMARY**

This chapter focused on a variety of data verifications that can be carried out to ensure that the data are correctly entered and contain no erroneous information. The first type of verifications are ranges—specifications of legitimate values for each variable. Another set of verifications are rules, which allow the comparison of different variables within a given record. Both ranges and rules are implemented in SPSS Data Entry.

More complicated verifications can be carried out with SPSS/PC+ programs. Interrecord checks compare different records within a single data file. Two examples of interrecord checks include identifying out of sequence or double records. Finally, there are interfile checks, which involve comparing data between different files. An example of an interfile check would be verifying that the different level files for a batch of questionnaires contain information on the same set of households.

After the computer checks have been completed and all outstanding data problems have been resolved, a visual verification is carried out to uncover data entry errors that could not be detected with the computer verifications. Alternatively, the double entry method may be used. Additional verifications may be required by the researcher if he or she has doubts about the quality of the data.



## 6 USING PREVIOUSLY COLLECTED DATA

As was mentioned in the introduction, one of the advantages of concurrent data entry is that one can use previously collected data to guide subsequent data collection. In this chapter an example will be presented to illustrate this technique.

Suppose that a study includes a consumption questionnaire that records the quantities of ingredients used to prepare a household's meals for the previous day. These measurements may be given in units that are household-specific, such as "one small bowl of millet" or "two coffee cans of rice." In order to calculate the household's calorie consumption, these distinct household measures need to be weighed to determine their kilogram equivalents. This can be a daunting task if there are a large number of units.

The following is an excerpt from the data file created from the first batch of the consumption data.<sup>9</sup> The data give the quantities of the various ingredients used to prepare a household's meals for the 24-hour period prior to the interview. The portion of the file shown in Figure 24 is for an interview that took place on 12-4-89 for household 1 in village 1.

The data in file CON01I.SYS can be used to create a list of all the unit measures referred to by households in the first batch of consumption questionnaires, along with the products measured by these units. The program for doing this is as follows.

Figure 24—Sample file of consumption quantities

File: CON01I.SYS									
VIL	HH	MO	DY	YR	MEAL	PRODUCT	FORM	QUANTITY	UNIT
1	1	12	4	89	1	millet	threshed	1.0	lg.bowl
1	1	12	4	89	2	rice	threshed	2.0	lg.bowl
1	1	12	4	89	2	cowpeas	shelled	3.0	scoop
1	1	12	4	89	2	okra	fresh	1.0	tomato can

<sup>9</sup> To facilitate the reader's understanding of the data, the labels "millet," "large bowl," "threshed," etc. have been substituted for what would actually be numeric codes in the variables PRODUCT, UNIT, and FORM.

```

*****
** SPSS Program for listing unit weights **
** and creating a unit weight file. **
*****

GET FILE 'CON01I.SYS'.

AGGREGATE OUTFILE=*
  /BREAK VIL HH UNIT PRODUCT FORM
  /N = NU.

** The next two commands create a missing **
** unit weight variable. **

COMPUTE UNITWT = 0.
RECODE UNITWT (0 = SYSMIS).

SAVE FILE 'CONUNITS.SYS'.

** Create listing of unit weights **.

REPORT
  FORMAT AUTOMATIC LIST(1) UNDERSCORE(ON)
  /VARIABLES
    PRODUCT FORM UNITWT
  /BREAK VIL (PAGE)
  /BREAK HH
  /BREAK UNIT
  /CTITLE '**** CONSUMPTION UNIT WEIGHTS ****'
  /LTITLE ')DATE'
  /RTITLE 'PAGE )PAGE'
  /OUTFILE 'CONUNITS.LIS'.

```

The program places the listing in file CONUNITS.LIS. An excerpt from the program output is given in Figure 25. The program also creates an SPSS data set called CONUNITS.SYS, which contains the same information as the listing. Once the unit weight information has been collected, the data can be entered directly into CONUNITS.SYS.

**Figure 25—Output from the program listing consumption unit measures (CONUNITS.LIS)**

20 Feb 1990 **** CONSUMPTION UNIT WEIGHTS **** PAGE 1					
<u>VIL</u>	<u>HH</u>	<u>UNIT</u>	<u>PRODUCT</u>	<u>FORM</u>	<u>UNITWT</u>
01	01	lg.bowl	millet	threshed	.
			rice	threshed	.
		scoop	cowpeas	shelled	.
		tomato can	okra	fresh	.

The enumerators visit each household, using the list to identify the unit, product, and form combinations to be weighed. The weights are recorded directly on the printout, and the completed unit weight lists are returned to the office to be entered into the CONUNITS.SYS file. The unit weight file can then be matched with the original consumption data (Figure 24) to fill in the unit weights for each ingredient. The following program illustrates this procedure, matching CONUNITS.SYS to CON01I.SYS.

```
*****
** SPSS program to add unit weights to batch 01 file **
** CON01I.SYS once all unit weights have been      **
** entered into CONUNITS.SYS                       **
*****

GET FILE 'CON01I.SYS'.

SORT BY VIL HH UNIT PRODUCT FORM.

JOIN MATCH
  /TABLE 'CONUNITS.SYS'
  /FILE *
  /BY VIL HH UNIT PRODUCT FORM.
```

Once appropriate calorie coefficients are added to the consumption file, the daily caloric intake of the household can be computed from these data.

## **SUMMARY**

This chapter presented an example of how previously entered data can be used to guide subsequent data collection activities. In the household consumption questionnaire, data on unit measures used in meal preparation are compiled into a list that can be brought into the field. This listing provides the enumerators with a convenient means for recording the weights of the unit measures. The completed list of unit weights can then be entered into a data file and combined with the original consumption data in order to evaluate the weights of the different ingredients used to prepare a household's meals.

# 7 DATA FILE MANAGEMENT

Proper organization and management of data files on the PC is vital to smooth coordination of data entry and processing. This chapter describes several important issues concerning data file management, including using DOS's directory structure to organize files on the hard disk, proper written documentation of data files, and, perhaps most important, implementation of a backup system to protect against loss of data files.

## ORGANIZING DATA FILES ON THE HARD DISK

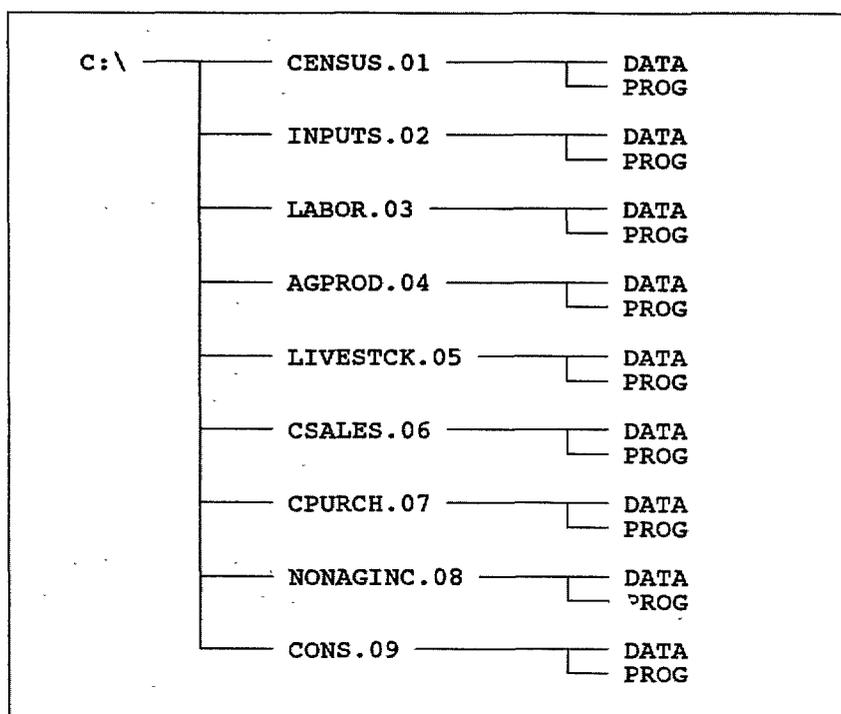
If the data files are being stored on a PC's hard disk, it is a good idea to keep the files for different types of questionnaires in separate subdirectories, rather than having all files in the root (or "\") directory. For example, all of the crop purchase data files should be kept in one subdirectory, the census data files in another subdirectory, and so forth. This type of structure makes it easier to find a specific data file. Furthermore, there will be less chance of confusing files from one type of questionnaire with files from another.

Directory names are under the same restrictions as DOS file names. They may consist of no more than eight letters or numbers followed by an optional extension of no more than three characters. As with files, one should choose names for the questionnaire directories that are mnemonic--either an abbreviation of the questionnaire name (\CPURCH), or the questionnaire ID number (\Q07), or both (\CPURCH.07). The advantage of the last format is that the directories can be sorted by their extension (using Norton Utilities DS command, for instance), thus putting them in order by the questionnaire ID numbers.

In addition to separating each questionnaire type, the questionnaire directories may further be divided into subdirectories for data files and programs. For example, the data files and programs for the crop purchase questionnaire would be found in the \CPURCH.07\DATA and \CPURCH.07\PROG directories, respectively. Separating data and program files in this way may be useful if there are a large number of files. A good rule of thumb is to have no more than 100 files in a directory.

Figure 26 shows a sample directory structure for organizing several different questionnaires. Note the additional subdirectories for programs and data. Parallel structure is very important in a directory scheme; the directories for each questionnaire should be arranged similarly. This will make it easier for someone not completely familiar with the data to find the information he or she needs.

Figure 26—Directory tree structure



## DOCUMENTING DATA FILES

It is extremely important to have good documentation of all data files. If there are several different types of questionnaires, it will be difficult to remember the meaning of all the different variables that have been created in every file. Documentation should be made not only for the files of directly entered data, but also for data files created by manipulating and combining other data files. The documentation should also include comments on the file, such as how to use or interpret certain variables. This type of documentation is useful when writing programs and preparing the analysis.

Figure 27 suggests a format for documenting data files. The header contains places for recording the name and a brief description of the file. The description should include the name of the questionnaire and the level of the data. The next space is for the file format, such as ASCII, SPSS/PC+, or dBase IV. In order to help locate the file more quickly, the full directory location is also included. For files that were created by manipulating other files, the name of the program that carried out this transformation should be noted. Finally, the header gives the number of variables in the file.

Below the header appears a list of all the variables in the file. There are columns for the name of the variable, its width, type (numeric or string), and description. If the variable takes a code as a value, a list of the codes or a reference to a code set documented

elsewhere should be included. For variables that are expressed in some kind of unit of measure, such as kilograms or dollars, this information is noted.

At the bottom of the form is a place for comments about the file. This is for indicating anything special that may not be obvious from the file and variable descriptions, such as how certain variables should be used. For example, "To calculate the total amount paid, multiply PRICE x QUANTITY."

The variables in SPSS/PC+ data files may be labeled either in SPSS Data Entry or with the SPSS/PC+ command VARIABLE LABEL. The advantage of adding labels to the data files is that they are saved permanently with the data set and will make it easier for someone not familiar with the files to interpret the data. In addition, if the variables have been labeled, a listing similar to the documentation form in Figure 27 can be produced automatically by the SYS INFO command.

Figure 27—Data file documentation format

```

                                Data File Documentation
*****
File Name:  CEN01H.SYS
Description: Questionnaire #01--Household Census
              Household-level data
File format: SPSS/PC+
Location:   C:\CENSUS.01\DATA
Creating program(s):
No. of Variables:  10
*****

                                Layout

Variable   Width   Type      Description
-----
VIL        3       Numeric   Village
HH         2       Numeric   Household identification number

MO         2       Numeric   Month of interview
DY         2       Numeric   Day of interview
YR         2       Numeric   Year of interview

CH         1       String    Y = Household head is village chief
AT         1       String    Y = Household uses animal traction
DISTANCE   4       Numeric   Distance from household to main road (meters)
ETHN       1       Numeric   Principal ethnic group of household
              (code set #1.1)

QUESTNO    3       Numeric   Questionnaire number

Comments:

```

## PREPARING BACKUPS

Preparing regular backups of the data files onto floppy disks or magnetic tape is another extremely important element of data file management. Backups help protect against loss of data files from hardware failure or accidental erasure. Tapes have the advantage of capacity—a quarter-inch DC-2000 mini-cartridge can hold 150 megabytes of data, while a 4-millimeter digital audio tape (DAT) cartridge can hold 2.5 gigabytes of data (1 gigabyte = 1,000 megabytes). Tape drive systems can be expensive, however. Fortunately, there now exist a number of impressive software packages for backing up files onto diskettes. These are quite fast and can compress files to "squeeze" more data onto a diskette.<sup>10</sup> Of course, there are also the DOS BACKUP and RESTORE commands, but these are rather slow and perform only minimal file compression.

The principles in a backup system are the same, however, regardless of the medium used. A full-scale backup of all data and program files on the hard disk should be performed every two weeks. At least two sets of disks or tapes are used for these complete backups, and they are reused on a rotating basis. For example, one set of disks could be for the beginning of the month, and another for the 15th of the month. The backup sets need to be rotated in this manner so that one backup is not done over the most recent backup. If something were to go wrong with the current backup and the hard disk became damaged, the previous backup set would still be intact.

In between the biweekly backups, a daily backup should be carried out using a separate set of rotating diskettes or tapes. For example, one could use 10 diskettes labeled "Monday #1," "Tuesday #1," "Wednesday #1," . . . "Friday #1," "Monday #2," "Tuesday #2," . . . "Friday #2." At the end of each day, only those files that have been modified since the last daily or biweekly backup would be copied onto the diskette. In this way, the last 10 days worth of file changes would be available, if needed.

DOS employs a special flag, called the *archive attribute*, to indicate whether or not a file has been backed up. Whenever a file is modified, DOS "sets" its archive attribute to mark the file for backup. If the archive attribute is not set (or "reset"), then the file does not require backing up. Most backup programs can read and modify a file's archive attribute, allowing the user to control which files are backed up.

For the complete biweekly backups, the program should copy all files, regardless of whether or not the archive attribute is set, and then reset the attribute for every file. For the daily backups, however, the program should backup only those files that have their archive attributes set (indicating that they were modified since the last backup was performed), and it should then reset the attribute for each of these files.

---

<sup>10</sup> See Mendelson (1991) for a review of backup programs.

In addition to the biweekly and daily backups described above, it would be prudent to make additional complete backups, which would be stored off site in a secure location. This will protect against theft or damage from fire. These additional backups do not need to be made as frequently as the biweekly backups (they could be done once every three months, for instance).

## **SUMMARY**

Three important topics in data file management were discussed in this chapter. The first involved organizing data files into subdirectories in order to facilitate access to the files. Files should be separated by type of questionnaire and perhaps further divided into subdirectories for programs and data. The second issue concerned the importance of good documentation of data files. Proper documentation will allow the researcher to keep track of crucial details concerning the data, and permit other users to be able to access the information more easily. Finally, the necessity of a backup system was emphasized and a system involving biweekly and daily backups was presented. In addition to the regular backups, special backups should also be made periodically and stored off site.

## **APPENDIX: SOFTWARE CONSIDERATIONS**

As an additional topic, some of the functions that are desirable in computer software used to implement a data entry system are presented in this appendix. The programs are divided into three categories: spreadsheets, data analysis packages, and database programs.<sup>11</sup> In making choices about software, the researcher needs to consider not only the capabilities of the program, but also the skill level of the personnel who will be operating it. It is better to have less powerful but user-friendly software than to rely on a more sophisticated package that few are capable of using effectively.

Figure 28 summarizes the capabilities and features of different software packages that may be used for data entry. For each package, the table shows the maximum number of variables and records allowed in a single data file, as well as the number of data files that the program is capable of accessing simultaneously during data entry. The presence of special data entry features—user-designed data entry screens, automatic computed variables, and the insertion of records in the middle of a file—as well as data verification capabilities—ranges, rules, interrecord checks, and interfile checks—are also indicated. Some packages also provide for user-defined applications (programs for automatically executing a series of commands) and custom report generation. In addition, the standard data file formats that the package is capable of reading and writing are listed.

### **SPREAD- SHEETS**

Spreadsheet programs, such as Lotus 1-2-3, are widely used and so it is usually possible to find people who have some experience with them. Although they can be used for data entry, spreadsheets lack some of the helpful features available in data entry programs. For instance, all data must be entered in a tabular format (rows and columns); there is no provision for designing custom data entry screens. In addition, there is no automatic verification of variable ranges at the time of data entry. While many of these features can be simulated by using *macros* (the saving of a sequence of key strokes that may later be replayed), writing such macros can be a daunting and unwieldy task. Because of these limitations, spreadsheets are probably best used only for small data entry needs.

---

<sup>11</sup> The mention of specific programs is for illustrative purposes only and should not be interpreted as an endorsement of these packages by the author.

Figure 28—Summary of features for data entry packages

Features	Lotus 123 (v2.01)	SPSS DE (v4.0)	SAS FSP (v6)	dBase IV (v1.1)	FoxPro (v1.02)	Paradox (v3.5)	ISSA (v2.28+)
Maximum variables per record	256	500	No limit <sup>a</sup>	255	255	255	940 <sup>b</sup>
Maximum records per file <sup>c</sup>	8192	No limit	2 billion	2 billion	1 billion	1 billion	No limit
Maximum Number files at same time	1	1	1	8	25	24	n.a.
Custom data entry screens	No	Yes	Yes	Yes	Yes	Yes	Yes
Automatic computed variables	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Record insertion	Yes	No	No	Yes	Yes	Yes	No
Ranges	No	Yes	Yes	Yes	Yes	Yes	Yes
Rules	No	Yes	Yes	Yes	Yes	Yes	Yes
Interrecord checks	No	No <sup>d</sup>	No <sup>e</sup>	Yes	Yes	Yes	Yes
Interfile checks	No	No <sup>d</sup>	No <sup>e</sup>	Yes	Yes <sup>f</sup>	Yes <sup>f</sup>	Yes <sup>f</sup>
User defined applications	No	No	No	Yes	Yes	Yes	Yes
Custom reports	No	No <sup>d</sup>	No <sup>e</sup>	Yes	Yes	Yes	Yes
Data file formats <sup>g</sup>	•Lotus •dBase •DIF •ASCII	•SPSS •dBase •Lotus •ASCII	•SAS •dBase •DIF •ASCII	•dBase •Lotus •ASCII	•dBase •DIF •ASCII	•dBase •Lotus •ASCII	•ASCII •SPSS <sup>h</sup>

Sources: Lotus Development Corp., SPSS Inc., SAS Inc., PC Magazine (May 28, 1991), IRD/MACRO.

Notes: n.a. is not applicable.

Lotus 1-2-3 is a product of Lotus Development Corp., 55 Cambridge Parkway, Cambridge, MA, U.S.A. 02142.

SPSS Data Entry and SPSS/PC+ are products of SPSS Inc., 444 N. Michigan Ave., Chicago, IL, U.S.A. 60611. Tel: 312-329-3500.

SAS PC and SAS FSP are products of SAS Institute Inc., Box 8000, Cary, NC, U.S.A. 27511-8000. Tel: 919-677-8008.

(continued)

**Figure 28—continued**

dBase III and dBase IV are products of Ashton-Tate, 20101 Hamilton Ave., Torrance, CA, U.S.A. 90509. Tel: 213-329-9989.

FoxPro is a product of Fox Software Inc., 134 W. South Boundary, Perrysburg, OH, U.S.A. 43551. Tel: 419-874-0162.

Paradox is a product of Borland International Inc., 1800 Green Hills Rd., P.O. Box 660001, Scotts Valley, CA, U.S.A. 95067. Tel: 408-438-5300.

Integrated System for Survey Analysis (ISSA) is a product of IRD/MACRO, 8850 Stanford Blvd., Suite 4000, Columbia, MD, U.S.A. 21045. Tel: 301-290-2800.

<sup>a</sup> Theoretical limit of program. Actual number of variables is limited by the amount of memory available.

<sup>b</sup> For ISSA, maximum number of variables available per questionnaire.

<sup>c</sup> Theoretical limit of program. Actual number of records is limited by the amount of disk space available.

<sup>d</sup> Feature can be implemented with SPSS/PC+ data analysis program.

<sup>e</sup> Feature can be implemented with SAS PC data analysis program.

<sup>f</sup> Table lookup verification is available at the time of data entry.

<sup>g</sup> Not all formats are shown. DIF = Data Interchange Format.

<sup>h</sup> ISSA does not actually create an SPSS file, but will produce a DATA LIST command that can be used to read an ASCII file into SPSS/PC+.

## **DATA ANALYSIS PACKAGES**

SPSS/PC+ and SAS PC are popular data analysis programs for IBM-compatible PCs that offer a wide variety of data manipulation and statistical analysis capabilities. With each of these programs, one can also obtain a data entry module (SPSS Data Entry and SAS FSP, respectively), which creates data files that can be read directly by the data analysis functions in the package. This offers the great convenience of being able to go immediately from data entry to data analysis without needing to convert the files to a different format.

Both the SPSS and SAS programs allow the user to draw custom data entry forms by placing text and data entry fields directly on the screen. These forms can then be used to enter the data, so that instead of typing into a spreadsheet display, the operator sees a form that closely resembles the actual questionnaire. In addition, it is easy to specify ranges for each variable that alert the operator if he or she enters an invalid value. Both programs also allow the user to define cleaning rules to compare different variables and to fill in values for nonentered variables automatically.

Although these programs are rather easy to use, even for non-programmers, they are unfortunately somewhat limited in their capabilities. For example, neither SPSS Data Entry nor SAS FSP allows the insertion of a new record into the middle of a file. So, if a record was omitted during the initial data entry, it must first be appended to the end of the file, and then the file has to be sorted to put the record in the correct position. In addition, more complicated interrecord and interfile checks can not be carried out using the data entry programs, but only with the data manipulation com-

mands available in the main SPSS or SAS packages. This is less convenient than being able to perform these tasks directly from the data entry program.

## **DATABASE PROGRAMS**

Like the spreadsheets, database packages such as dBase IV, FoxPro, and Paradox are also widely used and familiar to a large number of people. These programs are quite powerful and very well suited to data entry tasks. In fact, there are relatively few limits on what can be accomplished with database packages. Unfortunately, most databases require programming skills in order to access all of their features—skills that might not be available in the field.

Most database packages have the ability to create custom data entry screens, usually by positioning text and fields using the cursor (or arrow) keys. Some permit automatic verification of variable ranges to prevent entering incorrect data. The great advantage of these programs, however, is their ability to create a wide variety of custom reports. In this way, separate reports can be produced showing different types of errors. This is not possible with SPSS Data Entry or SAS FSP.

With database packages it is also possible to work with several data files at the same time. This means that different levels of data on a questionnaire can be entered into different files during one data entry pass. In addition, reports and tables drawing on data from different files can be created, and interfile checks (called *table-lookup* in database parlance) can be carried out during the data entry process.

Because most database packages come with full programming languages, one can create what are called "custom applications," that is, user-defined programs that can automate most of the data entry and verification procedures. This not only makes the system easier to use, but also reduces errors caused by an operator typing an incorrect command or file name.

For older database packages, such as dBase III, using the advanced features required someone with solid programming experience. Many of the newer packages, however, have come a long way in making their more advanced features accessible to nonprogrammers. In a review of Paradox, for example, *PC Magazine* says, "if you need to link two or more [files] . . . simply place matching values in the columns you want matched, and the program does the rest. There is no need for complex programming, predefined relations, or complicated indexing of fields in advance" (cited in Kalman 1991, 181).

Another program, Integrated System for Survey Analysis (ISSA), was developed by IRD/MACRO for the Demographic and Health Surveys Program, sponsored by the U.S. Agency for International Development. It is designed specifically to work with complex survey data. ISSA has extensive capabilities regarding data verifi-

cation, file manipulation, tabulation, and report production.<sup>12</sup>

A disadvantage with most database packages is that they are equipped with only the most basic statistical functions, such as means and standard deviations. They do not contain more sophisticated procedures such as linear regression or cluster analysis. To carry out these types of analyses, the user must either write programs in the database's programming language or transfer the data to a statistical analysis package like SPSS or SAS.

## SUMMARY

Three different types of software for performing data entry and verifications were presented in this section. When choosing a program, the researcher should balance the power of the software with its ease of use. Spreadsheets may be useful for small-scale data entry needs but lack the features needed to handle complicated data collection systems. The data entry programs that are available for the SPSS and SAS statistical packages are easy to use and have the advantage that the data files they create can be used directly by the data manipulation and statistics procedures. Although they are simple to use, they suffer from a limitation of capabilities. Finally, database programs are versatile but may require complicated programming to make full use of their features. Some of the newer packages, however, are designed to be more accessible to nonprogrammers.

---

<sup>12</sup> ISSA differs from the other data entry programs discussed in that it creates *hierarchical*, rather than rectangular, data files. This is a method of storing multilevel data in a single data file without wasted space or repeated information. Because of its different file structure, the maximum number of variables shown for ISSA in Figure 28 is not directly comparable with the other programs. For ISSA, this limit refers to the number of variables allowed per *questionnaire*, whereas for the other packages it indicates the number of variables permitted in each *data file*. For example, a household census questionnaire entered in SPSS Data Entry would be limited to 500 variables for the household-level data and 500 variables for the person-level data. In ISSA, however, the questionnaire would be limited to 940 variables for both household and person data items *combined*.

-62.

## GLOSSARY

<i>aggregation</i>	The process of combining into a single record all data file records with the same values for a subset of <i>key variables</i> . Normally, this process involves simultaneously calculating statistics on one or more <i>attribute variables</i> .
<i>alpha-numeric variable</i>	Another term for a <i>string variable</i> —one that can store both letters and numbers.
<i>archive attribute</i>	A special flag employed by DOS to indicate whether or not a file has been backed up. If the archive attribute for a file is <i>set</i> , then the file has been modified since the last backup. If the archive attribute is <i>reset</i> , then the file has <i>not</i> been modified since the last backup.
<i>attribute variables</i>	Variables that contain descriptive, rather than identifying, information; variables that are not key variables.
<i>backup</i>	(1) The process of creating reserve copies of computer files. (2) A set of reserve copies of computer files.
<i>batch</i>	A set of questionnaires of a given type that are to be entered and verified together.
<i>body</i>	The part of the questionnaire that records the main body of descriptive data.
<i>codebook</i>	A document containing a description of each data item and a list of all coded responses for that item.
<i>codes</i>	Numeric values used to represent qualitative responses to questionnaire items.
<i>concurrent data entry</i>	The strategy of entering questionnaire data concurrently with the process of data collection.
<i>data cleaning</i>	The process of verifying the quality of the data entered into a file. Data cleaning involves looking for both data entry errors and inconsistent or missing data.
<i>data entry errors</i>	Mistakes made while entering the data from the questionnaire into the computer. Can be identified by comparing data file values with those on the questionnaire.

<i>data file</i>	A named collection of related information stored together in a form accessible to a computer.
<i>database</i>	The complete set of data files for all questionnaires in a survey.
<i>double entry</i>	A method for validating the accuracy of the data entry by reentering all questionnaires a second time and comparing each value to that originally entered.
<i>header</i>	The part of a questionnaire that contains identifying information about the questionnaire data. The header normally appears at the beginning of the questionnaire form.
<i>inconsistent or missing data</i>	An error not related to data entry, but to the absence of data or the logical inconsistency of the information.
<i>interfile check</i>	A data verification involving comparisons between information in different data files.
<i>interrecord check</i>	A data verification involving comparisons between different records within the same data file.
<i>key variables</i>	A set of variables that uniquely identify each record in a data file. Directly corresponds to the file <i>level</i> .
<i>level</i>	The way in which records are classified in a data file; the characteristics that uniquely identify each record in a data file.
<i>missing interview</i>	An indication that there was no information whatsoever for an interview (because, for instance, the respondent could not be located).
<i>missing response</i>	A complete absence of information about a data item (a nonresponse).
<i>missing value</i>	Another term for a <i>missing response</i> . Also, a special value or code designated to represent a missing response.
<i>numeric variable</i>	A variable that is only capable of storing a number as its value.
<i>procedures manual</i>	A document containing a list of instructions covering data entry and verification procedures.
<i>questionnaire</i>	A survey instrument or data collection form designed to be administered as a single unit, which may consist of one or more pages.
<i>questionnaire ID number</i>	A unique number used to identify a questionnaire <i>type</i> . For example, the household census questionnaire has ID number 1, while the crop purchase questionnaire has ID number 7. The ID

	number of a questionnaire should not be confused with the questionnaire numbering.
<i>questionnaire numbering</i>	The unique, sequential numbering of questionnaires in a batch.
<i>range</i>	A list of all values that are valid for a particular variable.
<i>record</i>	A related set of values for a collection of variables, sometimes also referred to as a case. In a rectangular file, records are represented as rows.
<i>rectangular file</i>	A particular method of organizing information in a data file, consisting of a collection of records that contain information on an identical set of variables. A rectangular file can also be thought of as a two-dimensional table, with rows (records) and columns (variables).
<i>relational database model</i>	The method of organizing and relating a <i>database</i> of rectangular files, based on levels and key variables.
<i>rule</i>	A statement specifying a relationship between different variables within a given file record.
<i>string variable</i>	A variable that can store both letters and numbers.
<i>system missing value</i>	A special value used by data entry or analysis software (such as SPSS/PC+) to indicate a missing response.
<i>table-lookup</i>	A database term for the process of obtaining data from other data files during data entry. Similar to an interfile check.
<i>user missing value</i>	In SPSS/PC+, a numeric value designated by the user to represent a missing response. User missing values are treated like the system missing value for computations.
<i>variable</i>	The basic types or pieces of information that are contained in a data file. In a rectangular file, variables are represented as columns.
<i>visual inspection</i>	A manual record-by-record comparison of the information entered into a data file with the information recorded on the questionnaires.
<i>zero interview</i>	An explicit indication that the activity being investigated on a questionnaire did not take place, and therefore all data items relating to this activity are not applicable for the interview.
<i>zero response</i>	An explicit indication of zero (0) for a data item.

## **BIBLIOGRAPHY**

- Cardenas, Alfonso F. 1979. *Data base management systems*. Boston: Allyn and Bacon.
- Casley, D. J., and D. A. Lury. 1987. *Data collection in developing countries*. 2d ed. Oxford: Clarendon Press.
- Crawford, Eric W., John S. Holtzman, John M. Staatz, Chris Wolf, and Michael T. Weber. 1988. MSU experience in research design and data processing/analysis. Paper presented at the University of Zambia Food Security Research Network Workshop, 24-25 May, Lusaka, Zambia.
- Hadden, Louise, and Mireille Léger. 1990. *Codebook for the American housing survey*. Prepared by Abt Associates, Inc., Cambridge, Mass., U.S.A. under contract to the U.S. Department of Housing and Urban Development.
- Kalman, David. 1991. Fifteen relational databases: Easy access, programming power. *PC Magazine*, May 28. 101-200.
- Martin, James. 1977. *Computer data-base organization*. 2nd ed. Englewood Cliffs, N.J., U.S.A.: Prentice-Hall.
- Mendelson, Edward. 1991. Premium insurance: Backup software gets better. *PC Magazine*, June 11, 103-144.

**INTERNATIONAL  
FOOD  
POLICY  
RESEARCH  
INSTITUTE**

The International Food Policy Research Institute was established in 1975 to identify and analyze alternative national and international strategies and policies for meeting food needs in the world, with particular emphasis on low-income countries and on the poorer groups in those countries. While the research effort is geared to the precise objective of contributing to the reduction of hunger and malnutrition, the factors involved are many and wide-ranging, requiring analysis of underlying processes and extending beyond a narrowly defined food sector. The Institute's research program reflects world-wide interaction with policymakers, administrators, and others concerned with increasing food production and with improving the equity of its distribution. Research results are published and distributed to officials and others concerned with national and international food and agricultural policy.

The Institute receives support as a constituent of the Consultative Group on International Agricultural Research from a number of donors including Australia, Belgium, Canada, the People's Republic of China, the Ford Foundation, France, the Federal Republic of Germany, India, Italy, Japan, the Netherlands, Norway, the Philippines, Spain, Switzerland, the United Kingdom, the United States, and the World Bank. In addition, a number of other governments and institutions contribute funding to special research projects.