AERIAL SURVEY COMPUTER ANALYSIS

PART I

DATA ENTRY REFERENCE GUIDE

JANUARY 1986

RIM

AERIAL SURVEY COMPUTER ANALYSIS

PART I

DATA ENTRY REFERENCE GUIDE

AUGUST 1985

RIM

AERIAL SURVEY COMPUTER ANALYSIS

AUGUST 1985

RIM

RESOURCE INVENTORY AND MANAGEMENT LIMITED

FOR

INTERNATIONAL LIVESTOCK CENTRE FOR AFRICA

AERIAL SURVEY COMPUTER ANALYSIS

PART I DATA ENTRY REFERENCE GUIDE

Software and Documentation by ITAD Ltd

Information Technology and Agricultural Development

AERIAL SURVEY COMPUTER ANALYSIS

PART I DATA ENTRY REFERENCE GUIDE

System Design: John J Bennett and Derek Poate
Programs: John J Bennett
Documentation: Derek Poate

Software and Documentation by ITAD Ltd

Information Technology and Agricultural Development

TABLE OF CONTENTS

page number

SECTION I

INTRODUCTION	1
ABOUT THIS GUIDE	2
GETTING STARTED	3
Program Files	3
Execute the Program	4
USING A MENU DRIVEN SYSTEM	6
What is a Menu Driven System?	6
How to Enter Data	6
New Data	6
Correcting Mistakes	7
Pre-Prompted Characters	7
Editing Data	8
Exit from the Option	8
Primary Menu	8
Secondary Menus	8
From Within a Program	8
PRIMARY MENU OVERVIEW	9
ORGANISING YOUR WORK	11
Operating Procedures	11
Aerial Survey Data Entry Procedure	12
Create a New Data Set	13
Modify and List The Code File	14
Enter and List Data	15
Validate and Edit a Data Set	16
Create a Summary File	17
Create a Transfer File	18
Data File Management	19
Data Security	20
DEFINE SYSTEM PARAMETERS	21
Choose a Data Set	21
Set the Date	21
Assign Files to Disk Drives	21
Hard Disk Systems	22
Floppy Disk Systems	22
Default Settings	22
CODE MAINTENANCE	23
Data Structure	23
Code File	24
Code File Listing	26

DATA ENTRY 27
General 27
A : Summary File Data Entry/Edit 27
B : Summary File Block Data Entry/Edit 28
C : Summary File Validation 28
D : Summary File Listing 29
E : Observation File Data Entry/Edit 29
F : Observation File Validation 31
G : Observation File Listing 33
H : Photo File Data Entry/Edit 34
I : Photo File Validation 34
J : Photo File Listing 35

OBSERVER FILE MAINTENANCE 36
Observer File 36
Observer File Listing 38

BIAS AND SUMMARY FILE CREATION 39
A : Observer Bias Calculations 39
B : Photo to Summary File Transfer 39
C : Observation to Summary File Transfer 40
D : Transfer File Creation 41

FILE RECOVERY PROCEDURES 42

COUNTRY/ZONE SETUP 43

SECTION II

DATA COLLECTION 44
Data Collection 44
Grid Square Coordinates 44
Individual Squares 44
Blocks of Grid Squares 45
Rows or Columns of Grids 46

DATA FILE STRUCTURES 47
Code File 47
Summary Data File 47
Summary Data Transfer File Format 48
Observation Data File 49
Photo Data File 50
Observer File 50
Master Code File 51
Changing Data Structures 52
File Structures 52
Active Variables 52

SECTION III

dBASE II AND MS DOS COMMANDS 53
dBase II 53
Command Line Prompt 53
List a File 53
List a Variable 53
Delete a Record 53
Delete Several Records 54
Copy a File 54
MS DOS 54
Command Line Prompt 54
List a File 54
Delete a File 54
Rename a File 54
Copy a File 55
Directory Listing 55

SECTION IV

PROCEDURE PROMPT CARDS
Aerial Survey Data Entry Procedure
Create a New Data Set
Modify and List the Code File
Enter and List Data
Validate and Edit a Data Set
Create a Summary File
Create a Transfer File

APPENDIX I

Command File Listings

INTRODUCTION

The aerial survey analysis data entry program is designed to permit users of the system to enter survey data, conduct a simple validation, prepare a listing of the data, and create a data file for subsequent analysis and mapping. (See Part II)

The system is designed for use by operators who are familiar with the basic principles of aerial survey data collection, and who have a minimum of computer experience. Complete novices will require some training in the basic techniques of file management. It is recommended that one or two days of supervision by experienced operators should be provided before the package is used in earnest.

To this end, a small specimen data set is provided at the end of this manual, together with all the operations needed to create a data file that can be used with the mapping and analysis package that accompanies these data entry programmes.

This data entry package is a menu driven system - i.e. at each stage of operation, it gives the user a series of alternative courses of action. There are several menus within the package that are nested within each other. Thus to get from one menu to another, it is generally necessary to return to the primary menu (by following the instructions on the screen) before it is possible to select a new series of operating procedures. Some examples of this are given in the pormpt cards at the end of this manual.

The programmes are written in dBase 11, and it is thus capable of transfer to any computer which supports dBase 11. It also means that some limited editing and transformations can be performed from within dBase 11, as well as from within the menu driven package. It is recommended, however, that only those users with revious experience of handling dBase 11 files should use this facility.

The programme is designed spcecifically to enter data that has been collected on a grid basis, either as single observations per grid (for example flying height) or as multiple observations per grid (such as observed livestock herds).
Multiple observations are entered into either the OBSERVATION FILE or the PHOTO FILE, and can be taken directly from the in-flight recording sheets. The OBSERVATION FILE is designed to handle observer counts of livestock, together with the photographs used for bias calculations. The PHOTO FILE is designed to take a series of observations per grid, as interpreted from aerial photographs. Once these have been entered, the records for each grid are summarised into a SUMMARY FILE, which then contains a single summed value for each variable for each grid cell.

GETTING STARTED

Program Files

To run the aerial survey data entry system you need the following files on your computer disks. Section I, Define System Parameters explains which drives they are to be stored on.

The dBase II program files: DO.COM (main program. This may appear as DBASE.COM. It should be renamed DO.COM)

DBASEOVR.COM (overlays and messages)
DBASEMSG.TXT (HELP file, not essential)

The dBase II command files:
RIMGO
RIMHEAD1
RIMHEAD2
RIM
RIMA
RIMB
RIMC
RIMCA
RIMCB
RIMCC
RIMCD
RIMCE
RIMCF
RIMCG
RIMCH
RIMCI
RIMCJ
RIMD
RIME
RIMEA
RIMEB
RIMEC
RIMED
RIMG
RIMH
RIMI

The dBase II Data Files:
RIM^CF.DBF
RIM^SF.DBF
RIM^VF.DBF
RIM^PF.DBF
RIM^OF.DBF
RIMDATE.MEM

The above data files are master copies. After the creation of a new data set the program copies the .DBF files and creates new index files with the same filename but a .NDX extension. This is described in Section I, Country/Zone Setup.

PRIMARY MENU

```
RIM                                                      TEST

          Resource Inventory and Management Limited
                              for
          International Livestock Centre for Africa

-------------------------------------------------------------
            Aerial Survey : Primary Menu {1.0}
-------------------------------------------------------------

        A : Define System Parameters
        B : Code File Maintenance and Reporting
        C : Data Entry, Validation and Listings
        D : Observer File Maintenance and Reporting
        E : Bias and Summary File Creation
        F :
        G : File Recovery Procedures
        H : Country/Zone Setup
        I : File Status

        Q : Exit to the dBase II Program
        X : Return to Operating System

        Your selection is  :_:
```

NOTE Each of these options represents a separate menu, some of which may themselves give options to choose further sub-menus. To change from one sub-menu to another, it will first be necessary to return to the primary menu, whereupon the new menu of options may be selected.

In the case of data entry screens, the cursor appears at the first field to be entered. For example, Summary File Data Entry of Animals data:

Animal Code = 101

Number of Herds __

Number of Animals

The cursor will appear against Number of Herds. (Animal Code 101 is prompted by the program). The operator enters the value for Number of Herds, followed by RETURN, and the cursor will move to Number of Animals. Enter the Number of Animals, followed by RETURN. This completes the screen. The display will clear and then present the next data variable.

Correcting Mistakes

If you make a mistake on the display you can reposition the cursor and correct your mistakes.

CTRL E moves the cursor UP the screen
CTRL X moves the cursor DOWN the screen
CTRL S moves the cursor LEFT along the line
CTRL D moves the cursor RIGHT along the line

On some microcomputers the ARROW keys will also move the cursor around the screen.

If you make a mistake, but only realise it after you have progressed to the next screen you must complete entry for that data record and then recall the record to AMEND the incorrect value.

This restriction applies particularly to the entry of the grid coordinates during data entry and editing. Once the second grid coordinate has been entered, the cursor movement keys should not be used to correct the mistake, as it is then possible to enter the values 0,0 into the dataset. If this is done inadvertently, then the package editing facility is unable to delete that grid, and the dBase editing system will have to be used to remove the record. The use of the dBase editor is described on page 53.

Pre-Prompted Characters

Some displays present either a letter or a number at the cursor position when the screen appears. The value will be either the most likely response, or a 'safe' response (eg. that will prevent you from deleting or overwriting a record unless you confirm your intention), or the 'default' value for a system parameter. If the character that is shown is the one you wish to enter, just press the RETURN key. If it is not, enter your chosen value and then press RETURN.

PRIMARY MENU OVERVIEW

The primary menu presents ten options which are selected by entering the identity letter for the option.

A : Define System Parameters
B : Code File Maintenance and Reporting
C : Data Entry, Validation and Listings
D : Observer File Maintenance and Reporting
E : Bias and Summary File Creation
F :
G : File Recovery Procedures
H : Country/Zone Setup
I : File Status

X : Exit to the dBase II Program
Q : Return to Operating System

A: Define System Parameters

This option is used to select the dataset to be used in the current session, to record the date, and to assign the appropriate disk drives for te programme and data files. It is also used to name a new set of data files which are ten created using Option H.

WARNING
Option H is used in conjunction with Option A, but only when the data files are first set up. The use of Option H on an existing set of files will overwrite those files with empty ones, and thus any data in them will be lost.

B : Code File Maintenance and Reporting

The code file contains details of all the variables used in the survey, and the permitted range of values for each variable. The code file is used to define the variables present in each survey. The data summary file has a fixed structure, but still permits different variables to be used in different survey areas. This is controlled by the code file. When the system is set up for a new survey, each variable is nominated as Active, Grouped (into another Active category) or Unused. The code file must be set up for each new survey area prior to data entry, and left unchanged until entry is completed.

C : Data Entry, Validation and Listings

Option C contains the ten programs which handle data entry, validation and listing of the summary, observation and photo files. This forms the main section of the data entry suite of programs.

ORGANISING YOUR WORK

Before embarking on.analysis of a full survey, some thought must be given to the operating procedure to be followed. A number of considerations are important here:

- -it is desirable to achieve a steady flow of work, rather than face a large data entry task after the survey is completed, when there is pressure for speedy reports.
- -data recorded from visual observations will need to be corrected for bias. This may have to be done on a daily basis, to take account of weather changes, or in some other subset of the whole survey.
- -to guard against hardware breakdown and accidental damage, security copies of the data should be kept and stored separately from the working files.
- -aerial survey data sets may be very large in size. Processing under dBase II is not very fast, and operators will find it preferable to work with small sub-sets of the data for entry and validation. The final Summary File can be built up by appending data using the summary programs.

the process of building up summary files is described in more detail on pages 17 and 41

NOTE
If you have too many variables to fit into one set of files, then enter the data into two separate set of files, and use the Mapping and Analysis package to combine the two files.

Operating Procedures

Routine operation of the program will involve a sequence of tasks starting with the creation of a new data set, and ending with the transfer of a summary file into a format suitable for subsequent analysis and mapping. The next seven pages present a summary of the operations in a schematic format. They are reproduced as aide-memoire 'prompt cards' accompanying this manual, for reference by data entry operators.

CREATE A NEW DATA SET

COMMAND LINE	enter DO RIM.
PRIMARY MENU	select Option A : Define System Parameters enter 4 character filename set the date if required confirm file:drive assignment
PRIMARY MENU	select Option H : Country/Zone Setup check the correct filename Correct? enter Y to create files Wrong? enter N to exit Return and repeat Primary Menu Option A
PRIMARY MENU	select Option Q : Exit to Operating System
COMMAND LINE	enter DIR for file listing check you have your ten data files: **filename^CF.DBF** **filename^CF.NDX** **filename^SF.DBF** **filename^SF.NDX** **filename^VF.DBF** **filename^VF.NDX** **filename^PF.DBF** **filename^PF.NDX** **filename^OF.DBF** **filename^OF.NDX**

NOTE
Both the DBF and NDX files are needed for the operation of this package. Thus, when moving a set of files from one disk to another it is essential for both sets of files to be copied.

ENTER AND LIST DATA

COMMAND LINE	enter DO RIM.
PRIMARY MENU	confirm correct filename set on display use Option A: to correct if necessary select Option C : Data Entry, Validation and Listing
DATA ENTRY MENU	select Option: A : Summary File Data Entry/Edit or, B : Summary File Block Data Entry/Edit or, E : Observation File Data Entry/Edit or, H : Photo File Data Entry/Edit enter data to **EXIT** enter West/East Axis 0 North/South Axis 0
DATA ENTRY MENU	select Option: D : Summary File Listing or, G : Visual File Listing or, J : Photo File Listing enter grid coordinates and list data to Display or Printer
DATA ENTRY MENU	**EXIT** to Primary Menu with Option Q
PRIMARY MENU	**EXIT** from the system with Option Q or choose another option

CREATE A SUMMARY FILE

COMMAND LINE	enter DO RIM.
PRIMARY MENU	check correct filename at TOP RIGHT of display use Option A: to correct if necessary select Option E : Bias and Summary File Creation
SUMMARY FILE CREATION MENU	select Option C : Observation to Summary File Transfer confirm that you wish to **APPEND** data to the Summary File. (The first transfer will append to an empty file) **EXIT** to PRIMARY MENU when completed
PRIMARY MENU	use Menu C : Data Entry, Validation and Listings to validate, edit and revalidate your Summary File, or to enter new data. **Exit** to the operating system when completed
OPERATING SYSTEM	make a security copy of your original data file

NOTE
When a summary file is created from either a PHOTO FILE or an OBSERVATION FILE, the records of each variable for each grid are summed to give a single value for each variable for each grid cell. These grid values are then ammended to any existing records within the summary file. Thus if, for example, 500 records are entered into a data file and then summarised, an then an additional 500 records are entered, and the enlarged data file summarised, the SUMMARY FILE will contain both sets of data records. Th first 500 records will therefore be duplicated with the SUMMARY FILE.

There are two ways to avoid this happening: either the first 500 records must be deleted from the data file before the new records are entered, or a new data file must be created into which the secord 500 records are entered. This new file can then be copied and renamed to match the original file, whereupon it may be summarised into the first SUMMARY FILE. This process is described further on page 41.

NOTE
Because the summary process appends data to the end of the existing file, it is not possible to summarise the observation file and the photo file into the same summary file. If you need to combine these files, it can be done within the mapping and analysis package as described in Part 11 of this Reference Guide.

Data File Management

Data management will vary with the size of survey, the number of variables being recorded, and the circumstances of data entry. For guidance, it is suggested that data entry is conducted in blocks or subsets of the survey. The size of a subset will be a matter of preference for the data entry operator. It should relate either to a logical unit, such as one day's work, or a physical unit, such as a number of grid rows or columns, or a natural geographic sub-group of the data set. As a guide to inexperienced operators, a file size of 1000 records would be a sensible compromise between excessive fragmentation of the data, and inconvenience caused by the speed of execution of validation and listing programs.

After every subset the operator would obtain a clean data set by repeated validation and editing, and append the summary of the data to the Summary File (using Primary Menu option E). In this way, the summary file will grow in stages as the data are available. When the Summary File is complete it would be **transferred** to the format for subsequent analysis. Depending on the size of each subset, the schedule of data entry, validation, editing and summary would fit a daily work cycle.

DEFINE SYSTEM PARAMETERS

This is used at the start of each new set of data, to set up the data files, record the current date, if required, and choose the appropriate disk drives or directories for the program and data files. Option H is used in conjunction with option A.

Choose A Data Set

The first screen to appear requests the user to select a FOUR character name for the data set to be used or created. The program automatically adds various suffixes to this name for each of the required files. The four characters can be made up from any letters or numbers, but should not include any other characters.

If you wish to create a new data set, you should either follow the instructions on the screen, if you are already familiar with dBase II commands, or alternatively, use option H on the Primary Menu, after finishing with option A. If the data set already exists, it is sufficient to enter the correct name on this screen, for all other programs to make use of the required data set.

The data set currently being used by the program is indicated by the four character name at the TOP RIGHT of every screen display.

Set The Date

The second screen enables the user to assign a date to the programs. The date is used on data file listings and validation reports. The user may choose to set the current date as a record of when work was done. Alternatively, it may be preferable to use the date on which the data were collected, or entered to computer files, if it is thought desirable to differentiate between sub-sets of data in this way.

Assign Files to Disk Drives

The third screen allows the user to assign files to disk drives. For routine operation on the Hewlett Packard HP 150 microcomputer with a hard disk, all files should be assigned to drive A:. This is the default setting in the program and should not be changed without reference to your software advisor.

CODE MAINTENANCE

The code file contains details of all the variables used in the survey, and the permitted range of values for each variable. The code file is used to define the data present in each survey. The data summary file has a fixed structure, but still permits different variables to be used in different survey areas. This is controlled by the code file. When the system is set up for a new survey, each variable is nominated as Active, Grouped (into another Active category) or Unused. The code file must be set up for each new survey area prior to data entry, and left unchanged until entry is completed.

Data Structure

Each record on the summary data file consists of nine groups of coded data, two fields of identifying data, and one field of computed data, in the following sequence:

- -Grid reference West/East (identifier)
- -Grid reference South/North (identifier)
- -Number of photographs per grid (computed for photo surveys only)
- -Animals
- -Settlements
- -Land Use
- -Crops
- -Infrastructure
- -Grid details
- -Zone details
- -Environmental data

The two grid references and the number of photographs are single fields each of two characters in width. The remaining groups have multiple fields, as shown below.

		Width	Total
Grid West/East:		2	2
Grid South/North:		2	2
Number of Photographs:		2	2
Animals: Six sets of	Number of Herds	2	
	Number of Animals	4	36
Settlements: Seven sets of			
	Number of Settlements	2	
	Number of Dwellings	4	42
Land Use: Ten sets		2	20
Crops: Six sets		3	18
Infrastructure: Five sets		2	10
Grid Details: Two sets		4	8
Zone Details: Five sets		2	10
Environment: Five sets		4	20

Code File Listing

The code file is listed in alphabetical order of the code groups. The user can specify the listing with an alphabetic range option. The default values are **From A, To Z,** which results in a complete listing of the code file. A single group can be listed by specifying the identity letter in both the From, and To, statements. From L, To L, would list the Land Use codes only.

When the listing is completed the user is returned to the Primary Menu.

Setting the upper and lower limits and choosing your categories

If you are entering multiple observations per grid, then the upper limit entered into the code file must refer to the summed value per grid, and not the individual counts.

Considerable care must be taken in assigning the variables to specific codes. As a general rule, estimate the maximum summed value per grid, and then choose a category which has the same field width. If the upper limit is bordering on the next largest field width, (e.g. 95 or 995) then choose a category with the next highest number of spaces in it. If you assign a category that is too narrow to take the largest summed value per grid, then it cannot be fitted into the summary file, and so will be shown as a missing value (**). This is to be avoided, as the mapping and analysis package cannot read a file with any missing values in it, and so you will be unable to analyse your data.

B : Summary File Block Data Entry

Code values from the Zone and Environmental groups will often apply to large areas of the survey. In order to avoid the repetitive task of entering the same data values separately for each grid, the program includes an option to enter data for a block of grids at a time. The option requires the user to define the **TOP LEFT** and **BOTTOM RIGHT** coordinates for a rectangular block, and to enter the variable values required. Irregular shapes can be built up from a number of rectangles. Single grid squares can be addressed by entering the same coordinates for both TOP LEFT and BOTTOM RIGHT. The coordinate system is described in Section II, Grid Square Coordinates.

Block entry should take place after the main summary data entry because the program only updates existing records and does not create new ones. Block entry works only for the Environmental and Zonal groups of variables.

C : Summary File Validation

A validation check is performed on the values entered for each variable. Records in error are listed on the screen, or to a printer. The block of grids to be validated must be referenced with TOP LEFT and BOTTOM RIGHT coordinates in a similar way as for block data entry. The validation checks the data against the numerical range specified in the LOWER LIMIT, UPPER LIMIT options in the code file. (See Menu B : Code File Maintenance).

A value is in error if it is less than the lower limit, or greater than the upper limit. In the case of Animal and Settlement variables, the validation is applied to the number of animals, and the number of dwellings, respectively. There is no check on the number of herds, or number of settlements.

Errors are indicated on the validation listing by a > symbol printed between the variable code and the value in error.

The current version of the program does not support any error checking between different variables on the same record, or across different records. It only checks if the value entered is within the range specified in the Code File.

Corrections to the data set are made from the Summary File Data Entry option, by entering the specific grid coordinates of the records in error.

OBSERVATION FILE DATA ENTRY

```
RIMce                                                           TEST

           Resource Inventory and Management Limited

                              for

           International Livestock Centre for Africa
-------------------------------------------------------------------
     Aerial Survey : Observation File Data Entry/Edit {1.0}
-------------------------------------------------------------------

                  West/East Axis : 2:
                North/South Axis : 3:
                 Observer Number : 2:

              Observation Number : 1:

                        Category :A:
                            Code :105:

                 Number Observed :     44:

          Was a Photograph Taken :Y:

              Photograph Number :100213:
                          Count :     40:

    Do you wish to enter further observations? :Y:
```

OBSERVATION FILE VALIDATION

RIMcf page: 1
11 June 1985

Resource Inventory and Management Limited

for

International Livestock Centre for Africa

Aerial Survey : Observation File Validation {1.0}

Title:

WE	NS	Obs.	Cat	Code	Obs	Photo No	Count	Error Message
1	1	5	A	104	60	137	55	Inactive Code
1	2	4	A	104	10			Inactive Code
1	2	5	A	104	22	139	18	Inactive Code

WAITING _

H : Photo File Data Entry

The Photo Data File is intended for data from photographic surveys. The file structure is almost identical to the Summary File, except that each photograph observation is numbered and there is one data record for each photograph. A grid square would contain a number of photographs, typically between 5 and 15.

Because the record structure follows the Summary File, the screen displays prompt for each new variable in the same way as for the Summary File Data Entry. This means that any grouping of variables with few observations, into a collective category, must be specified on the code file, before data entry.

The Photo File should be used for entry of data for which there are multiple observations in each grid square. After the file has been summarised into the Summary File, data which are recorded once per grid square, such as zonal or environmental features, should be entered using the Summary File or Block Data Entry programs.

The Summary File created from the Photo File contains the SUM of values entered for each grid. It is assumed that classificatory data would not be collected from more than one observation per grid square, and thus the Photo File would consist only of cardinal data, such as crop area or settlement counts. For analysis, the summed values must be corrected to means or totals per grid, using the information about the number of photographs per grid which is also stored on the record. This operation should be performed during the subsequent analysis.

I : Photo File Validation

A validation check is performed on the values entered for each variable. Records in error are listed on the screen, or to a printer. The block of grids to be listed must be referenced with TOP LEFT and BOTTOM RIGHT coordinates in a similar way as for block data entry. The validation is confined to the numerical range specified in the LOWER LIMIT, UPPER LIMIT options in the code file. (See Menu B : Code File Maintenance). The coordinate system is described in Part II.

A value is in error if it is less than the lower limit, or greater than the upper limit. In the case of Animal and Settlement variables, the validation is applied to the number of animals, and the number of dwellings, respectively. There is no check on the number of herds, or number of settlements.

Entering grids with no records

There will almost certainly be grids in your dataset for which you have no records to enter, for example when no animals are seen. However, it will be necessary to open such grids, so that they can be mapped. To do this, enter either spaces or zeros into the category and code descriptions. The grids will then be entered into the dataset with zero values for each of the activated variables, and will be flagged during the validation process as INVALID CAT/CODE, so that you can check them.

OBSERVER FILE MAINTENANCE

The Observer File contains details of the Observers who collected data for the survey, and their counting bias. In order that bias can be corrected for each variable entered on the Observation File, an Observer Bias record is held for each variable in use. The bias is updated automatically whenever a fresh calculation is made under Primary Menu Option E. Predetermined values can however, be entered using the file maintenance option D.

OBSERVER FILE

The Observer File is used to record the code numbers and names of the Observers used in the survey, and record their bias for each variable collected. Visual counting only occurs with data entered in the Observation File, and only relates to data for which multiple entries per grid can occur. Bias calculation is restricted to **Animal and Settlement** variables only.

The data entry screen is shown overleaf. The operator is requested to enter the following data:

- **Observer Number,** a code for the observer
- **Group Code,** A (Animals) or S (Settlements) for the chosen variable
- **Code Number,** of the variable
- **Name,** of the observer
- **Bias Factor,** a known or default value. In the absence of any previous known bias, the value 1.0 should be entered

This is repeated for each observer and each variable included in the study.

The Bias Factor is automatically updated whenever a bias calculation is made. When the Observation File is aggregated into the Summary File the bias values contained on the Observer File are applied to the visual observations. Details of the bias calculations can be found in **Section I, Bias and Summary File Creation.**

To **EXIT** from the observer file maintenance, enter blank or zero values for the observer number, group code and code number. The program will then offer the option of **LISTING** the observer file, either on the screen (Display) or to a printer (Print).

Observer File Listing

The observer file is listed in numeric order of the observer codes.

When the listing is completed the user is returned to the Primary Menu.

If multiple observations per grid have been entered for variables which should only be recorded once per grid square, the program will sum the values entered and the resulting sum will appear on the summary file. Care should be taken that this does not happen, by listing both the Photo and Summary files, for comparison before analysis.

If a Summary File exists before this option is chosen the program will **APPEND** the new data to the existing file. In order to guard against this happening by accident, the user is asked to confirm that a Summary File should be created, before the program is executed.

C : Observation to Summary File Transfer

This program summarises data from the Observation File into a Summary File. The Observation File would normally contain observations of animals and settlements, and any other variables for which there is more than one observation per grid square. The values recorded are summed by the program, into the record structure of the Summary file.

Because of the nature of the Observation File data set, it is possible to enter variables which are not coded as Active on the Code file. When the Summary option is executed, values of variables which are not specified as Active are excluded from the summation process and do not appear on the Summary file. If it is necessary to enter data for a larger number of variables than can be accomodated on the Summary File, variables with few observations can be **Grouped** into other categories by using the Grouping option on the Code File. (See Menu B Code Maintenance) The category into which variables are grouped must be set to Active before the summary is made.

During the summary process, the calculated bias is applied in the following way:

- -unique bias factors are used for each set of observer/variable codes. Thus Observer 1, Animal 101, would have a different bias from Observer 2, Animal 101, or Observer 1, Animal 103, etc.
- -if a photo count is present, the photo count value is used without any bias or other modification.
- -if there is only a visual count, but with a value of 10 or less, the count is used without modification.
- -if there is only a visual count, with a value greater than 10, the bias factor for that combination of observer and variable is applied before the count is added to the Summary File.

FILE RECOVERY PROCEDURES

File recovery uses the dBase II utilities to repack and reindex the data files. It would normally be executed when there has been some sort of malfunction of the program or computer during data entry. One example would be after a power failure.

This option should not be used during routine operation of the Aerial Survey Data Entry Program. In the event of any program malfunction or problem with the microcomputer or disks, such as in the list below, this option should be employed before seeking professional assistance from your software advisor.

Power failure during program operation
Diskette corruption
Accidental erasure of any file with a **.NDX** suffix
Any break in the program with the message
INDEX FILE CANNOT BE OPENED

The recovery procedure can be applied to any of the program data files: Code File, Summary File, Visual File or Photo File. The program prompts for the user to specify which file is to be used.

DATA COLLECTION

Data Collection

The aerial survey data, for which this suite of programs was written, are collected from visual observations or photographs. The flight path followed by the aircraft is designed to cover the survey area in a systematic fashion, along regularly spaced transects.

In order to achieve this regular pattern, the survey area is divided into a series of grid squares, oriented east/west and north/south. The dimensions of the squares are chosen in accordance with the desired sampling intensity of the survey. Typical values are 5km by 5km, and 10km by 10km.

The transects may be flown along either axis, and in either direction. In a large study area, it is likely that data will be collected from all possible transects: east-west, west-east, north-south, and south-north. The direction of flight does not affect the statistical design, and the flights are chosen on purely practical grounds, relating to efficient operation of the aircraft.

Grid Square Coordinates

Individual Squares

Each grid square is identified by a West/East and a South/North coordinate. They are arranged on a cartesian basis with the origin at the bottom left of the study area. The West/East axis is numbered FROM West TO East. The South/North axis is numbered FROM South TO North. An individual grid cell is identified by the West/East coordinate, followed by the South/North coordinate. This is illustrated below:

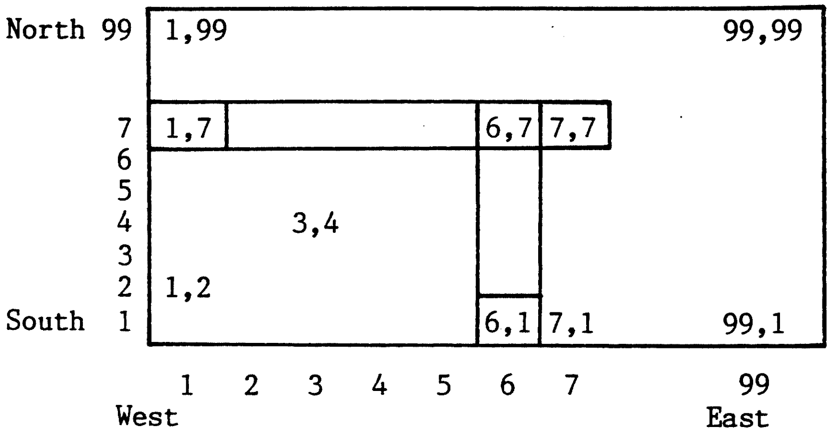
North	7	1,7						7,7
	6	1,6						
	5	1,5						
	4	1,4		3,4				
	3	1,3						
	2	1,2						
South	1	1,1	2,1	3,1	4,1	5,1	6,1	7,1
		1	2	3	4	5	6	7
		West					East	

The data entry program will accept up to 99 West/East grids, and up to 99 South/North grids.

The data files store information only for records which have been entered from the data entry program. If the study area is irregular in shape is is possible to define a grid dimension larger than the maximum coordinates of the area, to ensure that the whole area is listed, or otherwise dealt with. The cells with no data are merely ignored by the program.

Rows or Columns of Grids

The previous example can also be used to illustrate how to address a specific row or column of data. The West/East row with a South/North value of 7, is defined as 1,7 to 7,7. The South/North column with a West/East value of 6, is defined as 6,7 to 6,1.



Careful use of these coordinates will permit the operator to list or validate selected parts of the data set. Listing of a row or column can be used to study individual flight transects.

Summary Data Transfer File Format

Field	Name	Type	Width	Description	File columns
001	SF:WEST	N	002		1 - 2
002	SF:NORTH	N	002		3 - 4
003	SF:PHOTOS	N	002		5 - 6
004	SF:ANIMALS	C	001	The letter A	7
005	Herds1	N	002		8 - 9
006	Animals1	N	004		10 - 13
007	Herds2	N	002		14 - 15
008	Animals2	N	004		16 - 19
009	Herds3	N	002		20 - 21
010	Animals3	N	004		22 - 25
011	Herds4	N	002		26 - 27
012	Animals4	N	004		28 - 31
013	Herds5	N	002		32 - 33
014	Animals5	N	004		34 - 37
015	Herds6	N	002		38 - 39
016	Animals6	N	004		40 - 43
017	SF:SETTLE	C	001	The letter S	44
018	Settlement1	N	002		45 - 46
019	Dwellings1	N	004		47 - 50
020	Settlement2	N	002		51 - 52
021	Dwellings2	N	004		53 - 56
022	Settlement3	N	002		57 - 58
023	Dwellings3	N	004		59 - 62
024	Settlement4	N	002		63 - 64
025	Dwellings4	N	004		65 - 68
026	Settlement5	N	002		69 - 70
027	Dwellings5	N	004		71 - 74
028	Settlement6	N	002		75 - 76
029	Dwellings6	N	004		77 - 80
030	Settlement7	N	002		81 - 82
031	Dwellings7	N	004		83 - 86
032	SF:LAND	C	001	The letter L	87
033	Landcat1	N	002		88 - 89
034	Landcat2	N	002		90 - 91
035	Landcat3	N	002		92 - 93
036	Landcat4	N	002		94 - 95
037	Landcat5	N	002		96 - 97
038	Landcat6	N	002		98 - 99
039	Landcat7	N	002		100 - 101
040	Landcat8	N	002		102 - 103
041	Landcat9	N	002		104 - 105
042	Landcat10	N	002		106 - 107
043	SF:CROPS	C	001	The letter C	108
044	Crop1	N	003		109 - 111
045	Crop2	N	003		112 - 114
046	Crop3	N	003		115 - 117
047	Crop4	N	003		118 - 120
048	Crop5	N	003		121 - 123
049	Crop6	N	003		124 - 126

050	SF:INFRA	C	001	The letter I	127
051	Infra1	N	002		128 - 129
052	Infra2	N	002		130 - 131
053	Infra3	N	002		132 - 133
054	Infra4	N	002		134 - 135
055	Infra5	N	002		136 - 137
056	SF:GRID	C	001	The letter G	138
057	Grid1	N	004		139 - 142
058	Grid2	N	004		143 - 146
059	SF:ZONE	C	001	The letter Z	147
060	Zone1	N	002		148 - 149
061	Zone2	N	002		150 - 151
062	Zone3	N	002		152 - 153
063	Zone4	N	002		154 - 155
064	Zone5	N	002		156 - 157
065	SF:ENVIRON	C	001	The letter E	158
066	Environ1	N	004		159 - 162
067	Environ2	N	004		163 - 166
068	Environ3	N	004		167 - 170
069	Environ4	N	004		171 - 174
070	Environ5	N	004		175 - 178
		TOTAL	178		

Note. The field names chosen are purely illustrative in order to show the relationship between the transfer file and the original dBase II Summary file.

Observation Data File

dBase II filename: RIM^VF.DBF

Field	Name	Type	Width
001	VF:WEST	N	002
002	VF:NORTH	N	002
003	VF:PERSON	N	002
004	VF:OBS	N	002
005	VF:CAT	C	001
006	VF:CODE	N	003
007	VF:VALUE1	N	004
008	VF:PHOTO	N	006
009	VF:VALUE2	N	004

KEY:
Field: data field number
Name: program variable name
Type: Character or Numeric variable
Width: length of the field in characters

Master Code File

When a new set of data files are set up for a survey the details of the data file structures are taken from the five master files noted above:

RIM^CF.DBF
RIM^SF.DBF
RIM^VF.DBF
RIM^PF.DBF
RIM^OF.DBF

In the case of the code file however, the setup program (Option H on the Primary Menu) copies the content of the RIM code file as well. In this way, the RIM code file acts as a master code file for all surveys. If the need arises, the existing master file can be replaced by the survey code file. To do this, enter the following commands from the MS DOS operating system prompt.

DEL RIM^CF.DBF	this deletes the existing master code file
COPY filename^CF.DBF RIM^CF.DBF	this makes a new copy of the survey codefile (called filename^CF.DBF here) with the name RIM^CF.DBF.

The symbol ^ is not an embedded control character but the uparrow key, found at SHIFT 6 (ASCII chr 94), on many keyboards.

Before issuing the above commands it is prudent to check that you have a security copy of your files in case of an error in the commands. See Section I, Organising Your Work.

dBASE II AND MS DOS COMMANDS

This section presents a limited number of dBase II and MS DOS commands to enable the user to manipulate the data files. You are advised to take security copies of your files before using these commands to modify your file contents. For further details consult your computer operating manual, and the dBase II Product Manual. Users who are interested in programming with dBase II are recommended to read 'dBase II For Every Business' by Robert A Byers, published by Ashton-Tate, producers of dBaseII.

dBase II

Command Line Prompt

The dBase II program uses a period or full stop .

List a file

To list the contents of the Summary file, enter:

```
. USE filename^SF INDEX filename^SF
. LIST ALL
```

List a Variable

To list the animals category on the Summary file, enter:

```
. USE filename^SF INDEX filename^SF
. LIST ALL SF:ANIMALS
```

Delete a Record

First list the contents of the file and note the record number (nn) of the record you wish to delete. Then enter:

```
. DELETE RECORD nn
```

After you have issued the delete command for all the records to be deleted enter:

```
. PACK
```

This causes the records to be removed from the data file.

Copy a File

A> COPY sourcedrive:sourcename destinationdrive:destinationame

Display the Directory

A> DIR drive:

For the HP150, assuming the data and programme files are all on disk A, and that you wish to build a series of PHOTO FILES called PHO1, PHO2, and PHO3 into a master summary file called PHOM, the following (in bold face type) commands should be used:

```
do rim.          Then select Option A    Then assign name PHOM
                 Select Option H to set up PHOM files
                 Exit to Operating System

copy A:\PHO1PF.* A:\PHOMPF.*

do rim.         Select Option E          Then select Option B
                Summarise data file
                Exit to Primary Menu and then to Operating System

copy A:\PHO2PF.* A:\PHOMPF.*

do rim.         Select Option E          Then select Option B
                Summarise data file
                Exit to Primary Menu and then to Operating System

copy A:\PHO3PF.* A:\PHOMPF.*

do rim.         Select Option E          Then select Option B
                Summarise data file
```

ENTER AND LIST DATA

COMMAND LINE	enter DO RIM.
PRIMARY MENU	confirm correct filename set on display use Option A: to correct if necessary select Option C : Data Entry, Validation and Listing
DATA ENTRY MENU	select Option: A : Summary File Data Entry/Edit or, B : Summary File Block Data Entry/Edit or, E : Observation File Data Entry/Edit or, H : Photo File Data Entry/Edit enter data to **EXIT** enter West/East Axis 0 North/South Axis 0
DATA ENTRY MENU	select Option: D : Summary File Listing or, G : Visual File Listing or, J : Photo File Listing enter grid coordinates and list data to Display or Printer
DATA ENTRY MENU	**EXIT** to Primary Menu with Option Q
PRIMARY MENU	**EXIT** from the system with Option Q or choose another option

MODIFY AND LIST THE CODE FILE

COMMAND LINE	enter DO RIM.
PRIMARY MENU	confirm correct filename set on the display use Option A : to correct if necessary use Option A : to set the date if required
PRIMARY MENU	select Option B : Code File Maintenance enter/modify your codes: enter identifying group letter (eg. L land use) enter: new or existing code number description Active/Grouped/Unused Option lower limit/upper limit of values repeat until all codes checked/entered enter 0 for category 0 for code, to **EXIT** select Y to list code file, or N to **EXIT** choose alphabetic range of code groups to list and Display or Print
PRIMARY MENU	select Q to exit or choose another option

CREATE A TRANSFER FILE

COMMAND LINE	enter DO RIM.
PRIMARY MENU	check correct filename at TOP RIGHT of display use Option **A**: to correct if necessary select Option E : Bias and Summary File Creation
SUMMARY FILE CREATION MENU	select Option D : Transfer File Creation
TRANSFER MENU	choose the file to be transferred A : Code File or, B : Summary File or, C : Observation File or, D : Photo File enter a FILENAME up to 8 letters and numbers **EXIT** to PRIMARY MENU when completed
PRIMARY MENU	enter Q to EXIT to operating system
COMMAND LINE	enter DIR to list your files and confirm transfer file created. Note it has a **.TXT** extension
COMMAND LINE	enter TYPE filename to list the transfer file contents to your display

The procedures described below are designed to give a simple guide for first time users. As you become more familiar with the package, you will undoubtedly discover short cuts, and entry techniques which are more suited to your particular requirements.

The variables have been given the following names and codes:

NAME	CODE	RANGE	FILE	WHERE ENTERED
MULTIPLE OBSERVATIONS PER GRID				
ANIMALS				
Cattle	A 101	0 to 150	YYYY	Observation File
Sheep and Goats	A 102	0 to 150	YYYY	" " "
Camels	A 103	0 to 150	YYYY	" " "
CROP DENSITIES				
Millet	C 401	0 to 100	XXXX	Photo File
Groundnuts	C 402	0 to 100	XXXX	" "
Yams	C 403	0 to 100	XXXX	" "
Cotton	C 404	0 to 100	XXXX	" "
Rice	C 405	0 to 100	XXXX	" "
PHOTOINTERPRETATION COUNTS				
Cultivation	L 301	0 to 200	XXXX	" "
Grassland	L 302	0 to 200	XXXX	" "
Woodland	L 303	0 to 200	XXXX	" "
Forest	L 304	0 to 200	XXXX	" "
Bare ground	L 305	0 to 200	XXXX	" "
Rds and settle's	L 306	0 to 200	XXXX	" "
HUMAN HABITATION				
Pastoral	S 201	0 to 100	YYYY	Observation File
Arable	S 202	0 to 100	YYYY	" " "
SINGLE RECORDS PER GRID				
Rainfall	E 801	1500 to 1500	YYYY	Summary File
Altitude	G 601	750 to 850	YYYY	" "
Admin Region	Z 701	1 to 2	YYYY	" "
Development Region	Z 702	1 to 2	YYYY	" "

The dataset consists of nine grids.

Finally, print out your new, cleaned code file by entering 0,0 etc., which will first list the codes to be deleted (if shown WAITING, press RETURN), and ask you to confirm their deletion. The print out should look like Figure 1.

STEP FOUR: Entering your data (See pages 15 & 34)

The specimen dataset for you to use is presented in two ways: one on data coding sheets at the end of this manual, and the other as a print out of the completed data entry files.

Your are now about to enter the Crop Density and Photointerpretation Counts (i.e the LAND and CROPS categories) into the **PHOTO FILE**. Select Option C from the Primary Menu, wait, then select Option H. The programme will now wait for you to enter a pair of grids coordinates. To enter grid 1,1, simply press 1, then RETURN, then 1, then RETURN. If you make a mistake, use the cursor control keys (see page 7) to move the cursor to the error, and then retype it correctly. Once you have entered the second coordinate, you cannot go back and correct any mistakes using the cursor keys. You must complete the next step first.

The programme will then show Photograph Number :1:, indicating that this is the first photo to be entered in grid 1,1. This is correct, so press RETURN. Then you will be told that the record is not on file, and asked whether you wish to add the record, or to exit from the option. If you have made a mistake in entering the coordinates, then press E to exit, and try again. If not, then press A.

The first active variable will then appear on the screen. This will be L 301 (cultivation). Enter the appropriate value (20), and because the figure fills the space between the colons, the next variable (L 302 grassland) will automatically be shown. Enter its value (5), but you will have to press RETURN this time because the value does not fill the space between the colons. Now enter the remaining figures for that photograph. Note that the default value is zero, so to enter a value of zero, just press RETURN, and the next category will appear.

When all the active variables have been given, the grid coordinates will reappear on their own, showing the last coordinates entered. Since you now want to enter Photo Number 2 into the same grid, simply press RETURN twice, to enter the numbers shown on the screen. Photograph Number :2: will then automatically appear. This is correct, so press RETURN again to confirm it. Then wait a few moments for the next screen to appear, and enter A to add the counts from the second photograph to the file.

Now put in the rest of the photo records for grid 1,1, and start on grid 1,2. Remember to change the north south grid coordinate (the second one) when you change grids. Now enter the rest of the data for the remaining 7 grids.

Once you have done this enter 0,0,0 into the grid coordinates and photograph number, and you will be returned to the Option Menu.

STEP FIVE: Validating, listing and editing your new **PHOTO FILE**
(See Pages 16,34 & 35)

The purpose of validation is to pick out any entries that are outside the numerical limits or active categories defined in the code file. In other words, it is a quick way to find some of the more common errors. You will also need to list the file you have just made to see if there are any other mistakes in it.

To validate the File, select Option I, or if you have gone back to the opening menu of the programme, you will have to select Option C then Option I. Now answer the questions you are asked, so that you will be given a print out of any values in error. Note that for the whole data set to be validated, simply enter the default coordinates when asked, by pressing RETURN at each stage.

To list a file is a similar procedure to validating it. Select Option J, confirm you wish to go ahead by entering P, then enter a report title if you wish, press RETURN when asked for the grid coordinates to include in the report, so as to confirm the default settings and thus include the whole dataset, and then press P to get the report printed rather than displayed.

Now check the listing and validation printout to identify the errors, if any. To correct them, go into the Photo File Data Entry/Edit Option (H). Then one by one, call up the grid coordinates **and the relevant photograph number**. You will then be told that the record is on file, and will be asked whether you wish to amend or delete the record. To delete it press D and then move on to the next error. To amend the record, press A, wait, and keep pressing RETURN until you reach the offending value, and simply type in the right figure. Then keep pressing return until you reach the end of the record. Now repeat this process until all the errors are corrected.

Finally list the file contents a last time (Option J). It should be the same as the one shown in Figure 2.

STEP SIX: Summarising your cleaned **PHOTO FILE** (See Pages 17 & 39)

Summarising the File will add all the values of each variable from each grid cell, so that a file is created which contains a single figure for each variable for each grid. To do this, go back to the Primary Menu (Option Q), and then select Option E. Now select Option B, and when prompted, press Y to start the procedure. This process takes up to 30 seconds per record, and so will take a few minutes to complete.

Now list your **SUMMARY FILE** as a final check of the data, and to provide a hard copy record. To do this, return to the previous menu by selecting Option Q, and then select Option C, followed by Option D. Then proceed in exactly the same way as you did with the photo file listing.

FIGURE 2. (Continued)

```
RIMcj          Resource Inventory and Management Limited          Page:   3
01 NOV 1986                         for
               International Livestock Centre for Africa

                 Aerial Survey : Photo File Listing (1.0)

          Title:

WE NS    <-Animals-> <-Settle--> Land   Crops  Infra  <-Grid-> Zones  Environ

 2  1  2                         301 20 401   5
                                 302 10 402   5
                                 303  8 403   0
                                 304  0 404   0
                                 305  1 405   0
                                 306  1

 2  1  3                         301 20 401   6
                                 302  8 402   4
                                 303 10 403   0
                                 304  0 404   0
                                 305  1 405   0
                                 306  1

 2  1  4                         301  0 401   0
                                 302 20 402   0
                                 303 20 403   0
                                 304  0 404   0
                                 305  0 405   0
                                 306  0

 2  2  1                         301 20 401  10
                                 302  9 402   5
                                 303  6 403   0
                                 304  0 404   0
                                 305  2 405   9
                                 306  3

 2  2  2                         301 20 401  15
                                 302  7 402   5
                                 303 10 403   0
                                 304  0 404   0
                                 305  1 405   0
                                 306  2

 2  2  3                         301 18 401  10
                                 302  8 402   8
                                 303 10 403   0
                                 304  0 404   0
                                 305  2 405   0
                                 306  2

 2  2  4                         301 18 401   8
                                 302 11 402  10
                                 303 10 403   0
                                 304  0 404   0
                                 305  0 405   0
                                 306  1
```

```
RIMcj          Resource Inventory and Management Limited          Page:   4
01 NOV 1986                         for
               International Livestock Centre for Africa

                 Aerial Survey : Photo File Listing (1.0)

          Title:

WE NS    <-Animals-> <-Settle--> Land   Crops  Infra  <-Grid-> Zones  Environ

 2  2  5                         301 20 401  10
                                 302 10 402  10
                                 303  7 403   0
                                 304  0 404   0
                                 305  1 405   0
                                 306  2

 2  3  1                         301 20 401   0
                                 302  5 402   0
                                 303 10 403  20
                                 304  0 404   0
                                 305  1 405   0
                                 306  4

 2  3  2                         301 18 401   0
                                 302 10 402   0
                                 303 10 403  10
                                 304  0 404   0
                                 305  0 405   8
                                 306  2

 2  3  3                         301 20 401   0
                                 302 10 402   0
                                 303  5 403  20
                                 304  0 404   0
                                 305  1 405   5
                                 306  4

 2  3  4                         301 10 401   0
                                 302  7 402   0
                                 303 10 403   5
                                 304 10 404   0
                                 305  2 405   5
                                 306  1

 2  3  5                         301 30 401   0
                                 302  5 402   0
                                 303  5 403  15
                                 304  0 404   0
                                 305  0 405  15
                                 306  0

 3  1  1                         301  0 401   0
                                 302  0 402   0
                                 303  0 403   0
                                 304 40 404   0
                                 305  0 405   0
                                 306  0
```

STEP SEVEN: Creating a Transfer File from the **SUMMARY FILE**
(See Page 41)

A Transfer File is needed so that the data is in a form that is accessible to the mapping and analysis package, which cannot read dBase II files like the ones you have just made. The Transfer File is a fixed format file with the structure and field widths shown on pages 48 and 49.

To create a Transfer File, go back, if necessary, to the Primary Menu, by selection Option Q until you reach it. Then select Option E (Bias and Summary File Creation), followed by Option D. You want to Transfer the **SUMMARY FILE**, so now select Option B. Then you will be asked to provide a name for the Transfer File. You must call this file XXXX^SF if you wish to be able to use the OBEY files written for the specimen datasets and described in Volume II of this Reference Guide. A suffix .TXT will be added automatically, so that the resulting filename will be XXXX^SF.TXT.

Now go back to the opening menu by pressing Q until the Primary Menu appears.

STEP EIGHT: Setting up the YYYY file (See pages 13, 14, 21 to 26)

This file is needed so that you can enter the Animals, Human Habitation and Single Records. You could enter these values straight into your XXXX file, if you altered the Code File appropriately, but as explained on pages 17 and 41, you cannot summarise both the **PHOTO FILE** and the **OBSERVATION FILE** into the same **SUMMARY FILE**, as this would mean that the **SUMMARY FILE** would contain two sets of grid coordinates. Thus you need the YYYY files for the observation data.

You could easily enter all the YYYY file variables into the **OBSERVATION FILE**. However, to give you some practice, the Single Record variables should be entered into the **SUMMARY FILE**. Therefore you first need to enter the Multiple Records (i.e Animals and Human Habitation) into the **OBSERVATION FILE** and summarise it, so that you can enter the Single Records into the newly created **SUMMARY FILE**.

Setting up the YYYY files involves exactly the same procedure as described in STEP TWO, except enter YYYY rather than XXXX when you define the system parameters (Primary Option A). Then use Option H.

Now you must set up the YYYY Code File. This is very similar to STEP THREE, except the variables set as active will be the ones you need for the Animals and Human Habitation records. Do not, at this stage, activate the Single Record variables. This you will do just before you enter them into the **SUMMARY FILE**.

As you will realise, setting up the Code File from scratch is quite a complex process because the default File contains a lot of variables you do not need, which you have to delete. As you have already set up the XXXX Code File (which is very similar to the one you need for the YYYY file) there is a short cut you can take.

This short cut involves copying the YYYY Code File onto the XXXX, one and then merely changing the variables which are set as active, and so avoids all the deletions of the unwanted default variables. To do this, Exit to the Operating System, wait for the A> prompt to appear, and then type COPY XXXX^CF.* YYYY^CF.*, and press RETURN.

Then go back into the data entry package (DO RIM.), select Option B, and call up each of the Category and Codes in turn, setting them as active or inactive accordingly. The Animal and Human Habitation variables should be active, and the remainder inactive. Then print the Code File. It should look like Figure 4.

STEP NINE: Entering the Observer Counts (See Pages 6, 15, 29 & 30)
You will now be entering the A and S Code variables into the YYYY **OBSERVATION FILE**, and not the **PHOTO FILE**. This is because the **OBSERVATION FILE** programme calculates the observer bias for Animal and Settlement variables.

First locate the data on the specimen data coding sheets. The data consists of an Observed value (the Estimate), and if available, a count taken from Photographs (the Count). The first value for grid 1,1 should read Observer No 1, Category A 101, Observed value 30, Photo Count 28. Note that you will have to assign an identification number to each photograph taken, before you start data entry.

To start data entry, select Option C from the Primary Menu, then select Option E. You will then be prompted to enter the grid coordinates, followed by an Observer Number. Thus for the first value type 1, RETURN,and then 1, RETURN. Remember that once you have entered the second coordinate, you cannot correct any errors with the cursor movement keys, but must complete the entry until you are given the option to exit or delete the record (see STEP FOUR).

Observer Number :0: will then be shown. This is incorrect, so enter 1, then RETURN. Observation :1: will now appear. this is correct, so press RETURN. Now you will be told that the record is not on file, so press A to start adding it to the file. Then you will be prompted to enter a Category and a Code for the observer count. Enter A, then 101. Note that you do not need to press RETURN because your entry fills the space between the colons. Now enter the observation (30), and press RETURN. You will the be asked whether a photograph was taken, which in this case it was, so enter Y. Then enter whatever number you have assigned to the photo (e.g 1), press RETURN and finally, enter the photo count (28), and press RETURN.

Now you will be asked whether you want to add further observations. Type Y, and the grid coordinates will reappear. The default entries for the grid coordinates, and the observer number are the ones which were last entered (in this case 1,1,1). This is correct for the second observation, so simply confirm the entries by pressing RETURN. Observation number :2: will now appear. As this is the second entry for that observer in grid 1,1, then just confirm the entry by pressing **RETURN**. When prompted, type A to start record entry. The category and code which were last entered will now be shown, which is correct, so confirm them by pressing **RETURN**. Then enter observer count (35), **RETURN**, type Y, enter the photo number (2), **RETURN**, and finally, the photo count (30), **RETURN**.

You should now be able to enter the rest of the observations from the dataset. Be careful to change the Grid Coordinates, Observer Number, Category and Code when necessary. Three of the grids have no observer counts in them, but you will still need to open the grids so that they will be included in the data file, for subsequent mapping and analysis. To open grids with no records, simply enter the grid coordinates, and Observer 1, Observation 1 with blanks (spaces) for the Category and Code, then a value of zero for the observation.

When you have finished entering all the observer count data, answer N when asked if you wish to enter futher observations. this will return you to the Menu screen.

STEP TEN: Validating, Listing and Editing the **OBSERVATION FILE** (See Pages 16, 31 to 33)

This is essentially similar to STEP FIVE, though the Options selected are slightly different because you are now dealing with the **OBSERVATION FILE** rather than the **PHOTO FILE**.

Thus select Option F from Primary Menu Option C for Observation File Validation, and list the errors. Note that the grids with zero values, that you opened by entering blanks will be flagged as INVALID. Do not worry, this does not invalidate the entry, but is intended only to help you check the zero entries quickly.

To list the **OBSERVATION FILE** select Option G from Primary Menu Option C, or if you are still in the data entry, enter N when asked if you wish to enter more observations and select Option C . Refer to STEP FIVE above if you get stuck. To edit and correct the **OBSERVATION FILE** select Option E from Primary Menu Option C, call up the relevant grid coordinate, Observer Number **and Observation Number**, type A (to ammend the record) and enter the correct figures. Now get a complete listing of the newly corrected **OBSERVATION FILE**. It should look like Figure 5, though the observation numbers may differ as they depend on the order in which the records were entered.

STEP ELEVEN: Setting up the Observer File, and calculating the Observer Biases (See Pages 36 to 39)

The observer file is the one from which the observer bias values are taken when the **OBSERVATION FILE** is summarised. To set it up, go to the Primary Menu and select Option D. You then have to open a bias for each observer for each of the active Animal and Settlement variables that you have used in the **OBSERVATION FILE**. Thus, when the programme screen appears (see page 37 for an illustration), enter 1, **RETURN**, A, 101, and then A (for ammend) when prompted. Now enter a number or name (e.g 1 or Observer One), **RETURN**, followed by a bias value of 1 (and then **RETURN**). Do this for each observer (one and two) for each active Animal and Settlement Code (A 101 to 103, S 201 and 202). Then Exit from the programme by entering zero values for the Observer Number, Group Code and Code Number, and then get a listing of the new Observer File. It should look like Figure 6.

This process will assign a default bias of 1 to each of the Animal and Settlement Codes. You now need to calculate the actual biases derived from the data you have put into the **OBSERVATION FILE**. To do this, go to the Primary Menu (Option Q, if necessary) and select Option E. Then select Option A. Then confirm that the whole dataset is to be included in the bias calculations by pressing **RETUN** at each default coordinate displaed. The bias values will now be calculated, and put into the Observer File. When this has finished, go to the Primary Menu, select Option D, enter zero for the observer number, group code and code number, press Y, then P, and you will get a listing of the updated Observer File. It should look like Figure 7. If WAITING is displayed at any stage, press **RETURN**.

STEP TWELVE: Summarising the **OBSERVATION FILE** (See pages 17,& 40)
You have now entered all the multiple records per grid, and so are ready to summarise the **OBSERVATION FILE**, to give you the summed values for each variable, for each grid. To do this, go to the Primary Menu, select Option E, then Option C. Confirm your intention to summarise by typing Y, and then wait for the process to finish. This will take a few minutes.

Now get a listing of the **SUMMARY FILE**, by selecting Primary Menu Option C, then Option D (see page 29) and confirming the grid coordinates to include the whole dataset. It should look like Figure 8.

You are now ready to enter the single records per grid into the **SUMMARY FILE**.

STEP THIRTEEN: Entering data into the **SUMMARY FILE**.
(See Pages 16, 27 & 28)
Firstly you must activate the Codes for the variables to be entered, by using the Code File Maintenance and Reporting Option (Primary Menu Option B), and calling up the inactive E, G and Z Codes, and setting them to active. **Do not inactivate the animals or settlement codes**. If you get stuck, then refer to STEP THREE and STEP EIGHT). When you list the new Code File, it should look like Figure 9.

Now enter the Altitude, Admin Region and Development Region values into the **SUMMARY FILE**, using Primary Menu Option C, and then selecting Option A. Entering data into the **SUMMARY FILE** is very similar to putting data into the **PHOTO FILE**, except that you can only enter one value per grid, and there is no photograph number to select. When you enter the grid coordinates, and then confirm that you wish to add/ammend the record, all the active variables will be

FIGURE 8. YYYY SUMMARY FILE AFTER ANIMAL AND SETTLEMENT DATA ENTRY

RIMcd
01 NOV 1986

Resource Inventory and Management Limited
for
International Livestock Centre for Africa

Page: 1

Aerial Survey : Summary File Listing {1.0}

Title:

WE	NS	<-Animals->			<-Settle-->			Land	Crops	Infra	<-Grid->	Zones	Environ
1	1	101	4	145	201	2	11						
		102	1	71	202	1	65						
		103	3	48									
1	2	101	2	30	201	2	20						
		102	0	0	202	2	43						
		103	2	10									
1	3	101	2	158	201	3	23						
		102	2	115	202	2	83						
		103	1	2									
2	1	101	2	75	201	2	15						
		102	4	221	202	3	112						
		103	1	10									
2	2	101	4	150	201	3	33						
		102	0	0	202	2	46						
		103	1	10									
2	3	101	2	28	201	3	28						
		102	2	103	202	2	97						
		103	1	20									
3	1	101	0	0	201	0	0						
		102	0	0	202	0	0						
		103	0	0									
3	2	101	0	0	201	0	0						
		102	0	0	202	0	0						
		103	0	0									
3	3	101	0	0	201	0	0						
		102	0	0	202	0	0						
		103	0	0									

** End of Report **

displayed in order. The animal and settlement codes will already have values in them from the **OBSERVATION FILE**, so remember to confirm them by pressing RETURN when the values are shown. When an empty varaible is shown, which will be Altitude, enter its value, and press RETURN. And so on. If you get stuck refer to STEP FOUR.

You can also add the Rainfall values at this point, if you wish, but, to get the maximum practice it is suggested that you use the Block Data Entry for this variable (see page 28), which means you should enter a value of 0 at this stage (i.e just press RETURN when the code appears).

When you have entered all the G and Z variables, exit the Data entry mode by entering 0,0 into the grid coordinates, and start to put the Rainfall data into the Block Data Entry. To do this, select Primary Menu Option C if necessary, then Option B. You will be asked whether you wish to enter a block of data for the Environmental and Zonal activated Codes. When Environment Code 801 appears type Y, then enter 1500 (no RETURN is needed as the value fills the space between the colons), and then enter the top right and then bottom left grid coordinates. These are 1,3 and 3,1 respectively. The completed screen should look like Figure 10. Now enter N because you do not want to enter any more values for E 801. Then enter N when asked if you want to enter data from the Zonal Codes.

Finally list the completed **SUMMARY FILE** (Primary Menu Option C if needed, then Option D). It should look like Figure 11.

STEP FOURTEEN: Creating the Transfer File from your **SUMMARY FILE**.
(See Pages 18 and 41)
This is the almost the same as STEP EIGHT. Select the Primary Menu Option E, then Option D, and then Option B, as in STEP EIGHT. However call this Transfer File **YYYY^SF**. If you call it XXXX^SF, then the Transfer File you made in STEP EIGHT will be overwritten and lost. Further, the Obey Files presented in Part II of this Reference Guide will not work.

STEP FIFTEEN: Analysing the data.
You have now completed the data entry section of this specimen data set, and should be left with two .TXT files:- XXXX^SF.TXT with the observation records, and YYYY^SF.TXT with the photo data. Each file contains different information for the same grid cells, and now needs to be combined into a single dataset. This can not be performed within the data entry package, and so must be done from the mapping and analysis package. The procedures are described fully in Part II of this Reference Guide.

AERIAL SURVEY COMPUTER ANALYSIS

PART II : ANALYSIS AND MAPPING REFERENCE GUIDE

System Design: Colin Titcombe/
Bill Campbell
Program: Colin Titcombe
Documentation: Bill Campbell

Software and documentation by MAPICS Limited

TABLE OF CONTENTS

SECTION I : INTRODUCTION.................................1

SECTION II : EXAMPLES.................................4

SECTION III : CONCEPTS.................................10

COMMAND LANGUAGE.................................10
Commands.................................10
Arguments.................................11
Special Characters.................................11
Abbreviations.................................11
Filenames.................................12
Column Labels.................................12
Types of Command.................................13
Obey Files.................................14

DATA ENTRY AND STORAGE.................................14
Workspace.................................14
Attribute Data.................................15
Locational Data.................................15
Entering Data.................................15
Displaying Data.................................17
Storing Data.................................18
Workspace Management.................................18

MANIPULATION AND ANALYSIS.................................19
Summary Statistics.................................19
Arithmetic Operations.................................19
New Columns using Conditional Logic.................................21
Classification.................................21
Population Estimates.................................23

MAPPING.................................25
Drawing Device.................................25
Positioning and Scaling.................................25
Point Mapping.................................26
Other Map Features.................................28

SECTION IV : REFERENCE.................................30

APPENDIX.................................66

ANALYSING THE SPECIMEN DATASETS FROM VOLUME I 67

SECTION I INTRODUCTION

Mapping

Two types of mapping appropriate to air survey data are provided. The first method is more suited - but not confined - to pre-categorized variables. Symbols are plotted at point locations - with optional control of symbol type, size and colour - denoting the value or category of a variable. Symbol characteristics are either constant or can be made to vary according to the values of variables. The second method requires data to be classified before mapping. A variable is grouped into data classes, and a symbol plotted at each location according to the class in which that point falls. The size, colour, type and format of symbols are defined independently, and vary by class and not by individual value of a variable.

Maps can have linear features and text labelling added to them. The pen used for plotting - and therefore the colour and thickness of line - can be altered. The size and position of point maps and linear map features can be controlled as required.

Commands:

MARK	POINT	STRING	SYMBOL	DEVICE	ORIGIN	SCALE	
KEY	KEYOUTPUT	KEYTEXT	TEXT	SIZE	ITALIC	PLACES	PEN

Documentation

The rest of this guide is organised as follows:

- SECTION II provides worked examples, which the user will find helpful and should run on the system before starting work.

- SECTION III explains basic command language concepts, and the three major groups of commands in some detail. It is essential reading for any user attempting to put together new applications.

- SECTION IV is a reference guide to all commands, listed in alphabetical order. It gives command syntax, description, examples and cross-references to related commands.

As well as the worked examples in the section immediately following, an annotated list of commands has been provided which can be used with the AAAA and BBBB specimen filed presented in Volume 1 of this Reference Guide. These commands should be followed to take the novice user from initiating the mapping and analysis programme to drawing a sample map.

EXAMPLES

This section provides examples of basic command language operations of the kind readily accessible to new users. Each example takes the form of an OBEY file of commands - these should be available on the HP150 on which the system is installed. The new user should start the program (normally with the command AM) and execute each obey file in turn, as follows:

```
OBEY <filename>
 .
 .      $ execution of commands
 .
 .
STOP
```

Files ONE to THREE demonstrate data handling; files FOUR to SEVEN require a plotter to be connected (see APPENDIX). While these last four files execute independently, they build up four plots on one A3 sheet which can be left in place between each command file example (see Figure 1). In order to examine output on the screen, the xon/xoff facility (keys CTRL-S and CTRL-Q) should be used to stop and resume output.

After running these simple examples, the user can proceed to **SECTION III CONCEPTS** for an explanation of commands and discussion of more advanced features.

Obey file ONE.OBY

```
$
$ read data set for Abet reserve
$ using free-format input to workspace
$ label data columns
$
read 'abet.att' c1 to c13
label c1 east  c2 north  c3 setful  c4 nomful  c5 agro
label c6 arabtin  c7 arable  c8 cattle  c9 catherd
label  c10 shoat  c11 shoather  c12 vegindex  c13 cultivat
$
$ create and label new variables
$
let c20=setful+nomful+agro
let c21=arable+arabtin
let c22=c20+c21
label c20 totpast  c21 totarab  c22 totalpop
$
$ look at workspace contents, write selected columns
$
```

```
status
write east north totpast totarab totalpop
$
$ make save file of complete data set
$
save 'abet.sav'
```

Obey file TWO.OBY

```
$
$ restore saved file to workspace and examine contents
$
retrieve 'abet.sav'
status
$
$ create subset of variables
$
erase c3 to c7 c12 c13 c20 to c22
status
$
$ output to file for printing using optional format
$
write 'temp.att' east north cattle catherd shoat shoather \
                                         '(2f4.0,4f7.1)'
$
$ empty workspace and read back shoats data
$
erase c1 to c50
read 'temp.att' c1 to c4 '(2f4.0,14x,2f7.1)'
label c3 shoats c4 shoathds
status
write shoats shoathds
```

Obey file THREE.OBY

```
$
$ retrieve saved data set, create cattle and shoat density
$ get descriptive statistics
$
retrieve 'abet.sav'
let c40=cattle/(5*0.8608)  $ divide by cell length*strip width
let c41=shoat/(5*.8608)
describe c8 c9 c40 c10 c11 c41
$
$ classify cattle distribution by alternative methods
$
class cattle 4
```

```
class cattle 4 2
class cattle 8
class cattle 4 9
class cattle 1 4 1 20 40 60 100
$
$ ratio estimates - unstratified, then stratified by
$ cultivation level for all classes and one specific class
$
estimate c1 c2 cattle width 5 5 0.8608
estimate c1 c2 totalpop width 5 5 0.8608
$
make c25 if(cultivat gt 25) then (3) \
     elseif(cultivat gt 10 and cultivat le 25) then (2) \
     else (1)
estimate east north cattle c25 width 5 5 0.8608 \
         output='catcul.tab'
$
estimate c1 c2 cattle c25 subclass 3 width 5 5 .8608
```

Obey file FOUR.OBY

```
retrieve abet.sav
status
$
$ set up plotter
$
device
$
$ point mapping using MARK - cattle distribution
$
let cattle=cattle/(5*.8608)
origin 0 0 10 50
scale 6
mark c1 c2 size=cattle maxsize 5.25  minsize 0 mark 5  1
$
$ add text and study area boundary
$
string abet.str
text 20 44 'Cattle distribution'
```

Obey file FIVE.OBY

```
retrieve abet.sav
$
device
$
$ point mapping using MARK - distribution
```

```
$ of high animal numbers
$
origin 0 0 10 130
scale 6
make c41 if(cattle>10 and shoat>=15)then (1) else (0)
mark c1 c2 size=2 mark c41 colour 3
$
$ add text and study area boundary
$
string abet.str
size 1.5 1.5
italic 15
text 20 124 'High animal numbers'
```

Obey file SIX.OBY

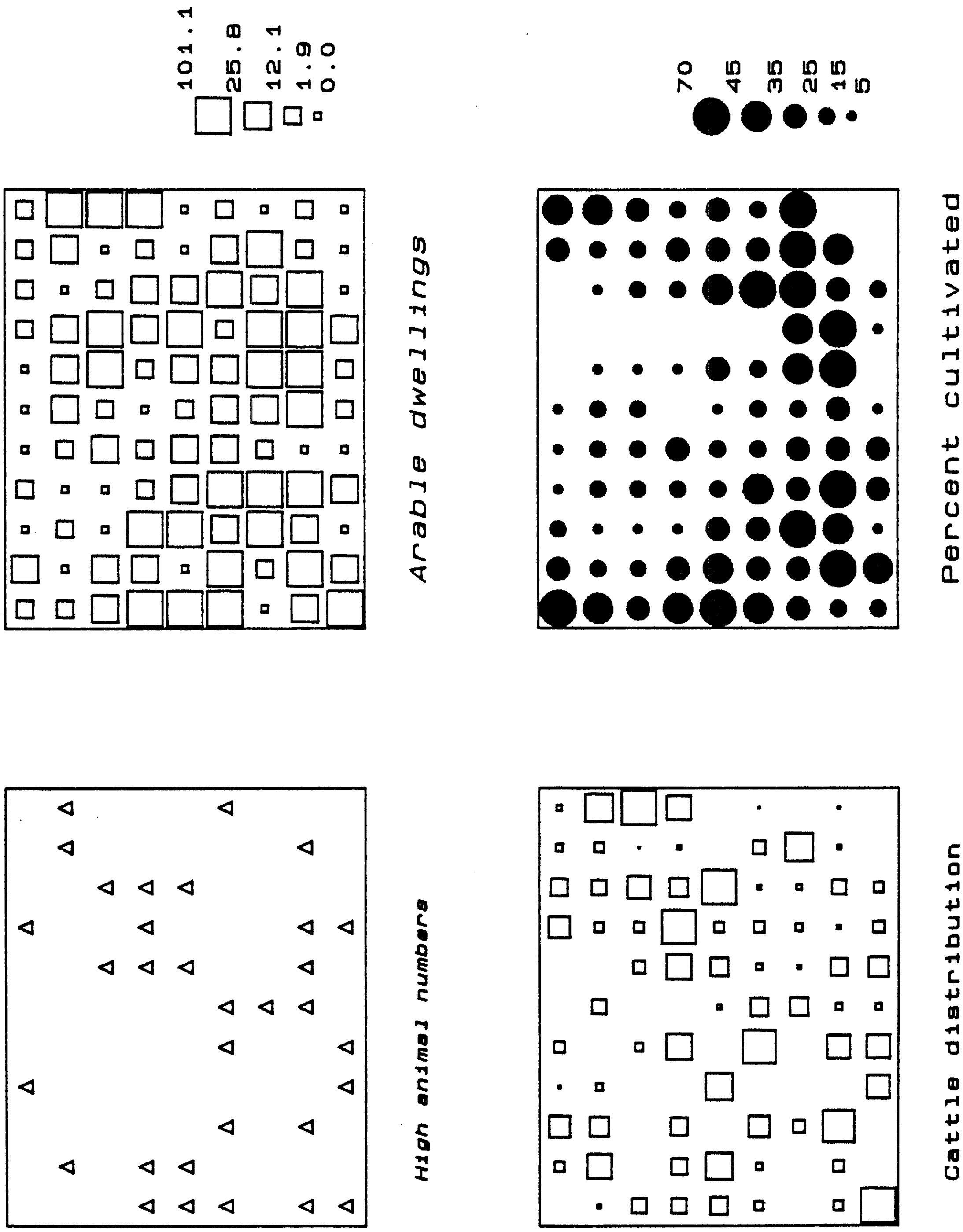
```
retrieve abet.sav
$
device
$
$ point mapping using POINT - distribution
$ of arable population
$
origin 0 0 100 130
scale 6
let arable=arable/(5*0.8608)
class arable 4 11
key 180 140
places 1
symbol 1 1.0 1 1 2
symbol 2 2.5 1 1 2
symbol 3 4.0 1 1 2
symbol 4 5.3 1 1 2
point c1 c2
$
$ add text and study area boundary
$
string abet.str
size 2.25 2.25
italic 15
text 110 124 'Arable dwellings'
```

Obey file **SEVEN**.OBY

```
retrieve abet.sav
$
device
$
$ point mapping using POINT - distribution
$ of cultivation
$
origin 0 0 100 50
scale 6
$
$ define new symbols
$
symbol 1 1.25 5 1 1
symbol 2 2.25 5 1 1
symbol 3 3.25 5 1 1
symbol 4 4.25 5 1 1
symbol 5 5.25 5 1 1
symbol
$
class cultivat 1 5 5 15 25 35 45 70
key 180 60
point c1 c2
$
$ add text and study area boundary
$
string abet.str
size 2.25 2.25
text 110 44 'Percent cultivated'
```

SECTION II EXAMPLES

Figure 1 : output from obey files FOUR to SEVEN



CONCEPTS

This section systematically explains the command language and data handling methods provided. Much the best way to understand many of the concepts is by practice, and the guide should be used in conjunction with experimentation at the keyboard.

COMMAND LANGUAGE

Commands

A session is controlled through commands issued at the keyboard or from a command or obey file stored on disk. Commands are executed as they are given until a STOP is issued to terminate the session. A command specification consists of the command name followed by a list of parameters or arguments controlling its action. Some commands do not require any arguments.

The REFERENCE section provides an alphabetic list of all commands with details of correct syntax, descriptions of function and examples. Documentation of command syntax uses standard notation. For example:

```
READ [<filename>] <Cn>...<Cn>  [<format>] !
     <Cn>...<Cn> [BEGIN] <datalist> END
```

where:

!	exclamation mark separates alternative forms of the command
<param>	parameter or argument supplied by the user; may be a real or integer number or a character string - a parameter name is given in angled brackets
<Cn>...<Cn>	one or more column numbers supplied by user; for example C23,C24 or C1 C2 C5 TO C8,C15
<KEYWord>	keyword constant with the significant part in upper case; for example OFF,SUBCLass,LABEl
<.null.>	the null parameter; giving no parameter invokes some default action
[<param>]	items in square brackets are optional and may be omitted

Separators between arguments or data values can be space, comma or equal sign. Commands, arguments and filenames can be given in upper or lower case or a mixture of both.

SECTION III CONCEPTS

Arguments

The following arguments are recognized:

Type	Examples
Integer:	1,2,3
Real:	1.0,2.0,3.0
Text string:	'One Two Three' 'TRANSF.TXT'
Column:	C1,C2
Keyword constant:	ON,OFF

Reals may be replaced by integers but most commands do not permit the reverse substitution. Text strings should be enclosed within single or double quotes but are recognized without them if the string is not a column or number and contains no special characters. For example:

```
READ C1 TO C3              $ column arguments
7 3 5.4                    $
7 4 18.9                   $ integer and real numbers
7 5 13.6                   $
END                        $ keyword constant
WRITE 'TEMP.TXT' C1 TO C3  $ text string for filename
```

Special characters

A number of characters have special meaning:

Separators - space, comma, equals. Command lines have parameters with one or more separators between them. Separators may be included within text strings if they are enclosed within single or double quotes.

Quotes (' and ") - single and double quotes are equivalent. Pairs of quotes may be used to begin and end text strings. Special characters are not recognized within quotes.

Continuation (\) - allows commands to be extended on to one or more continuation lines.

Comment ($) - text following this character in a command line is treated as comment and ignored.

Abbreviations

Only the first four characters of command names and keywords are significant. The following are valid versions of the ESTIMATE command:

```
ESTIMATE
ESTI
ESTIAMTE
esTiMATE
```

SECTION III CONCEPTS

Names and keywords can be entered in lower case or in a mixture of upper and lower case. Command names and keywords consisting of less than four characters cannot have additional characters appended to them.

Filenames

Filenames given as part of a command follow operating system conventions. It is advisable to enclose filenames in quotes, but they are recognised correctly without them provided no embedded separators or special characters are included. All filenames used in examples have been named according to the convention:

<filename>.<ext>

where <ext> is a three character extension or qualifier describing the data contained in the file. The full list of extensions is as follows:

file type	extension
attribute data	.ATT
obey or command	.OBY
SAVEd data	.SAV
string	.STR
text	.TXT

File names do not require these extensions; the user is advised, however, to adopt some form of file naming convention to help file maintenace and avoid confusion between file types.

Column labels

Column labels are significant in their first eight characters only. They are stored and retrieved automatically by the SAVE and RETRIEVE commands and make commands more understandable. Their use is recommended in applications with large numbers of data columns. Where column labels are used, column number equivalents are still recognised. For example, six columns containing tree data labelled thus:

```
LABEL C1 AGE C2 BIOMASS C3 GIRTH
LABEL C4 HEIGHT C5 OSGRIDX C6 OSGRIDY
```

can be referenced in any of the following ways:

```
C1 TO C6
AGE TO OSGRIDY
C1 TO OSGRIDY
AGE,C2 TO OSGRIDX, C6
```

SECTION III CONCEPTS

Types of command

Three types of commands are recognized:

- simple
- intelligent
- keyworded

Simple commands consist of a command name followed by zero or more parameters. These are given strictly in order although some trailing parameters may be omitted. For example:

```
SIZE <charwd> <charht> [<italic angle>] [<decimal places>]
```

may be used in several ways:

```
SIZE 8 10        $ specifies <charwd> and <charht>
SIZE 4 4 15 2    $ specifies all parameters
SIZE 4 3 20      $ specifies <charwd>, <charht> and <italic angle>
```

Intelligent commands are like simple ones but have alternative forms. Depending on the type of parameters supplied, different actions are taken. The READ command is a good example:

```
READ [<filename>] <Cn>...<Cn> [<format>] !
     <Cn>...<Cn> [BEGIN] <datalist> END
```

Thus:

```
READ 'TREES.ATT' C1 C4 TO C7
```

reads data from the file 'TREES.ATT' into columns C1, C4, C5, C6 and C7 using free format since the optional format specification has been omitted. The alternative form is for entering data interactively and is automatically recognized because of the arguments used:

```
READ C1 C2 BEGIN  1,3  2,7  4,9  7,8  END
```

This example causes the data list specified after the BEGIN keyword to be entered into columns 1 and 2. The data list is terminated by an END keyword. The list can be split over as many lines as required. BEGIN is optional if the data list starts on a new line.

Keyworded commands consist of a command name followed by a keyword/parameter list. Keywords allow parameters to be entered with a great deal of flexibility:

```
ESTIMATE C1 C2 C3 C10 SUBCLASS 3 4 6 WIDTH 5 5 0.8
```

In this example, the keyword SUBCLASS select an optional method for specifying stratification of estimates, and WIDTH separates subclass values from grid cell dimensions. Keywords are normally documented in positional order as for simple commands. See the REFERENCE section for details of keyword use for individual commands.

Obey Files

Command seqeunces can be stored in text files, created using the system editor, and executed by the OBEY instruction. This is particularly convenient when repeating standard sets of commands. Once a command file has been created, it is executed using OBEY. For example:

OBEY 'WATER.OBY'

causes the commands in 'WATER.OBY' to be executed as if entered from the keyboard. This continues until all the commands in the file are exhausted, when control returns to the keyboard. If an obey file contains a STOP, execution terminates normally. Obey files can include OBEY commands nested to five levels.

OBEY files must be written using the editor facility installed in the computer (e.g. Edlin, Editor or Wordstar). They cannot, therefore, be written or changed from within the mapping and analysis package. The commands needed for an OBEY file are precisely the same as those used during on-screen operation of the package.

DATA ENTRY AND STORAGE

This section describes data entry and storage methods internal to the package; types of data that can be handled; and the types of file that can be recognised or created. An important distinction must be drawn between data held in computer memory (the workspace) during a programme session, and data held on disk. The workspace is temporary and so is lost when a session ends (using the STOP command) unless it is stored on disk first. Disk files, in contrast, are permenant records of data.

Workspace

This is part of computer memory which holds user-defined columns of data; these can vary in length as required. In many sessions, the contents of the workspace are changed as columns are created, altered and erased - almost all operations in analysis and mapping use data from the workspace. Data columns can be input from and output to text files, entered interactively and displayed on the screen. Data entered into columns directly from the keyboard, or created during a session, must be stored before the session is terminated, otherwise they are lost.

The entire workspace can be saved and retrieved. If several different workspaces are saved in separate files, the contents of the current workspace may be quickly switched.

There is a restriction on the size of the workspace to 50000 numbers. The maximum number of data columns that can be referenced simultaneously is 50. As the workspace becomes full, commands may generate error messages indicating that there is insufficient space to start or complete an operation. The problem can be overcome by selectively erasing unwanted columns or by dividing the data into smaller sections.

SECTION III CONCEPTS

Attribute data

describe the characteristics or attributes of a set of individuals, cases, or spatial features - for example animal or habitation counts, land use, crop or environmental factors. Each attribute is linked with or corresponds to a spatial feature definition - in this package, normally the grid cell of an air survey grid. Attribute data can be stored in files, but are operated on in columns of the workspace.

Locational data

describe or define spatial features by one or more pairs of coordinates. Locational data in this package are of two types. First, the grid cell coordinate consisting of a West-East and South-North sequence number (see PART 1, Section II). Second, strings of coordinates defining linear features - for example, an administrative boundary, river, survey zone limit - which may or may not be in the same coordinate units as the survey grid cells. Strings are not held in the workspace, but are processed from disk files.

Most commands handling locational data use the grid reference - ESTIMATE, MARK and POINT for example; only STRING handles coordinate strings from file.

String files consist of a header (integer values) followed by as many coordinate pairs (integer or real values), separated by spaces, as required to define the feature. All numbers are in free format. The header contains the following information:

- identification number
- dimensionality - normally two, where the data consist of X,Y coordinate pairs
- the total number of coordinates (X plus Y values) used to define the string (twice the number of data points).

For example:

```
1 2 10
0 0 0 12 12 12 12 0 0 0
```

is a coordinate string defining a square.

Entering data

Numeric data contained in text files are entered to the workspace by specifying a filename in the READ command followed by a data column list. For example:

```
READ 'GOURMA.ATT' C1 TO C25
```

reads the data in the file 'GOURMA.ATT' into columns 1 to 25 of the workspace. Data are read in free format - numbers must be separated by a valid separator (space, comma). Data can be split across lines, but data for each new row of the column list must start on a new line in the file. Reading from file can optionally use a FORTRAN-style format specification; this option allows, for example, fields to be skipped or numbers in scientific notation to be read:

```
READ 'ANOMALY.ATT' C1 TO C3 '(10X,3E10.3)'
```

It also allows data stored with no gaps between fields to be read. For example, the following READ commands enter a complete transfer file into the workspace from data file 'TRANS.TXT':

```
READ 'TRANS.TXT' C1 TO C3 '(3F2.0)'        $ east, north, photos

READ 'TRANS.TXT' C5 TO C16 '(7X,6(F2.0,F4.0))'        $ animals

READ 'TRANS.TXT' C18 TO C31 '(44X,7(F2.0,F4.0)'       $ settle

READ 'TRANS.TXT' C33 TO C42 '(87X,10F2.0)'            $ land

READ 'TRANS.TXT' C44 TO C49 '(108X,6F3.0)'            $ crops

READ 'TRANS.TXT' C31 TO C35 '(127X,5F2.0)'            $ infra

READ 'TRANS.TXT' C37 C38 C40 TO C44 '(138X,2F4.0,1X,5F2.0)'
                                                $ grid and zone

READ 'TRANS.TXT' C46 TO C50 '(158X,5F4.0)'            $ environ
```

Notice that:

1. only numeric data are entered - characters cannot be stored in the workspace and are skipped over.
2. these examples assume that all fields of each group of variables have been used or are required for analysis. Where this is not so, the column list and format can be adjusted accordingly.
3. any part(s) of the transfer file can be selected and put into the workspace by means of the format specification. In the above example, columns 31 to 50 are reused (see **Storing data** and **Workspace management** below)

Data can also be read into the workspace from the keyboard. The following command sequence stores a series of data values in C1:

```
READ C1 BEGIN 10 23 43 45 67 78 90 12
10 23 45 67 34 END
```

The BEGIN keyword indicates that data input is to follow on the same line. Data can be split over as many lines as required; commands are not recognised until an END keyword is given to signify that data entry has finished. Control then returns to the command level. The BEGIN keyword can be omitted if data entry is to commence on a new line:

```
READ C1
10 23 43 45 67 78 90 12 10 23 45 67 34 END
```

Data can be entered into more than one column at a time. For example:

```
READ C10 TO C12
1   6  11           $ first row
2   7  12
3   8  13
4   9  14
5  10  15           $ last row
END
```

enters the values 1 to 15 in C10, C11 and C12. Numbers are assigned to columns in the order specified - the following command sequence has the same effect as the previous example:

```
READ C10 TO C12
1 6 11 2 7 12 3 8 13 4 9 14 5 10 15 END
```

Note the relationship between the column list and data list. In the above example three columns are specified. Numbers are picked off the data list and stored in rows of the column list in turn. Thus, the first 3 numbers in the data list are stored in the first row of the three columns, the 4th to 6th numbers in the second row and so on until the data list is exhausted. Columns can be specified in any order, for example:

```
READ C15 C10 C1
1 10 100
2 11 101
3 12 102
END
```

causes C15 to contain 1,2,3; C10 - 10,11,12; C1 - 100,101,102. Numbers are read in free format; integers are converted into real form (i.e. containing a decimal point). It is not possible to enter numbers using scientific (exponential) notation in interactive mode.

Displaying data

Once entered in the workspace, data are displayed using the WRITE command. For example:

```
READ C1 BEGIN 10 5 23 45 67 END
WRITE C1
```

causes the contents of C1 to be displayed as a column down the screen. Multiple columns can also be inspected. For example:

```
WRITE C1 C20 TO C22
```

displays the contents of C1, C20, C21 and C22 row by row down the screen. If data are to be inspected in any quantity or detail, then a text file should be sent ot the printer outside the program.

Storing data

Selected columns can be written to text files using an alternative form of WRITE:

```
WRITE 'SUBSET.ATT' C1 C10 C9
```

This example writes the contents of C1, C10 and C9 to a file called 'SUBSET.ATT'. The data are written using a default format of (6F12.3). Alternative formats can be specified as an input:

```
WRITE 'SUBSET.ATT' C1 C10 C9 '(3F6.2)'
```

Note that format specifications, issued with either READ or WRITE, should be enclosed in quotes and brackets. Users unfamiliar with FORTRAN format specifiers should consult a FORTRAN text for a definition of the syntax. The SEQUENCE command is useful for generating an index number to identify each row of a workspace column when it is written to file.

The entire contents of the workspace can be written to file using SAVE:

```
SAVE 'WORK.SAV'
```

In this form the workspace is saved as an unformatted file for efficiency and cannot be edited, looked at or printed outside the program. Saved files are read back using RETRIEVE:

```
RETRIEVE 'WORK.SAV'
```

Workspace Management

A data set can be split into several working parts, recalled as required by RETRIEVE or READ. Note that READ allows new columns to be read selectively from text files, and placed anywhere in the workspace. RETRIEVE reads the complete saved workspace file, and automatically positions columns as they were when the file was saved. RETRIEVE provides a quicker and more efficient way to handle sub-divisions of a large data set, provided that the grid cell coordinate references are duplicated in each part.

Data columns can be referenced by name as well as number if they have been labelled. For example:

```
READ 'TRIAL.ATT' C1 TO C4
LABEL C1 EAST C2 NORTH C3 CATTLE C4 CATHERDS
DESCRIBE CATTLE CATHERDS EAST NORTH
SAVE 'TEMP.SAV'
```

Labels are saved and retrieved.

Workspace content can be determined at any time by the STATUS command, which shows used columns by their number, length and label (if any).

Columns can be erased - for example when creating subsets of data, or removing temporary columns before saving a workspace. For example:

```
ERASE C10 C11 C16 TO C18 C20
```

removes the specified columns. ERASE C1 TO C50 empties the entire workspace, but can be a very lengthy process.

MANIPULATION AND ANALYSIS

Data held in the workspace can be processed to produce summary statistics; arithmetic manipulation of data; new variables generated by conditional evaluation of expressions; classification of data for summary and mapping; and population and variance estimates.

Summary Statistics

Summary statistics for columns can be generated using the DESCRIBE command. The arithmetic mean, standard deviation, minimum, maximum and count are calculated and displayed in tabular form for all columns specified in the command. For example:

```
READ C1 BEGIN 10 23 45 34 67 89 90 100 23 END
READ C2 BEGIN 44 68 51 29 90 81 22  99 12 END
DESCRIBE C1 C2
```

generates the following output:

COLUMN	MEAN	STAN-DEV	MINIMUM	MAXIMUM	COUNT
1	53.44	29.38	10.00	100.00	9
2	55.11	26.12	12.00	99.00	9

Arithmetic operations

New variables are computed or columns overwritten using the LET command. LET is a powerful facility and uses statements ranging from simple equivalence to complex expressions. For example, LET can be used to sum a set of columns:

```
LET C5 = C1+C2+C3+C4
```

or transform a column:

```
LET C1 = LOGT(C1)
```

or simply copy columns to tidy up the workspace before saving it:

```
LET C10=C23
LET C11=C16
LET C12=C21
```

A full list of the available operations and functions, with examples, is given in the REFERENCE section. They can be summarised under four headings:

ARITHMETIC	(+,-,*,/,**)
LOGIC	(AND,OR,NOT)
RELATION	(GT,GE,EQ,NE,LE,LT)
FUNCTION	LOGE,LOGT,ANTI,SQRT,RECI,ROUN,EXP, ABS,SIN,COS,TAN,ASIN,ACOS,ATAN,MEAN, STAN,COUN,MAXI,MINI,RANG,SUM,SIGN

Expressions are evaluated using FORTRAN-style operator precedence:

```
LET C1 = (C2+C3)*(C4+C5)
```

Logical expressions may also be used:

```
LET C10 = (C1 GT C2) AND (C4 GT C5)
```

Columns generated by logical expressions contain ones and zeros, for true and false respectively. Logical expressions may operate on logical columns as well as produce them:

```
LET C10 = C1 GT C2
LET C11 = C3 LT C5
LET C12 = C10 OR C11
```

Some typical uses of LET:

```
LET C10=SETFUL+NOMFUL+AGROP          $ total sample population

LET C11=C10/(5*0.8608)               $ sample population density

LET C20=CAMELS+CATTLE*0.7+DONKEYS*0.5+SHOATS*0.1      $ find TLU

$ straight line distance of each grid cell
$ from the point (5.5,6.5)
LET C30=SQRT((X-5.5)**2+(Y-6.5)**2)

$ adjust for altitude of the aircraft - mean 800ft
LET CATTLE=CATTLE*800/ALTITUDE
```

New columns using conditional logic

The MAKE command permits the creation of new columns according to conditional logic. The following command:

```
MAKE C5 IF(C1>C2) THEN (3) ELSE (4)
```

sets rows of C5 equal to 3 if C1 is greater than C2, otherwise to 4. The full command syntax allows an unlimited number of ELSEIF clauses. For example:

```
MAKE C4 IF(C1>C2) THEN (3) \
     ELSEIF(C1 EQ C2) THEN (4) \
     ELSE(-99)
```

ELSEIF and ELSE clauses are optional. ELSEIF may be repeated as many times as required. Expressions in ELSEIF or ELSE clauses should be enclosed in brackets. Three distinct forms of the command are possible. Examples:

```
MAKE C10 IF(C1 GT C2) THEN (1)

MAKE C10 IF(C1 GT C2) THEN (1) \
     ELSE (2)

MAKE C10 IF(C1 GT C2) THEN (1) \
     ELSEIF(C1 EQ C2) THEN (2) \
     ELSE  (C2*10.0)
```

Some typical applications of MAKE might be:

```
$ take distance from LET example above;
$ flag grid cells over 5km from point (5.5,6.5):
MAKE C31 IF (C30>5) THEN (2) ELSE (1)

MAKE C50 IF (WATER EQ 0 AND VEGIND < 200) THEN (CATTLE)
LET C49=SUM(C50)        $ total cattle in selected grids
```

Classification

Class intervals are set using the CLASS command as a means of summarising data, and of generating categories required for point symbol maps. Classification allows a discrete number of symbols on a map to represent grouped data.

The following commands read a data column and classify it into four classes:

```
READ C1 BEGIN 15,4,7,9,30,22,23,29,24,10 END
CLASS C1 4 1
```

This form of CLASS selects by default class interval calculation method 1 - equal arithmetic class intervals across the data range. Twelve class interval calculation methods are provided and are listed under CLASS in the REFERENCE section. Alternative classification methods are appropriate for different data distributions, or to achieve a particular type of class distribution for mapping or analysis purposes. The third argument of CLASS is used to select the method required. For example:

```
CLASS C1 4 2    $ rounded arithmetic intervals
```

CLASS generates a table showing class interval values and the number of data values - frequency - assigned to each class. For example:

```
READ C1 BEGIN 10 23 45 34 67 89 90 100 23 END
CLASS C1 6 1
```

generates the following table:

```
CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
  1        10.000       25.000        3     ***
  2        25.000       40.000        1     *
  3        40.000       55.000        1     *
  4        55.000       70.000        1     *
  5        70.000       85.000        0
  6        85.000      100.000        3     ***
EACH * DENOTES     1 OBSERVATION(S)
```

The categories or classes into which each data value is placed by CLASS are numbered sequentially from 1 to n where n is the number of classes requested. These numbers are stored internally for use by the point command. Data values not falling into a class are assigned a class value zero. A maximum of twelve classes can be generated.

A value is assigned to a particular class where it is greater than or equal to the lower bound and less than the upper bound - except for the upper limit of class N. In this case the assignment test is less than or equal to the upper bound. For example:

```
READ C1 BEGIN  1 2 3  END
CLASS C1 2
```

gives the result:

```
CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
  1         1.000        2.000        1     *
  2         2.000        3.000        2     **
```

Note that the value 2.00 is assigned to the second class.

SECTION III CONCEPTS

In some applications automatically generated class intervals are unsuitable because they fall at unrounded or unsuitable points in the data distribution. An alternative form of CLASS allows limits to be set explicitly, for example:

```
CLASS C1 1 5  0,100,200,300,400,500
```

sets the class vector for C1 using the following intervals:

class 1	-	0	to	100
class 2	-	100	to	200
class 3	-	200	to	300
class 4	-	300	to	400
class 5	-	400	to	500

The second parameter indicates that class intervals are being entered for class 1 onwards and the third is the number of classes for which they will be entered. N+1 intervals then follow.

A further form of CLASS can be used with the same result:

```
CLASS C1 5
CLASS 1   0      100
CLASS 2   100    200
CLASS 3   200    300
CLASS 4   300    400
CLASS 5   400    500
CLASS C1
```

The first command of the sequence is used only to set the number of classes. Successive commands set class limits individually. The last command applies the intervals and C1 together to set the class vector.

Population estimates

Statistical estimation procedures are used to calculate population values from sample strip measurements in grid cells. The method is due to Jolly and is described in detail in Jolly, G., 1969, Sampling Methods for Aerial Census of Wildlife Populations, **East Africa Agriculture and Forestry Journal**,35 (special issue). In addition, an alternative estimate of population variance is supplied which in general yields lower values than the Jolly ratio method.

The basic form of the ESTIMATE command assumes a survey grid recorded by two columns of coordinates - West-East and South-North position - with values of a surveyed variable in a matching column. The dimensions of the grid cell - width and length in direction of flight, and width of the sample strip - must be given. For example:

```
ESTIMATE C1 C2 C8 WIDTH 5 5 0.8606
```

yields the following type of table:

Variable 8 CATTLE

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	2475	99	C	11	69509.6	5073.4	7.3	28.08	5073.4	7.3
None	2475	99	R	9	69509.6	5144.6	7.4	28.08	5144.6	7.4

Estimates are computed for flight lines assumed to run in both South-North and West-East directions, denoted by C for column and R for row in the table. If insufficient data are available, an error message is written and the table entry ommitted; this arises if there are (a) too few flight lines, or (b) no observed data values within the designated cells. The table generated by a command can be written to a file using the OUTPUT=<filename> option.

An important feature of ESTIMATE is the ability to produce stratified estimates. The same calculation is employed, but only grid cells falling into a specified subclass are included. Subclasses are integer values, normally 1,2,3...N, contained in a data column. Subclass categories may be existing subdivisions - environmental, land use, zonal characteristics - or can be created from other data. For example:

```
MAKE C20 IF (CULTVAT>25) THEN (2) ELSE (1)
ESTIMATE C1 C2 C8 C20 WIDTH 55 0.8606
```

generates estimates by two subclasses - over 25% cultivated and up to 25% cultivated - as follows:

Variable 8 CATTLE

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
20/ 2	1175	47	C	11	33494.1	3173.8	9.5	28.51	3248.2	9.7
20/ 2	1175	47	R	9	33494.1	4575.5	13.7	28.51	4582.8	13.7
20/ 1	1300	52	C	11	36015.6	3845.9	10.7	27.70	4037.2	11.2
20/ 1	1300	52	R	9	36015.6	3959.6	11.0	27.70	3691.3	10.2

All subclass values present are normally used, but but an option allows only specified classes to be employed. For example:

```
ESTIMATE EAST NORTH HUTS RAIN SUBCLASS 3 5 \
         OUTPUT='RAINHUTS.TXT'
```

generates estimates for rainfall categories 3 and 5.

SECTION III CONCEPTS

MAPPING

This section deals with characteristics and control of the drawing device; positioning and scaling of map output; drawing point symbol maps; and output of key, text and line features on maps.

Drawing device

Drawing commands have no specific knowledge of the graphics device on which output is generated - they are device independent. However, at the graphics interface level the system uses one specific device - the HP7475 plotter - and the user is assumed to be familiar with the basic features of its operation. The maximum effective plotting area available is 375mm in the X-axis and 240mm in the Y-axis when in A3 size setting. Drawing which falls outside this rectangle is automatically clipped or windowed. The plotter can be connected using the HP-IB or RS232 interface protocol - the user must know which type of connection is in use (see APPENDIX).

Before any drawing is attempted, the command DEVICE must be issued and the type of interface specified - incorrect selection will have adverse effects. Strings of characters may appear on the screen if DEVICE has not been given prior to issuing drawing commands.

The HP7475 has six pens available at any one time, permitting choice of colour and line thickness. Selection is made with the PEN command; by the COLOUR keyword in MARK; and with the <colour> argument in SYMBOL. The effect achieved is dependent on the currently installed pens.

STRING draws linear features with a choice of line colour and thickness controlled by PEN.

The STOP command, which terminates a session, should be given in order to stall the current pen and perform a tidy shutdown of the plotter.

Positioning and scaling

All drawing commands assume an (X,Y) coordinate system on the graphics device, with an origin 0,0 in the bottom left corner of the drawing area. Some commands - KEY, MARK <size> option, SIZE, SYMBOL and TEXT - position output or give dimensions in device coordinate units and their arguments are therefore specified in millimetres. For example:

```
SIZE 3.5 4
TEXT 140 35 'Dry season vegetation'
```

results in a 3.5mm by 4mm character box for text, the bottom left corner of the first letter being positioned 140mm from the left edge of the drawing area and 35mm above the bottom edge.

Other commands - MARK, STRING and POINT - assume that locational data are measured in map coordinates and require transformation to ensure correct size and positioning when drawn on the plotter surface. SCALE defines a factor by which these commands multiply coordinates before drawing takes place. The default factor is 1 - no scaling takes place; values over 1 result in enlargement of drawing, values below 1 result in reduction. SCALE is not applied to arguments given in device units. ORIGIN defines the relationship between map position and device position - that is, the location in device coordinate space of a point in map coordinate space. The combined effects of these commands is to establish a transformation applied to all map coordinates before they are plotted. For example:

```
ORIGIN 4000 5000 30 10
SCALE .4
```

reduces the value of map coordinates by a factor of .4 before plotting and places map coordinate (4000,5000) at device position (30,10).

ORIGIN can be used to position multiple maps on a single drawing area. For example:

```
SCALE 5
ORIGIN 1 1 30 20          $ grid cell 1,1 positioned at 30,20
CLASS CATTLE 4 2
POINT EAST NORTH          $ first point map
$
ORIGIN 1 1 30 120         $ grid cell 1,1 positioned at 30,120
CLASS CAMELS 4 2
POINT EAST NORTH          $ second point map
```

Different SCALE and ORIGIN commands can be combined in one composite map if coordinates are derived from different sources - for example, the nominal west-east and south-north coordinate system of air survey grid cells used by POINT, and the map coordinates used to define boundaries or linear features from maps of the survey area, used with STRING. If map coordinates are already in units that match the device drawing space, then one or other or both transformation commands can be omitted.

Point mapping

Data columns representing an attribute at a series of point locations can be mapped by MARK and POINT using symbolism appropriate to the type of data portrayed. The major distinction is that in general MARK takes ready-categorised or discrete data and associates a distinct symbol with each value - for example, different land system, environmental or administrative areas. In contrast, POINT requires that data first be CLASSed, and plots a symbol at each location reflecting the data class into which that point falls. This process is suited to mapping continuous data for which there are no pre-determined categories - the user is free to devise a classification appropriate to the mapping problem in hand.

MARK

plots point symbols at locations defined by two data columns, taking the definition of map symbolism from other columns. The simplest form of the command is:

    MARK EAST NORTH MARKER 4

which positions crosses (marker type 4) at points defined by coordinates in columns EAST and NORTH. Point locations are subject to SCALE and ORIGIN transformation, if defined. A more effective use of the command is to plot up to eight different marker types according to discrete values in a data column - for example:

    MARK C1 C2 MARKER=ADMIN

Marker values outside the range 1 to 8 are ignored and no symbol is plotted. Marker colour can be altered either by specifying a single pen for all symbols, or by giving a column of data values so that pen colour reflects an attribute of each point. Similarly, marker size can be fixed or may vary according to values of an attribute in a data column. In this latter case, to allow symbols to be scaled appropriately, MAXSIZE and optionally MINSIZE can be specified to give a size range to which all markers are scaled. For example:

    MARK C1 C2 SIZE=CATTLE MARKER=5 MAXSIZE=6.5

sets the maximum value of CATTLE equivalent to a square of size 6.5mm and the minimum value to zero - intervening values are scaled accordingly.

No key is generated by MARK, but a key can be constructed by KEYOUT using SYMBOL and KEY definitions associated with POINT.

Careful use of MARK enables more than one attribute to be mapped simultaneously. Symbol type, colour and size can each be used to denote a different attribute. Attribute values may be predefined at time of survey - administrative areas, ecological zones, soil types for example - or can be derived using column manipulation and arithmetic. The MAKE command in particular, used in conjunction with MARK, allows quite complex combinations of environmental factors to be evaluated and mapped.

POINT

takes the result of CLASSing a data colum, together with X and Y coordinates, and plots at each point a symbol representing its data class. Point locations are treated as map coordinates, and are subject to SCALE and ORIGIN transformation before plotting. The default is to use open squares of increasing size to denote increasing values of the mapped variable. Six symbols are pre-defined. For example:

```
CLASS SHOATS 6                 $ default equal-interval classes
POINT EAST NORTH               $ default point symbolism
```

The symbol used for each class can be redefined by SYMBOL, giving control over size, shape, colour and format. For example:

```
SYMBOL 1  5.5   5  3  1
SYMBOL 2  3.75  5  3  1
SYMBOL 3  2.0   5  2  2
SYMBOL 4  3.75  5  7  1
SYMBOL 5  5.5   5  7  1
CLASS  CHANGE  1 5  -15 -5  -2  2  5  15
POINT C1 C2
```

sets up five-class symbolism, using three circle sizes, three colours, and with filled and open format to show decrease, little change, and increase in a variable - assumed to have been previously calculated.

A key is automatically drawn showing the number and type of symbols used, and the numeric values of the class boundaries are used as labelling. Key position is in device units, and is not affected by map coordinate transformation. The number of decimal digits drawn can be set by PLACES or SIZE; the size and style of text is defined by SIZE and/or ITALIC. The default key position is (10,10), but this can be changed by KEY. Numeric labelling can be switched off, and the key labelled with lines of text from a file. For example:

```
SIZE 2.0 2.5 15 3
KEY  180  37  2
KEY LABEL OFF
POINT EAST NORTH
KEYTEXT 'CHANGE.TXT'
```

invokes some of the key drawing options. Point keys can be drawn by KEYOUT independently of the POINT command.

Other map features

Maps can be annotated with the TEXT command, which draws a text string at a position specified in device coordinates. For example:

```
TEXT 56 14 'Vegetation zones'
```

draws text starting at 56mm in X and 14mm in Y, in a character box 1.8mm square, and using pen 1. Text characteristics can be altered by SIZE - changes the width, height and italic angle; by PEN - changes the colour and line thickness; and by ITALIC - changes the italic angle. Text definitions established by these commands apply to all subsequent text until changed, including the drawing of map keys. For example:

```
SIZE 2 2.5
KEY LABEL OFF
POINT C1 C2
```

```
KEYTEXT 'BELLA.TXT'
SIZE 3.5 4 15
PEN 3
TEXT 100 54 'Bella huts - May 1984'
```

Linear features can be added to map output to make it more interpretable - survey zone limits, administrative areas, landscape features for example. Coordinate strings are described in **DATA ENTRY AND STORAGE.** The STRING command, in combination with PEN, enables a variety of linear features to be displayed on one map. All coordinates drawn by string are assumed to be in map units and are subject to SCALE and ORIGIN transformation.

REFERENCE

This chapter provides a reference description of all commands, listed alphabetically. Documentation covers - syntax and definition of parameters; summary description; example(s); and cross-reference(s) to related commands.

The following commands are available:

A. FROM THE COMPUTER OPERATING SYSTEM

AM	**the command needed to start the package from the computer operating system**
MARRIOTT	**starts the alternative population and error estimation programme**

B. WITHIN THE MAPPING AND ANALYSIS PACKAGE

CLASS	class intervals for a data column
DESCRIBE	descriptive statistics for a group of columns
DEVICE	defines the graphical device to be used
ERASE	deletes data column(s)
ESTIMATE	provides population and variance estimates
ITALIC	defines italic angle for text drawing
KEY	sets key position and options
KEYOUTPUT	draws a point map key
KEYTEXT	draws text on a point map key
LABEL	defines label(s) for workspace column(s)
LET	evaluates column expressions
MAKE	forms new data columns using conditional logic
MARK	plots symbols at point locations
OBEY	executes commands contained in a file
ORIGIN	establishes position of map on plotter
PEN	changes the plotter pen
PLACES	number of decimal places for point map key
POINT	plots symbols at point locations
READ	enters data to workspace from file or keyboard
RETRIEVE	restores a saved workspace
SAVE	saves the entire workspace
SCALE	defines scale factor applied to map coordinates
SEQUENCE	generates regular number sequence
SIZE	sets text characteristics, and decimal places for key
STATUS	displays information on data in the workspace
STOP	terminates a session
STRING	coordinate string drawing facility
SYMBOL	defines symbols subsequently drawn by POINT command
TEXT	draws text string
WRITE	writes data column(s) to screen or file

CLASS

Generates class intervals for a data column:

```
    CLASS  <Cn>  <kclass>  [<klims>]  !
           <class>  <lower>  <upper>  !
           <Cn>  <1st class>  <n of classes> <n+1 limits> !
           <Cn>

    <Cn>           - data column to be classed
    <kclass>       - number of classes required
    <klims>        - method used to calculate class intervals
    <class>        - class number to be defined
    <lower>        - lower limit of class
    <upper>        - upper limit of class
    <1st class>    - specifies the start of a group of class limits
    <n of classes> - number of classes to be defined - must be in
                     the range 1 to 12.
    <n+1 limits>   - n+1 class limits. First number is the lower
                     limit of <first class>, second number divides
                     <first class> from <first class+1> and
                     last number the upper limit of class <n>.
```

The value of <klims> determines which of the following class interval calculation methods is used:

1 equal arithmetical intervals across range
2 equal rounded arithmetical intervals
3 curvilinear progression of limits across range
4 geometrical progression of limits from zero
5 geometrical progression of class widths
6 arithmetical progression of class widths
7 equal intervals on reciprocal scale
8 equal intervals on trigonometrical scale
9 intervals based on normal percentiles
10 intervals proportional to standard deviates
11 quantile based intervals
12 nested means intervals

CLASS is designed for maximum flexibility in the way that class intervals are generated for a data column. The command has four forms (see examples):

1. generates class intervals automatically - using one of the twelve interval selection methods listed above. If a selection method is not specified, method 1 is used.

2. sets class limits for individual classes. Thus, n successive CLASS commands are required to set limits for n classes. The number of classes must first be set, and the defined classes finally applied to a column.
3. similar to method 2, but sets a series of class limits in a single command
4. applies previously defined class limits to a given data column. It is required after using form (2) of CLASS.

CLASS must be given before the POINT command can work - each point symbol is associated with the class of a data value. Any data values not falling into a class are assigned a class of zero and are ignored by POINT. A maximum of twelve classes can be defined.

Examples:

1
```
  $$$$$$ automatically generated intervals
READ 'ETH84.ATT' C1 TO C4
CLASS C4 6 2  $$ uses classification method 2
POINT C1 C2
```

2
```
  $$$$$$ setting individual classes
READ 'ETH84.ATT' C1 TO C4
CLASS C4 4    $$ must define number of classes
CLASS 1 10.0 20.0
CLASS 2 20.0 30.0
CLASS 3 30.0 40.0
CLASS 4 40.0 50.0
  $$$$$$ now apply classes using form 4
CLASS C1
POINT C1 C2
```

3
```
  $$$$$$ classes of example 2 set by a single command
READ 'ETH84.ATT' C1 TO C4
CLASS C4 1 4 10 20 30 40 50
POINT C1 C2
```

4 see final CLASS command in example 2

See MAKE, POINT and SYMBOL commands.

NOTE
A maximum of 2500 **rows** can be classed at once.
Before defining the class intervals, the number of classed must first be assigned (e.g. CLASS C4 4 as above).

DESCRIBE

Provides descriptive statistics for a group of columns:

DESCRIBE <Cn>...<Cn>

<Cn>...<Cn> - column group

DESCRIBE provides a table of summary statistics for the specified column group. The arithmetic mean, standard deviation, minimum, maximum and count are displayed. All column values are included in the count and calculation of statistics.

Example:

```
$ restore data set and examine its data columns
RETRIEVE 'MALIDRY.SAV'
DESCRIBE C1 TO C30
```

See CLASS, ESTIMATE, LET and MAKE commands.

DEVICE

Specifies the type of graphical device to be used:

DEVICE

Initialises the graphical device after ascertaining from the user which type of interface is to be used (HP-IB parallel or RS232 serial). If DEVICE is not given before issuing drawing commands, then strings of (meaningless) characters are written to the screen. The drawing device is normally a plotter. STOP releases the device and ends a session.

See drawing commands and STOP.

NOTE
The DEVICE command cannot be included within an obey file.

ERASE

Deletes data columns:

```
ERASE <Cn>...<Cn> ALL

<Cn> - column(s) to be deleted
```

Used to delete data columns in the workspace. The current contents of the workspace can be determined using the STATUS command. ERASE deletes one or more specified columns, or all columns if the range C1 to C50 is given. In the latter case, the entire contents of the workspace are erased.

Examples:

```
$ save selected columns of complete data set
RETRIEVE 'WHOLE.SAV'
ERASE C4 TO C7 C10 C15 TO C19
SAVE 'PART.SAV'
```

See the READ, RETRIEVE, SAVE, STATUS and WRITE commands.

ESTIMATE

Provides population and variance estimates for a data column:

```
ESTIMATE <Cx> <Cy> <Cn> [<Cs> [SUBCLASS=<value...value>]]
         WIDTH=<cell width> <cell length> <strip width>
         [OUTPUT=<filename>]
```

```
<Cx>               - column of x-coordinates for grid cells
                     (west-east map coordinates)
<Cy>               - column of y-coordinates for grid cells
                     (south-north map coordinates)
<Cn>               - column variable for which estimates are
                     required
<Cs>               - optional stratification variable to provide
                     estimates by subclass
SUBCLASS=
  <value..value>   - optional method for specifying subclasses
WIDTH=
  <cell width>     - total width of survey grid cell
  <cell length>    - length of survey grid cell along sample strip
  <strip width>    - width of observed sample strip within grid cell
OUTPUT=
  <filename>       - optional filename for tabular output
```

Generates estimated population and density, with two alternative variance estimates, for a specified air survey variable. Estimates are produced for both row (R) and column (C) directions in the grid - normally west-east and south-north. Cell size and sample strip width must be supplied, and are measured in the same units - for example kilometres. An option allows stratification of estimates by subdividing grid cells by group, denoted by a column of integer values <Cs>. Estimates are produced for each subclass value in <Cs>. The SUBCLASS option allows selected values contained in <Cs> to be used for stratified estimates. If OUTPUT is used, then the table of estimates generated by a single command is stored in <filename>.

Examples:

```
$ unstratified estimate
RETRIEVE 'ABET.SAV'
ESTIMATE EAST NORTH CATTLE WIDTH 5 5 0.8608

$ stratification on newly created 2-class variable
RETRIEVE 'ABET.SAV'
MAKE C20 IF(CULTVAT>25) THEN (1) ELSE (2)
```

```
ESTIMATE C1 C2 C8 C20 WIDTH 5 5 0.8608

$ stratification on two selected rainfall zones
RETRIEVE 'GOURMA.SAV'
ESTIMATE C1 C2 CAMELS RAIN SUBCLASS 2 4 \
         WIDTH 9.053 9.053 0.8

$ stratified estimate by pastoral pressure, output
$ in file 'GOURMA.TB1'
RETRIEVE 'GOURMA.SAV'
ESTIMATE EAST NORTH TLU PASTPRES \
   WIDTH 9.053 9.053 0.8 \
   OUTPUT='GOURMA.TB1'
```

See CLASS, DESCRIBE, LET and MAKE commands.

To estimate the Standard Errors for pecentage measurements, such as **grass** cover, assign a width and length of 1, with a sample band of the sample proportion (e.g 0.095). The resulting %SE will be valid, but the density shown should be multiplied by thesample band width given (i.e 0.095).

An alternative method for eliminating both population levels and standard errors is to use the MARRIOTT command from within the computer operating system. This command cannot be used from within the mapping and analysis package, and is described in Appendix 2 on page 68.

ITALIC

Defines italic angle for text drawing:

```
ITALIC <angle>

<angle> - italic angle in degrees
```

Text is drawn italicised at the angle given by this command until the angle is redefined by SIZE or another ITALIC command. Default angle is zero. This command allows italic angle to be reset without changing other text characteristics.

Example:

```
$ draw main title
SIZE 3 4 0 2
TEXT 10 50 'Water points used in 1984 dry season'
$ italicise sub-title
ITALIC 15
TEXT 10 43 'WoDaaBe only'
```

See SIZE and TEXT commands.

KEY

Sets key position and options:

```
KEY  <xkey> <ykey> [<gap>] !
     <ON ! OFF> !
     LABEls <ON ! OFF>
```

```
<xkey>       - x coordinate of bottom centre of key (device units)
<ykey>       - y coordinate of bottom centre of key (device units)
<gap>        - gap between successive symbols in key
<ON ! OFF>   - turns key drawing on or off
LABEls
<ON ! OFF>   - turns key labelling on or off
```

Used to set key position and symbol spacing. The number of decimal places shown on numeric-labelled keys is specified by the PLACES or SIZE command. Key drawing is performed by the POINT command when mapping, or by KEYOUTPUT if only a key is required. Options are provided for turning key drawing on or off and key labelling on or off. In the latter case, keys can be labelled with text from a file using KEYTEXT. The default position is (10,10) if no KEY command is given.

Examples:

```
  $$$$$$ position key with 3 unit gap between symbols
RETRIEVE 'WETSURV.SAV'
CLASS C4 6 2
KEY 120 100 3
POINT C1 C2

  $$$$$$ key text labelling option
RETRIEVE 'WETSURV.SAV'
CLASS CATTLE 4
KEY LABEL OFF
POINT C1 C2
KEYTEXT 'CATTLE.TXT'
```

See the KEYOUTPUT, KEYTEXT, PLACES, and POINT commands.

KEYOUTPUT

Draws a point map key:

```
KEYOUTPUT
```

Causes a point map key to be drawn immediately without a map. A CLASS command must be given for a meaningful key to be drawn. The key is positioned using the KEY command, while its graphic style is determined by KEYTEXT, SIZE, SYMBOL and PLACES. This facility is useful for constructing point symbolism keys where symbol type, size, or colour need to have individual keys. Symbolism schemes can also be tested without drawing a complete map.

Example:

```
$ sets open and filled circle symbols and tests the key
KEY 100 20 2
SYMBOL 1 3.5 5 2 1
SYMBOL 2 3.5 5 2 2
CLASS C5 2
KEYOUTPUT
```

See CLASS, KEY, KEYTEXT, PLACES, POINT, SIZE and SYMBOL commands.

KEYTEXT

Draws text on a map key in place of numeric labelling:

```
KEYTEXT <filename>

<filename>  a text file containing labels for map key
```

Allows numeric labelling of keys to be replaced with text from a file, created using the system editor. Each text line in the file corresponds to one class or symbol of the key. Standard key labelling must be switched off.

Example:

```
$ use text labelling of key from file
CLASS CATTLE 4 2
KEY LABEL OFF
POINT C1 C2
KEYTEXT 'WETCAT.TXT'
```

where the file 'WETCAT.TXT' might contain:

```
under  0.5
0.5 to 2.0
2.1 to 5.0
over 5.0
```

See KEY, KEYOUTPUT and POINT commands.

NOTE
The key text cannot be positioned independently of the key output. Thus a symbol key must be positioned and drawn before any previous location of the key text is invalidated.

LABEL

Defines labels for columns of the workspace:

```
LABEL <Cn> <string> [{<Cn> <string>}]

<Cn>     - column number to be labelled
<string> - label for column
```

Once defined, column labels can be used in place of column numbers in any commands referencing the workspace. Labels can be up to eight characters long - further characters are not significant. Labels are SAVEd and RETRIEVEd, and displayed by STATUS.

Examples:

```
$ set and use column labels
LABEL C3 CATTLE C4 CATHERDS C20 HERDSIZE
LET HERDSIZE=CATTLE/CATHERDS
WRITE HERDSIZE


$ use previously saved labels
RETRIEVE 'RIVERINE.SAV'
STATUS
LET C8=(FLOW-MEAN(FLOW))/STAN(FLOW)   $ standardization
CLASS C8 10 9
POINT XSTATN YSTATN
```

See READ, RETRIEVE, SAVE, STATUS and WRITE commands.

NOTE
When columns are ERASED, the labels are note affected, and so remain valid even though they are not displayed by the STATUS command. Thus if a new column is READ into the workspace, the old label will reappear. This means that if the new column is relabelled with the same name as a previously erased column, an error message will appear on the screen (DUPLICATE LABEL). This will not invalidate the labelling of the new column, and is intended to be a remined of previous labels only.

LET

Evaluates column expressions:

```
LET <Cn>=<expression>

<Cn>         - destination column
<expression> - any valid FORTRAN-style arithmetic
               or logical expression
```

Evaluates column expressions to form new columns or overwrite existing ones. Expressions may contain columns, constants, functions, parentheses and operators. Parentheses are used to override the normal order of expression evaluation. FORTRAN-style logical and relational operators are permitted. Logical operators return 1 for TRUE and 0 for FALSE and may only be applied to columns consisting of ones and zeros. The full list of valid operators and functions follows, with a short example of their use:

Arithmetic	-	(+,-) unary plus and minus	(C3=-C2)
		(+,-) binary addition and subtraction	(C3=C1+C2)
		(*,/) binary multiplication and division	(C3=C1*C2)
		(**) exponentiation	(C3=C1**C2)
Logical	-	(OR or !) logical inclusive OR	(C3=C1 OR C2)
		(AND or &) logical AND	(C3=C1 AND C2)
		(NOT) logical negation	(C3= NOT C1)
Relational	-	(GE or >=) greater than or equal to	(C3= C1 GE C2)
		(GT or >) greater than	(C3=C1 GT C2)
		(LE or <=) less than or equal to	(C3= C2 LE C1)
		(LT or <) less than	(C3=C1 LT C2)
		(NE or <>) not equal	(C3=C1 NE C2)
		(EQ) equal	(C3=C1 EQ C2)

Functions	-	LOGE	log to the base e	C2=LOGE(C1)
		LOGT	log to the base ten	C2=LOGT(C1)
		ANTI	anti-logarithm	C2=ANTI(C1)
		SQRT	square root	C2=SQRT(C1)
		RECI	reciprocal of a number	C2=RECI(C1)
		ROUN	round values to nearest integer	C2=ROUN(C1)
		EXP	exponential	C2=EXP(C1)
		ABS	absolute of a number	C2=ABS(C1)
		SIN	sine of angle in radians	C2=SIN(C1)
		COS	cosine of angle in radians	C2=COS(C1)
		TAN	tangent of angle in radians	C2=TAN(C1)
		ASIN	arcsine of angle in radians	C2=ASIN(C1)

ACOS	arccosine of angle in radians	C2=ACOS(C1)
ATAN	arctangent of angle in radians	C2=ATAN(C1)
MEAN	mean of column	C2=MEAN(C1)
STAN	standard deviation of column	C2=STAN(C1)
COUN	count of column rows	C2=COUN(C1)
MAXI	maximum value in column	C2=MAXI(C1)
MINI	minimum value in column	C2=MINI(C1)
RANG	range of a column	C2=RANG(C1)
SUM	sum of column	C2=SUM(C1)
SIGN	sign - (minus=-1,zero=0,plus=1)	C2=SIGN(C1)

The MAKE command also uses LET-style expressions.

Examples:

```
  $$$$$$ arithmetic on columns
READ 'DATA.ATT' C1 TO C4
LET C5=C1+C2+C3+C4
LET C6=((C3/C4)*(C1-C2))**1.73

  $$$$$$ using functions
READ 'DATA.ATT' C11 C12
LET C3=MEAN(C11)
LET C4=LOGT(C11)
LET C5=STAN(C12)
LET C6=SQRT(C12*2+C11)
WRITE C3 TO C6
```

See the CLASS, DESCRIBE, ESTIMATE and MAKE commands.

NOTE
The exponential command works using logarithms. Therefore, if there are any zero values in the data column, an error message will be shown. To overcome this, if, for example, you wish to square the values in a column, use the following command:
LET Cn=Cn*Cn

MAKE

Forms new data columns using conditional logic:

```
MAKE <Cn> IF <expression> THEN <expression>
     [ ELSEIF <expression> THEN <expression> ]
     [ ELSE   <expression> ]
```

```
<Cn>    - column to be created or overwritten
IF      - logic keyword used before first logical expression
ELSEIF  - logic keyword used before subsequent logical
          expressions
THEN    - logic keyword used before result column expressions
ELSE    - logic keyword used before expression for all
          remaining cases
```

Expressions given after IF or ELSEIF keywords must return a logical or true/false result so that THEN and ELSE actions can be taken. Expressions given after THEN and ELSE keywords can be any valid logical or arithmetic expression. All expressions should be enclosed in brackets for safety and clarity. Expression syntax follows the same conventions as for the LET command. Note that any number of 'ELSEIF <exp> THEN <exp>' sequences can be given. These can be placed on subsequent lines using the continuation (\) character.

Examples:

```
$ sets two-fold classifiaction
RETRIEVE 'WET.SAV'
MAKE C7 IF(C3 GT 2500) THEN (1) ELSE (0)
CLASS C7 2

$ sets result to be function of other column(s)
RETRIEVE 'WET.SAV'
MAKE C20 IF(C1<10) THEN (C11*10) ELSE (C11*100+ABS(C5))

$ extended form of MAKE giving 4 possible outcomes
READ 'DATA.ATT' C1 TO C15
MAKE C6  IF( C1+C2 GE C3+C4 ) THEN LOGT(C11)   \
     ELSEIF( C8+C9 GT C7    ) THEN LOGT(C12)   \
     ELSEIF( C14 > C15      ) THEN LOGT(C13)   \
     ELSE LOGE(C14)
```

Doesn't work!

See the CLASS, DESCRIBE, ESTIMATE and LET commands.

NOTE
Complex MAKE commands use a large number of temporary columns within the workspace. If insufficient room is available, then break the command down into smaller units. The resulting columns may be added together to create the desired final column.

The MAKE command may also be used to change single values within a column if an error in data entry is found. Thus to alter the value in column 4 recorded at coordinate 3,3 from 1 to 2, use the commands (assuming C41 is empty):

```
MAKE C41 IF(C1 EQ 3 AND C2 EQ 3) THEN (2) ELSE (C4)
LET C4=C41
```

MARK

Plots symbols at locations defined by two columns:

MARK <KEYWord>

The following keywords are available:

X=<Cn>	- column of x coordinates (map units)
Y=<Cn>	- column of y coordinates (map units)
SIZE=<size>!<Cn>	- size of all markers or column of sizes
MARKer=<imar>!<Cn>	- marker number used for all markers, or column of marker values
COLOur=<ipen>!<Cn>	- colour or pen number for all markers or column of colour numbers
MAXSIZE=<size>	- maximum size for markers; scales all markers so that the largest is <size>. The column given with SIZE=<Cn> provides values for scaling markers - the largest value in the column is assigned a marker of MAXSIZE. If MINSIZE is not specified, zero is assumed, and markers are scaled from MAXSIZE to zero.
MINSIZE=<size>	- minimum size for markers - used in conjunction with MAXSIZE to scale markers. When MINSIZE and MAXSIZE are given together the SIZE column is treated as data so that markers vary in size from MINSIZE to MAXSIZE.

Markers available:

1	- triangle (point up)
2	- triangle (point down)
3	- plus sign
4	- cross
5	- square
6	- diamond
7	- asterisk
8	- hexagon

This is a general purpose facility for displaying point data using pre-defined symbols. It is particularly useful for displaying large numbers of points where the classification and symbol definition facilities linked to the POINT command are not required. Symbol size, type and colour can be held constant or varied by specifying a column containing sizes, marker types or colours for each point. If the

MINSIZE and MAXSIZE keywords are used the column specified with the SIZE keyword is treated as a data column, markers being scaled in the range MINSIZE to MAXSIZE.

Examples:

```
$ draw grid points using a plus of size 2
LABEL C1 EAST C2 NORTH
MARK EAST NORTH SIZE=2.0 MARK=3

RETRIEVE 'GOURMA.SAV'
$
$ draw squares proportional to TLU, maximum size 3
MARK C1 C2 SIZE=TLU MARK=5 MINSIZE 0 MAXSIZE 3
$
$ add symbol to show presence or absence of water
$ stored as 1 and 0 in data column WATER
MARK C1 C2 SIZE=1.5 COLOUR 3 MARK=WATER
```

See the ORIGIN, POINT and SCALE commands.

OBEY

Executes commands contained in a file:

OBEY <filename>

<filename> - file containing commands

OBEY causes commands contained in the supplied file to be executed as if entered from the keyboard. Control passes back to the previous command source after the last command in the file is executed. If the file contains a STOP command, execution terminates. OBEY commands may themselves be used in the supplied file, nested to five levels. OBEY files are created using the system editor.

Example:

```
$ uses obey file of commands to generate standard map
RETRIEVE 'ABET.SAV'
CLASS CATTLE 4 2
TEXT 10 20 'Cattle distribution: wet season 1983'
OBEY 'MAPABET.OBY'
```

ORIGIN

Establishes positional relationship between map coordinates and device coordinates:

```
ORIGIN  <xmap>  <ymap>  <xdev>  <ydev>

<xmap>    - X-coordinate of map position (map units)
<ymap>    - Y-coordinate of map position (map units)
<xdev>    - device X-coordinate corresponding to <xmap>
            (device units)
<ydev>    - device Y-coordinate corresponding to <ymap>
            (device units)
```

Positions an arbitrary map position on the drawing device; normally given before any mapping command. Default values are xmap,ymap=0 and xdev,ydev=0. Can also be used to position individual maps within a composite drawing. Several ORIGIN commands can be given without changing SCALE.

Example:

```
$ map position 4000,2000 corresponds to device position 10,20
ORIGIN  4000 2000  10 20

$ positions two side by side maps on a page
ORIGIN 1 1 10 10
CLASS CATTLE 4
POINT C1 C2
ORIGIN 1 1 200 10
CLASS SHOATS 4
POINT C1 C2
```

See the SCALE command.

NOTE
The map coordinate 1,1 refers to the centre of grid cell 1,1. Thus the bottom left had corner of rectangular survey area will be at map coordinates 0.5, 0.5.

PEN

Changes the device pen:

```
PEN <itype>

<itype>  - number to select pen type or colour
```

Changes the device drawing instrument - for example a plotter pen - usually to alter colour or line thickness. Pen attributes are a function of the pens currently installed in the plotter. The selected pen remains in use until changed by the user or by a plotting command. Numeric codes for <icolour> are device specific - these are normally integers in the range 1 to n where n is the number of available pens or colours.

Example:

```
PEN 2
SIZE 2 3 15
TEXT 15 100 'Water points shown as red'
PEN 3
TEXT 15 95 'Veterinary stations as blue'
```

See MARK and SYMBOL commands.

PLACES

Specifies number of decimal places for a point map key:

PLACES <number>

<number> number of decimal places drawn on numeric labelled key

Redefines the number of decimal places drawn when using numeric key labelling without giving a SIZE command. Default is no decimal place figures drawn.

Example:

```
$ adjusts numeric key to show low density values
LET C10=CAMELS/81.47
CLASS C10 4
PLACES 2
POINT EAST NORTH
```

See the KEY, KEYOUTPUT, POINT and SIZE commands.

POINT

Plots symbols at points:

```
POINT <Cx> <Cy> !

<Cx> - column containing X-coordinate of points (map units)
<Cy> - column containing Y-coordinate of points (map units)
```

Maps a single classed variable, using the currently defined symbols corresponding to each class. A CLASS command must be given before point can work. The default symbol is an open square; this can be redefined with the SYMBOL command. A key is drawn (default position 10,10) unless KEY OFF is in effect. Keys use numeric labelling of class intervals, unless KEY LABEL OFF and KEYTEXT are used. PLACES or SIZE changes the numeric detail shown on the key.

Example:

```
$ default point map
READ 'DATAFILE.ATT' C1 TO C3
CLASS C3 8 12
POINT C1 C2

$ change some default map characteristics
SYMBOL 1 3 5 4 1
SYMBOL 2 3 5 4 2
KEY 20 150 3
KEY LABEL OFF
CLASS C4 2
POINT C1 C2
KEYTEXT 'DRY84.TXT'
```

See the CLASS, KEY, KEYOUTPUT, KEYTEXT, PLACES, SIZE and SYMBOL commands.

READ

READ takes considerably longer than RETRieve. To READ 30 columns of a 1000 grid dataset can take up to 3 or 4 hours on a HP150 and will be even slower on an IBM.

Enters data to workspace from file or keyboard:

```
READ [<filename>]  <Cn>...<Cn>   [<format>] !
     <Cn>...<Cn> [BEGIN] <datalist> END
```

<filename>	- name of input data file
<Cn>	- column number(s) in the workspace
<format>	- FORTRAN-type format used to read data from file; read in free format if none is given
[BEGIN]	- specifies start of data entry from keyboard (optional if data starts on new line)
<datalist>	- list of data values separated by spaces or commas. The datalist may extend over as many lines as required
END	- terminates the data list

If no filename is supplied data are read as typed from the keyboard until an END is encountered. Data supplied in this way must be in free format - that is each datum is separated by a space or comma. The BEGIN keyword is optional if the data list starts immediately on a new line. Data read from file are assumed to be in free format unless an optional FORTRAN-style format is supplied. Format specifications should be enclosed in quotes.

Examples:

```
$ reads separately two columns of seven numbers typed at keyboard
READ C1
10 20 102.46 34 29 55 6.8 END
READ C2 BEGIN 1 1 3 2 2 3 1 END

$ reads ten columns from file : two lines per input record
$ skips part of record 2
READ 'NEBU.ATT' C1 TO C4 C13 C14 C10 TO C6 '(6F8.2/13X,5F10.3)'

$ reads 20 columns in free format from file
$ note: values must be separated by a space or comma
READ 'DRY.ATT' C1 TO C20
```

See LABEL, RETRIEVE, SAVE, STATUS and WRITE commands.

NOTE
READ can be used to combine several columns from different datasets into one working data file, thus:

```
READ 'AAAA.TXT' C1 to C8
READ 'BBBB.TXT' C9 to C15
```

The resulting working file will contain data from both AAAA and BBBB.

If the dataset is in memory and the command READ Cn to Cm is given, then those columns will be read into memory, and the other columns lost.

RETRIEVE

Restores a saved workspace:

RETRIEVE <filename>

<filename> - a file used previously to save the workspace

The complete contents of a workspace, including column labels, are restored to memory from file, having been previously stored by a SAVE command. RETRIEVE only works on the special type of file created by SAVE. Files that are RETRIEVEd are entered much more quickly than with READ, and have the added advantage of retaining previously labelled data columns.

Example:

```
$ restore and look at a file: output selected columns
RETRIEVE 'WATER.SAV'
STATUS
WRITE 'SOMEWAT.ATT' EAST NORTH OWNER TYPE
```

See the READ, SAVE, STATUS and WRITE commands.

SAVE

Saves the entire workspace:

SAVE <filename>

<filename> - file containing saved workspace

The complete contents of the workspace, including column labels, are saved. The file is written in internal format, which is much faster than WRITE. The file can only be read again by RETRIEVE, and cannot be edited or altered in any way. Files that are RETRIEVEd are entered much more quickly than with READ, and have the added advantage of retaining previously labelled data columns.

Example:

```
$ reads data file and saves it for rapid retrieval
READ C1 TO C23 '(14F5.1/6F8.3,3F2.0)'
LABEL C1 XABET C2 YABET
STATUS COLUMNS
SAVE 'ABET.SAV'
```

See the READ, RETRIEVE, STATUS and WRITE commands.

SCALE

Defines a scale factor applied to map coordinates before drawing:

```
SCALE  <factor>

<factor>  - a number by which all coordinates are multiplied
            before plotting
```

All map coordinate data - air survey grid positions and coordinate strings, for example - are scaled by this factor, until it is reset. Default factor is 1 - no scaling. The scale factor is not applied to positions given in device coordinates by commands such as KEY and TEXT. SCALE commands can be given without changing ORIGIN, and vice-versa.

Example:

```
$ double the point map size
SCALE 2.0
POINT C12 C13

$ drawing takes place at one-fifth previous scale
SCALE 0.2
STRING 'FOREST.STR'
```

See the ORIGIN command.

SEQUENCE

Generates a column of numbers in a regular sequence:

```
SEQUENCE <Cn> <n> !
         <Cn> <start> <end> [<increment>]

<Cn>        - destination column
<n>         - number of values in sequence
<start>     - start value of sequence
<end>       - end value of sequence
<increment> - optional increment
```

SEQUENCE generates a regular number sequence. The command has two forms. If a destination column is given with a sequence length <n>, numbers are generated from 1 to <n> with an increment value of 1. If sequence start and end values are specified, the sequence runs from <start> to <end> with an increment of 1 - unless <increment> is given.

Examples:

```
SEQUENCE C5 100        $ numbers 1 to 100 put in column 5
SEQUENCE C6 10 20 2    $ numbers 10,12,14,16,18,20 in column 6
```

See WRITE command.

SIZE

Sets character width, height and italic angle for text, and decimal places for key:

```
SIZE  <char width>  <char height>  [<angle>]  [<places>]

<char width>     - character width (device units)
<char height>    - character height (device units)
<angle>          - clockwise angle from the vertical for italics
                   (degrees)
<places>         - number of decimal places drawn on numeric
                   labelled keys
```

Definitions established by SIZE remain in effect for all text drawing, including map keys, until altered by a further command. Character size is not affected by SCALE commands. Italic angle and number of decimal places can also be set by ITALIC and PLACES, without the need to specify other text characteristics. Default values are:

```
<char width>      1.8
<char height>     1.8
<angle>           0.0
<places>          0
```

Examples:

```
$ sets text characteristics for title
SIZE 4 5.25 12
TEXT 20 37 'Cattle Distribution'
ITALIC 0
TEXT 20 30 'Dry Season'


$ set characteristics for key drawing
SIZE 2.5  3  0  2
POINT C5 C6
```

See ITALIC, PLACES and TEXT commands.

STATUS

Displays information on data currently in the workspace:

```
STATUS
```

STATUS shows which columns of the workspace contain data by giving their column number, length and label if one is set.

Example:

```
RETRIEVE 'NIGER.SAV'
STATUS
```

See LABEL, READ, RETRIEVE and WRITE commands.

STOP

Terminates a program session:

```
STOP
```

Performs a tidy end to a session, including the release of any graphical device currently in use. All data in the workspace are lost when STOP is given.

See SAVE and WRITE commands.

STRING

String drawing facility:

STRING <filename>

<filename>   file containing coordinate string(s)

STRING draws coordinate strings from a specified file. Line thickness and colour may be altered by PEN if suitable drawing pens are available. Strings are assumed to be in standard format and are scaled according to the current factor set by SCALE.

Example:

```
$ draw strings from file
$
PEN 7
STRING 'RESERVE.STR'
STRING 'LIMITS.STR'
PEN 2
STRING 'RIVER.STR'
```

See PEN and SCALE commands.

The dots on distribution maps are plotted and centred on the coordinates given. Thus to draw a square around a 10x10 grid area, with the origin at 0,0, the required string is 0.5 0.5 0.5 10.5 10.5 10.5 10.5 0.5 0.5 0.5.

SYMBOL

Defines symbols subsequently drawn by POINT command:

```
SYMBOL  <class>  <size>  <type>  <colour>  <format> !
        <.null.>
```

```
<class>   - class to which symbol is linked
<size>    - symbol fitted into box with sides of length <size>
            (device units)
<type>    - number indicating symbol shape
<colour>  - integer defining pen number
<format>  - symbol format
<.null.>  - no argument shows current symbols
```

SYMBOL changes the size, type, colour and format of symbols linked to specified classes and used by POINT. Without an argument, currently defined symbols are shown in the table form (see defaults below).

SYMBOL <type> refers to shape as follows:

1 - square
2 - point-down triangle
3 - point-up triangle
4 - diamond
5 - circle

Symbol <colour> can be altered provided an appropriate pen is available corresponding to the number specified. Symbol <format> is a number specifying how the symbol is drawn, as follows:

1 - solid filled
2 - open
3 - graduated

Graduated symbols are open and show all the symbols of the current class and below - for example default symbol 4 with format 3 would show three successively smaller squares nested inside the class 4 square.

The following are the default symbols:

class	size	type	colour	format
1	2.5	1	1	2
2	3.0	1	1	2
3	3.5	1	1	2
4	4.0	1	1	2
5	4.5	1	1	2
6	5.0	1	1	2

Example:

```
$ defines symbols for classes 1 to 3
SYMBOL  1  3.0  5  2  1
SYMBOL  2  4.5  5  2  1
SYMBOL  3  6.0  5  2  1
```

sets up circles (type 5) of increasing size (3 to 6 device units). Symbols are of colour 2 and are drawn filled (format 1).

See the CLASS, PEN and POINT commands.

TEXT

Draws text string:

TEXT <xpos> <ypos> <text string>

<xpos> - X coordinate of bottom left corner of text string (device units)

<ypos> - Y coordinate of bottom left corner of text string (device units)

A text string containing any combination of printing characters, and enclosed in quotes, is drawn at device position (X,Y). Text position is not affected by current SCALE or ORIGIN definitions. Text output is subject to any character size and italic angle definitions previously given by SIZE or ITALIC.

Examples:

```
SIZE 5 6 15
TEXT 10 160 'Calatropis procera'
SIZE 2.5 2.5 0
TEXT 10 155 'Distribution during 1984 dry season'
```

See the ITALIC and SIZE commands.

WRITE

Writes the contents of one or more data columns to the screen or to a file:

```
WRITE  [<filename>]  <Cn>...<Cn>  [<format>]

<filename>  - name of output file
<Cn>...<Cn> - group of column numbers
<format>    - FORTRAN-type format used to write data values
```

If no <filename> is given, data are displayed on the screen. The default format is (6F12.3). A user-specified format shpuld be used when numbers are too large or too small to be successfully written with the standard format.

Examples:

```
WRITE C3    $ single column to screen
$
$ selected columns to data file
WRITE 'SUBSET.ATT' C1 TO C6 C8 C10 TO C15 '(11F6.3)'
$
WRITE C1 TO C4 '(4F20.2)'   $ four formatted columns to screen
```

See the READ, RETRIEVE, SAVE and STATUS commands.

WRITE takes considerably longer than SAVE. On a large dataset, the available memory on many 'micros' will place an effective limit on the number of columns that can be WRITten at once of approximately 20. WRITing this many columns can take up to 5 hours for a dataset of 1000 rows.

CONFIGURING THE HP7475 PLOTTER

Plotter with HP-IB interface

The HP-IB interface cable should connect the HP-IB adaptor at the rear of the HP-150 with the interface socket of the plotter. The interface switches at the right rear side of the plotter should be set as follows:

US	A3	16	8	4	2	1
1	1	0	0	1	0	1
MET	A4	--------	ADDRESS	---		

The software will then reference the plotter as PLT: (HP-IB address 7)

Device Configuration

The software for this package is written to the Auxiliary Port. Thus it is essential to ensure that the AUX configuration is set to Port 1. Otherwise the plotter will not work. This configuration should be set when the computer is first turned on, by selecting DEVICE CONFIG from the opening screen, and pressing START APPLIC.

Plotter with RS-232 interface

The RS-232 interface cable should connect the socket marked DATACOMM PORT1 at the rear of the HP-150 with the interface socket of the plotter. PORT 1 should be configured to run at 2400 band with XON/OFF set for transmit only. To perform the changes (if any) required, do the following:

1. on the keyboard press <user system> twice followed by <f8> once to select device configuration options.

2. touch the screen option 'PORT1 config' to select PORT 1 configuration options.

3. the following parameters are relevant:

Parameter	Setting
Baudrate	2400
Recvpace	none
Xmitpace	Xon/Xoff
Stop bits	2

If any of these settings do not agree with the above, move the flashing cursor to the appropriate highlighted setting

ANALYSING AND MAPPING THE SPECIMEN DATASETS PROVIDED IN VOLUME I

Essentially there are four stages involved in this process:- loading the data from each transfer (.TXT) file into the mapping and analysis programmmes; making any necessary corrections to that data (e.g. for variations in flying height) and converting the observations into the desired forms (e.g. photo counts into % cover); analysing the data to get basic statistical descriptions and population estimates; and finally, producing distribution maps.

The following pages are intended to give a step by step guide of the commands needed to analyse and map a selection of variables from these files. Four obey files have been provided with the specimen dataset. To use them, simply turn on the computer, type AM, wait for the -- prompt to appear on the screen, and then type OBEY 'PHOTDATA.OBY', and then watch the screen. When action is needed by you, the double prompt will appear again, and the necessary commands will be displayed above. **You must, however, first make sure that the computer is configured correctly, and that the following files are on your disk:**

AM.EXE **i.e the mapping and analysis programme**
AAAA^SF.txt **i.e the specimen photo transfer file**
BBBB^SF.txt **i.e the specimen observation transfer file**
PHOTDATA.oby, OBSDATA.oby, SPECIMEN.OBY, SPECIMEN.MAP, and SPECIMEN.STR
The plotter should also be turned on, with paper positioned in it.

These obey files are reproduced overleaf. Note that any line beginning with a dollar sign ($) is a comment, or a description of the function of the succeeding commands. **Only those lines without the $ are actual commands.**

These files are then provided again, with the screen output that results from their operation.

Once you have tried this, it is suggested that you try to reproduce the output, by entering the individual commands yourself, line by line, from within the mapping and analysis package. To do this, first exit from the package (if you are in it) by typing STOP. Then type AM, wait for the prompt (--), and start entering the commands. The first one begins READ 'AAAA^SF.txt'

Note that you must wait for the prompt (--) before entering any command, and that it makes no difference whether the commands are in capital letters, or in lower case.

Also, enter only those lines which do not have a $ sign as the first letter.

Briefly, the steps necessary are:
1. Read the data into the mapping and analysis package, using the format of the Summary File shown on pages 49 and 50 of Volume 1.
2. Label the columns
3. Check the column values, to see if, for example percentages add up to 100. Correct if necessary.
4. Read in data from another file if necessary, and label and check that.
5. Read in the second set of figures, label the columns, and correct the values (e.g for variations in altitude)
6. Save the cleaned and corrected set of data.
7. Perform various analyses such as obtaining standard statistics, and estimating population levels.
8. Map the data.

Good luck.

Using the XXXX^SF and YYYY^SF .TXT Files you prepared from Volume I

To use the instructions provided in the Obey Files on you own datasets, you must remember to use the second set of obey files that are provided. These are XXXX.OBY and YYYY.OBY, and are exactly the same as PHOTDATA.OBY and OBSDATA.OBY respectively, except that the Filenames used in the first READ instruction have been changed to XXXX^SF.TXT and YYYY^SF.TXT.

The MARRIOTT command

This command is run from the computer operating system, rather than the mapping and analysis package. It provides a third set of standard error estimates that are based on the gridcell and its neighbours as a sampling unit rather than a flight line. A full discussion can be found in Marriott, FHC and Wint, W: (1985) Sampling and Statistics in Low Level Aerial Survey. Report to the International Livestock Centre for Africa by Resource Inventory and Management Ltd. It also provides a short cut method to estimate population levels of the selected parameters within the text file chosen.

It will read any fixed format text file, such as those WRITten by the mapping and analysis programme, or those transferred by the Data Entry Package described in Volume 1 of this Reference Guide. Only one variable can be analysed at a time, but the process is very rapid, so that 25 variables of a 1000 grid dataset can be assessed in between 15 and 20 minutes.

The information needed to be able to use the information within a particular file is: the filename; the format statements to locate the north-south and east-west grid coordinates; and the format statement of the variable to be analysed.

To operate the programme, first check that the relevant file is present on your disk, using the DIR command from within the computer operating system. This file is MARRIOTT.EXE. If it is present, type MARRIOTT, and after a short while the following screen will appear:

```
Give name of survey datafile: yyyy^sf.txt

Give format to read x,y and variable to be estimated: (2f2.0,5x,f4.0)

Values for first record:    1    1    144.9999950
```

Then enter the required file name, remembering to include the filename extension (e.g. .TXT). You will then be asked to give the format for the grid coordinates and the variable to be analysed. This format may either be one you have selected while WRITing the .TXT file from the mapping and analysis package, or perhaps that produced by the transfer file program from within the Data Entry Package. In the following example, the specimen dataset assembled in Part I has been used. Note that this data set is actually far too small to use this program (see below), and is used here for purely illustrative purposes.

If it is the latter, then the format for the Coordinates and the number of animals in the first animal category will be (2f2.0,5x,f4.0). Note that the brackets must be put around the format statement, but that inverted commas are not required.

The programme will then give you the first value in the column to be analysed. If this is wrong, then you have got the format statement wrong, and should try again.

The programme will then produce the following type of table, though without the letters A to L used here to identify each figure:

Survey data read with grid size: 3 by 3
Cells with values: 9

Method	Mean	Variance	Lower	Upper	SE-totpop	N
STAN	*A* 49.00	*B* 4145.50	*C* 6.93	*D* 91.07	*E* 193.16	*F* 9
4 CELL	*G* 49.00	*H* 10904.45	*I* -19.22	*J* 117.22	*K* 313.27	*L* 1

NB - SE of total population = SE / sample percentage

The top line of figures (A to F) gives the mean, variance, lower and upper 95% confidence limits, standard error and sample size of the values in the chosen column. Value E is the standard error of value A times the number of grids in the sample (value F). These values are calculated in the standard way, and use the records from every grid cell in the survey zone.

The bottom line of figures (G to L) again gives the mean grid cell value (G). The remaining figures (H to L) refer only to those cells which contain records in four neighbouring grids (above, below, left and right).

The value (H) represents the variance of the central grids from their four neighbours. Values I to L are respectively the lower and upper 95% confidence limits of the values in the grids with four neighbours; the standard error times the total number of grids with four neighbours; and the total number of grids in the survey zone which have four neighbours.

If you are using a transferred **SUMMARY FILE**, then the values are those observed within the sampled strip. Thus to calculate the total observed population multiply value G by value F. To calculate the total estimated population then divide the total observed population by the sample proportion which your sample strip represents.

To calculate the absolute value for the standard error of the total estimated population, then divide value K by the sample proportion. Finally, to calculate the percentage standard error, divide value K by the total observed population (G x F).

The program then gives you the choice of analysing another variable from the dataset in current use, or at the next stage, of changing the data set being read.

NOTES AND HINTS
To get a print out of the tables produced, simply type CTRL P (if you are using an HP150), before starting the programme operation.

The programme will accept grids with the x-coordinate (generally the east west coordinate) as zero, though not negative. This means that the mapping and analysis package can be used to create a conditional column of coordinates where the EW value is zero for certain strata. These than then form part of a WRITten text file, and so the MARRIOTT programme can then be used to calculate errors and population levels of particular strata. Care should be taken with this, however, as the statistics used are only valid if the chosen strata are spatially concise.

In contrast, the programme will accept neither zero or negative values for the Y-coordinate. Thus, if, for example, the coordinate 0,0 appears in you dataset, then the programme will tell you.

This method of calculating standard errors is particularly suitable for populations which have clumped distributions. However, it should not be used unless the proportion of grids with four neighbours exceeds 30, and is more than half the total number of surveyed grids. If these conditions are not observed, then the assumption that the grids with neighbours are representative of the survey zone as a whole breaks down, and the method becomes invalid.

```
  $ read photographic variables into mapping file
read 'aaaa^sf.txt' c1 to c14 '(3f2.0,81x,6f2.0,9x,5f3.0)'
$ label the columns you have read
$ PCS means Photo Count Sum
label c1 ew c2 ns c3 photoNo c4 PCSCult c5 PCSGrass c6 PCSWood
label c7 PCSFor't c8 PCSbagnd c9 PCSRdset c10 PCSMill
label c11 PCSGnut c12 PCSYam c13 PCSCott c14 PCSRice
$ Check that all PCS land category values add up to a multiple of 40
$ (Total number of counts in land categories is 40 per photo)
let c30=(c4+c5+c6+c7+c8+c9)/c3
describe c30
$ convert PCS land category figures to percentages if values OK
let c31=c3*0.4
let c15=c4/c31
let c16=c5/c31
let c17=c6/c31
let c18=c7/c31
let c19=c8/c31
let c20=c9/c31
$ label newly calculated columns
label c15 %Cult c16 %grass c17 %wood c18 %forest c19 %baregnd
label c20 %rdsset
$ now calculate crop variable scores per photo
$ and put the new values into the original columns
let c10=c10/c3
let c11=c11/c3
let c12=c12/c3
let c13=c13/c3
let c14=c14/c3
$ now re-label columns (using P/P do denote values per/photo)
label c10 millP/P c11 gnutsP/P c12 yamsP/P c13 cottP/P c14 riceP/P
$ check again that all land category % values sum to 100
let c32=c15+c16+c17+c18+c19+c20
describe c32
$ if all these figures are OK then erase working columns C30-c32)
erase c30 to c32
$ now display workspace contents and column labels
status
$ now read observation data into file using 'obsdata.oby'
$ type OBEY 'OBSDATA.OBY' when -- prompt appears on screen
```

LISTING OF OBSDATA.OBY

```
$ read observation data into file
$ if using this obey file after aaaa.oby
$ you dont need to read in grid coordinates, as you have them
$ first read animals and settlements
read 'bbbb^sf.txt' c21 to c30 '(7x,3(f2.0,f4.0),19x,2(f2.0,f4.0))'
$ now read remaining variables into dataset
read 'bbbb^sf.txt' c31 to c34 '(138x,f4.0,5x,2f2.0,7x,f4.0)'
$ label new columns
label c21 cathds c22 catNo c23 shgohds c24 shgoNo c25 camhds
label c26 camNo c27 pastset c28 pastNo c29 arabset c30 arabNo
label c31 altitude c32 admin c33 devreg c34 rainfall
$ correct animals and settlements for variation in altitude
$ assuming constant altitude of 800 ft above ground
let c45=800/c31
let c35=c21*c45
let c36=c22*c45
let c37=c23*c45
let c38=c24*c45
let c39=c25*c45
let c40=c26*c45
let c41=c27*c45
let c42=c28*c45
let c43=c29*c45
let c44=c30*c45
$ label corrected columns
label c35 cathdsC
label c36 catNoC
label c37 shgohdsC
label c38 shgoNoC
label c39 camhdsC
label c40 camNoC
label c41 pastsetC
label c42 pastNoC
label c43 arabsetC
label c44 arabNoC
$ now display workspace contents and column labels
status
$ save assembled dataset containing both photo and observations
save 'specimen.sav'
$ now you can try some analyses
$ if you want to do this use 'specimen.oby'
```

LISTING OF SPECIMEN.OBY

```
$ first describe the basic statistics of each column
desc cl to c45
$ assume that grid dimensions are 5 by 5 kilometers
$ assume that strip width at 800 ft is 800 meters
$ now estimate the population levels of animals and settlements
estim cl c2 c35 width 5 5 0.8
estim cl c2 c36 width 5 5 0.8
estim cl c2 c37 width 5 5 0.8
estim cl c2 c38 width 5 5 0.8
estim cl c2 c39 width 5 5 0.8
estim cl c2 c40 width 5 5 0.8
estim cl c2 c41 width 5 5 0.8
estim cl c2 c42 width 5 5 0.8
estim cl c2 c43 width 5 5 0.8
estim cl c2 c44 width 5 5 0.8
$ now estimate these population levels by development region
$ NB THIS DATA SET IS SMALL SO SOME STRATA WILL HAVE TOO FEW
$ FLIGHT LINES TO CALCULATE POPULATIONS
estim cl c2 c35 devreg width 5 5 0.8
estim cl c2 c36 devreg width 5 5 0.8
estim cl c2 c37 devreg width 5 5 0.8
estim cl c2 c38 devreg width 5 5 0.8
estim cl c2 c39 devreg width 5 5 0.8
estim cl c2 c40 devreg width 5 5 0.8
estim cl c2 c41 devreg width 5 5 0.8
estim cl c2 c42 devreg width 5 5 0.8
estim cl c2 c43 devreg width 5 5 0.8
estim cl c2 c44 devreg width 5 5 0.8
$ now look at frequency distributions of % cover from photographs
class cl5 3
class cl6 3
class cl7 3
class cl8 3
class cl9 3
class c20 3
$ now select grids with more than 50 cattle observed and 25% cult
make c46 if(c36 gt 50 and cl5 gt 25) then (1) else (0)
$ now you can draw some maps using 'specimen.map'
$ first put plotter on line by typing DEVICE
$ then enter appropriate number on request
```

LISTING OF SPECIMEN.MAP

```
$ NB this will only work if plotter has been put on line (DEVICE)
$ set map origin at bottom left corner of paper
origin 0 0 0 0
$set scaling factor (otherwise each grid will be 1mm square!)
scale 10
$ plot grid squares to check that all are present and correct
mark cl c2 size=10 marker=5 colour=1
$ move map origin 100mm to right
origin 0 0 100 0
$ plot a map of cattle numbers
$ NB you must first locate key, and define plotting classes
key 150 10
class c22 3
$ define map symbols
symbol 1 3 5 1 1
symbol 2 6 5 1 1
symbol 3 9 5 1 1
$ draw border around survey site
$ but first check you have written the string file
$ called 'specimen.str' using editor in MS-DOS operating system
$ LINE 1: 1 2 10
$ LINE 2: 0.5 0.5 0.5 3.5 3.5 3.5 3.5 0.5 0.5 0.5
string 'specimen.str'
point cl c2
$ move map origin up 50mm
origin 0 0 100 50
$ map millet values without classes
mark cl c2 marker=1 size=c10 maxsize=9 minsize=0 colour=2
$ put in a title for millet map
$ draw survey border
string 'specimen.str'
text 100 45 'Millet values'
$ move map origin 100 mm to left
origin 0 0 0 50
$ plot grids with more than 50 observed cattle and 25% cultivation
mark cl c2 marker=c46 size=9
$ draw survey boundary
string 'specimen.str'
```

Printout of operating 'photdata.oby'

```
am

-----------------------------------
Aerial Survey: Analysis and mapping
-----------------------------------
--obey photdata.oby
--$ read photographic variables into mapping file
--read 'aaaa^sf.txt' c1 to c14 '(3f2.0,81x,6f2.0,9x,5f3.0)'
--$ label the columns you have read
--$ PCS means Photo Count Sum
--label c1 ew c2 ns c3 photoNo c4 PCSCult c5 PCSGrass c6 PCSWood
--label c7 PCSFor't c8 PCSbagnd c9 PCSRdset c10 PCSMill
--label c11 PCSGnut c12 PCSYam c13 PCSCott c14 PCSRice
--$ Check that all PCS land category values add up to a multiple of 40
--$ (Total number of counts in land categories is 40 per photo)
--let c30=(c4+c5+c6+c7+c8+c9)/c3
--describe c30

COLUMN      MEAN     STAN-DEV     MINIMUM     MAXIMUM      COUNT
-----------------------------------------------------------------
  30       40.00       0.00        40.00       40.00          9
--$ convert PCS land category figures to percentages if values OK
--let c31=c3*0.4
--let c15=c4/c31
--let c16=c5/c31
--let c17=c6/c31
--let c18=c7/c31
--let c19=c8/c31
--let c20=c9/c31
--$ label newly calculated columns
--label c15 %Cult c16 %grass c17 %wood c18 %forest c19 %baregnd
--label c20 %rdsset
--$ now calculate crop variable scores per photo
--$ and put the new values into the original columns
--let c10=c10/c3
--let c11=c11/c3
--let c12=c12/c3
--let c13=c13/c3
--let c14=c14/c3
--$ now re-label columns (using P/P do denote values per/photo)
--label c10 millP/P c11 gnutsP/P c12 yamsP/P c13 cottP/P c14 riceP/P
--$ check again that all land category % values sum to 100
--let c32=c15+c16+c17+c18+c19+c20
--describe c32

COLUMN      MEAN     STAN-DEV     MINIMUM     MAXIMUM      COUNT
-----------------------------------------------------------------
  32       100.00      0.00       100.00      100.00          9
--$ if all these figures are OK then erase working columns C30-c32)
--erase c30 to c32
--$ now display workspace contents and column labels
--status

** COLUMN    LENGTH      LABEL
** C1           9        EW
** C2           9        NS
** C3           9        PHOTONO
** C4           9        PCSCULT
** C5           9        PCSGRASS
** C6           9        PCSWOOD
** C7           9        PCSFOR'T
** C8           9        PCSBAGND
** C9           9        PCSRDSET
** C10          9        MILLP/P
** C11          9        GNUTSP/P
** C12          9        YAMSP/P
** C13          9        COTTP/P
** C14          9        RICEP/P
** C15          9        %CULT
** C16          9        %GRASS
** C17          9        %WOOD
** C18          9        %FOREST
** C19          9        %BAREGND
** C20          9        %RDSSET
--$ now read observation data into file using 'obsdata.oby'
--$ type OBEY 'OBSDATA.OBY' when -- prompt appears on screen
--
```

Printout of operating 'obsdata.oby'

```
obey obsdata.oby
—$ read observation data into file
—$ if using this obey file after aaaa.oby
—$ you dont need to read in grid coordinates, as you have them
—$ first read animals and settlements
—read 'bbbb^sf.txt' c21 to c30 '(7x,3(f2.0,f4.0),19x,2(f2.0,f4.0))'
—$ now read remaining variables into dataset
—read 'bbbb^sf.txt' c31 to c34 '(138x,f4.0,5x,2f2.0,7x,f4.0)'
—$ label new columns
—label c21 cathds c22 catNo c23 shgohds c24 shgoNo c25 camhds
—label c26 camNo c27 pastset c28 pastNo c29 arabset c30 arabNo
—label c31 altitude c32 admin c33 devreg c34 rainfall
—$ correct animals and settlements for variation in altitude
—$ assuming constant altitude of 800 ft above ground
—let c45=800/c31
—let c35=c21*c45
—let c36=c22*c45
—let c37=c23*c45
—let c38=c24*c45
—let c39=c25*c45
—let c40=c26*c45
—let c41=c27*c45
—let c42=c28*c45
—let c43=c29*c45
—let c44=c30*c45
—$ label corrected columns
—label c35 cathdsC
—label c36 catNoC
—label c37 shgohdsC
—label c38 shgoNoC
—label c39 camhdsC
—label c40 camNoC
—label c41 pastsetC
—label c42 pastNoC
—label c43 arabsetC
—label c44 arabNoC
—$ now display workspace contents and column labels
—status

** COLUMN    LENGTH    LABEL
** C1          9       EW
** C2          9       NS
** C3          9       PHOTONO
** C4          9       PCSCULT
** C5          9       PCSGRASS
** C6          9       PCSWOOD
** C7          9       PCSFOR'T
** C8          9       PCSBAGND
** C9          9       PCSRDSET
** C10         9       MILLP/P
** C11         9       GNUTSP/P
** C12         9       YAMSP/P
** C13         9       COTTP/P
** C14         9       RICEP/P
** C15         9       %CULT
** C16         9       %GRASS
** C17         9       %WOOD
** C18         9       %FOREST
** C19         9       %BAREGND
** C20         9       %RDSSET
** C21         9       CATHDS
** C22         9       CATNO
** C23         9       SHGOHDS
** C24         9       SHGONO
** C25         9       CAMHDS
** C26         9       CAMNO
** C27         9       PASTSET
** C28         9       PASTNO
** C29         9       ARABSET
** C30         9       ARABNO
** C31         9       ALTITUDE
** C32         9       ADMIN
** C33         9       DEVREG
** C34         9       RAINFALL
** C35         9       CATHDSC
** C36         9       CATNOC
** C37         9       SHGOHDSC
** C38         9       SHGONOC
** C39         9       CAMHDSC
** C40         9       CAMNOC
** C41         9       PASTSETC
** C42         9       PASTNOC
** C43         9       ARABSETC
** C44         9       ARABNOC
** C45         9
—$ save assembled dataset containing both photo and observations
—save 'specimen.sav'
—$ now you can try some analyses
—$ if you want to do this use 'specimen.oby'
—
```

Printout of operating 'specimen.oby'

```
obey specimen.oby
—$ first describe the basic statistics of each column
—desc cl to c45
```

COLUMN	MEAN	STAN-DEV	MINIMUM	MAXIMUM	COUNT
1	2.00	0.67	1.00	3.00	9
2	2.00	0.67	1.00	3.00	9
3	3.67	1.11	2.00	5.00	9
4	52.11	34.74	0.00	98.00	9
5	25.11	18.10	0.00	46.00	9
6	33.67	22.44	0.00	67.00	9
7	27.78	34.81	0.00	80.00	9
8	3.00	2.00	0.00	6.00	9
9	5.00	4.00	0.00	11.00	9
10	4.13	3.75	0.00	10.60	9
11	2.86	2.62	0.00	7.60	9
12	1.56	2.77	0.00	14.00	9
13	0.46	0.61	0.00	1.60	9
14	0.93	1.45	0.00	6.60	9
15	28.83	19.22	0.00	50.00	9
16	13.92	9.73	0.00	28.75	9
17	18.96	12.64	0.00	36.25	9
18	33.89	44.07	0.00	100.00	9
19	1.67	1.11	0.00	3.13	9
20	2.74	2.07	0.00	6.25	9
21	1.78	1.19	0.00	4.00	9
22	65.11	59.46	0.00	158.00	9
23	1.00	1.11	0.00	4.00	9
24	56.67	62.96	0.00	221.00	9
25	1.00	0.67	0.00	3.00	9
26	11.11	10.17	0.00	48.00	9
27	1.67	1.11	0.00	3.00	9
28	14.44	10.40	0.00	33.00	9
29	1.44	0.96	0.00	3.00	9
30	52.78	38.86	0.00	112.00	9
31	811.11	23.46	750.00	850.00	9
32	1.33	0.44	1.00	2.00	9
33	1.33	0.44	1.00	2.00	9
34	1500.00	0.00	1500.00	1500.00	9
35	1.73	1.15	0.00	4.00	9
36	63.11	57.47	0.00	152.29	9
37	0.97	1.09	0.00	3.86	9
38	55.00	61.11	0.00	213.01	9
39	0.96	0.64	0.00	2.82	9
40	10.72	9.77	0.00	45.18	9
41	1.63	1.08	0.00	3.04	9
42	14.15	10.28	0.00	33.00	9
43	1.40	0.94	0.00	2.89	9
44	51.29	37.67	0.00	107.95	9
45	0.99	0.03	0.94	1.07	9

```
—$ assume that grid dimensions are 5 by 5 kilometers
—$ assume that strip width at 800 ft is 800 meters
—$ now estimate the population levels of animals and settlements
—estim cl c2 c35 width 5 5 0.8
```

Variable 35 CATHDSC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9 C	3	97.2	44.6	45.9	0.43	44.6	45.9
None	225	9 R	3	97.2	23.7	24.4	0.43	23.0	23.6

```
—estim cl c2 c36 width 5 5 0.8
```

Variable 36 CATNOC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9 C	3	3549.8	1660.0	46.8	15.78	1660.0	46.8
None	225	9 R	3	3549.8	582.9	16.4	15.78	602.6	17.0

```
—estim cl c2 c37 width 5 5 0.8
```

Variable 37 SHGOHDSC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9 C	3	54.7	29.2	53.4	0.24	29.2	53.4
None	225	9 R	3	54.7	25.7	47.0	0.24	25.4	46.4

```
—estim cl c2 c38 width 5 5 0.8
```

Variable 38 SHGONOC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	3093.6	1577.9	51.0	13.75	1577.9	51.0
None	225	9	R	3	3093.6	1341.9	43.4	13.75	1331.7	43.0

—estim cl c2 c39 width 5 5 0.8

Variable 39 CAMHDSC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	54.2	28.2	52.1	0.24	28.2	52.1
None	225	9	R	3	54.2	16.1	29.7	0.24	15.5	28.6

—estim cl c2 c40 width 5 5 0.8

Variable 40 CAMNOC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	603.2	288.6	47.8	2.68	288.6	47.8
None	225	9	R	3	603.2	187.6	31.1	2.68	181.0	30.0

—estim cl c2 c41 width 5 5 0.8

Variable 41 PASTSETC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	91.5	42.4	46.4	0.41	42.4	46.4
None	225	9	R	3	91.5	12.0	13.1	0.41	13.7	14.9

—estim cl c2 c42 width 5 5 0.8

Variable 42 PASTNOC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	796.1	384.1	48.3	3.54	384.1	48.3
None	225	9	R	3	796.1	119.5	15.0	3.54	136.4	17.1

—estim cl c2 c43 width 5 5 0.8

Variable 43 ARABSETC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	78.9	36.7	46.4	0.35	36.7	46.4
None	225	9	R	3	78.9	2.3	2.9	0.35	2.2	2.8

—estim cl c2 c44 width 5 5 0.8

Variable 44 ARABNOC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
None	225	9	C	3	2885.0	1339.0	46.4	12.82	1339.0	46.4
None	225	9	R	3	2885.0	272.3	9.4	12.82	252.9	8.8

—$ now estimate these population levels by development region
—$ NB THIS DATA SET IS SMALL SO SOME STRATA WILL HAVE TOO FEW
—$ FLIGHT LINES TO CALCULATE POPULATIONS
—estim cl c2 c35 devreg width 5 5 0.8

Variable 35 CATHDSC

Stratum	area	grid cells		flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6	C	3	72.5	33.2	45.9	0.48	33.2	45.9
33/ 1	150	6	R	2	72.5	22.2	30.6	0.48	23.5	32.5
33/ 2	75	3	C	3	24.7	11.3	45.9	0.33	11.3	45.9

** TOO FEW FLIGHT LINES
—estim cl c2 c36 devreg width 5 5 0.8

Printout of operating 'specimen.oby' (cont)

Variable 36 CATNOC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	2420.8	1145.1	47.3	16.14	1145.1	47.3
33/ 1	150	6 R	2	2420.8	651.0	26.9	16.14	690.5	28.5
33/ 2	75	3 C	3	1129.0	803.5	71.2	15.05	803.5	71.2

** TOO FEW FLIGHT LINES
—estim cl c2 c37 devreg width 5 5 0.8

Variable 37 SHGOHDSC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	30.0	19.9	66.5	0.20	19.9	66.5
33/ 1	150	6 R	2	30.0	25.9	86.2	0.20	27.4	91.5
33/ 2	75	3 C	3	24.7	11.3	45.9	0.33	11.3	45.9

** TOO FEW FLIGHT LINES
—estim cl c2 c38 devreg width 5 5 0.8

Variable 38 SHGONOC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	1749.0	1080.9	61.8	11.66	1080.9	61.8
33/ 1	150	6 R	2	1749.0	1371.7	78.4	11.66	1454.9	83.2
33/ 2	75	3 C	3	1344.7	617.1	45.9	17.93	617.1	45.9

** TOO FEW FLIGHT LINES
—estim cl c2 c39 devreg width 5 5 0.8

Variable 39 CAMHDSC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	41.8	23.6	56.3	0.28	23.6	56.3
33/ 1	150	6 R	2	41.8	14.5	34.7	0.28	15.4	36.8
33/ 2	75	3 C	3	12.4	5.7	45.9	0.16	5.7	45.9

** TOO FEW FLIGHT LINES
—estim cl c2 c40 devreg width 5 5 0.8

Variable 40 CAMNOC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	464.6	274.9	59.2	3.10	274.9	59.2
33/ 1	150	6 R	2	464.6	173.5	37.3	3.10	184.0	39.6
33/ 2	75	3 C	3	138.6	110.9	80.0	1.85	110.9	80.0

** TOO FEW FLIGHT LINES
—estim cl c2 c41 devreg width 5 5 0.8

Variable 41 PASTSETC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	54.5	25.6	47.0	0.36	25.6	47.0
33/ 1	150	6 R	2	54.5	11.2	20.6	0.36	11.9	21.8
33/ 2	75	3 C	3	37.1	17.0	45.9	0.49	17.0	45.9

** TOO FEW FLIGHT LINES
—estim cl c2 c42 devreg width 5 5 0.8

Variable 42 PASTNOC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	480.4	237.7	49.5	3.20	237.7	49.5
33/ 1	150	6 R	2	480.4	127.9	26.6	3.20	135.6	28.2
33/ 2	75	3 C	3	315.8	147.9	46.8	4.21	147.9	46.8

** TOO FEW FLIGHT LINES
—estim cl c2 c43 devreg width 5 5 0.8

Variable 43 ARABSETC

Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	54.2	25.5	46.9	0.36	25.5	46.9
33/ 1	150	6 R	2	54.2	0.0	0.0	0.36	0.0	0.0
33/ 2	75	3 C	3	24.7	11.3	45.9	0.33	11.3	45.9

** TOO FEW FLIGHT LINES
—estim cl c2 c44 devreg width 5 5 0.8

Variable 44 ARABNOC

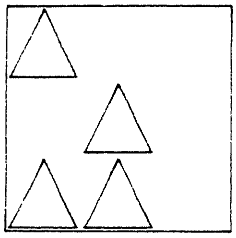
Stratum	area	grid cells	flight lines	population	standard error	%SE	density	alternate std.error	%SE
33/ 1	150	6 C	3	1771.1	820.7	46.3	11.81	820.7	46.3
33/ 1	150	6 R	2	1771.1	154.6	8.7	11.81	164.0	9.3
33/ 2	75	3 C	3	1113.9	518.4	46.5	14.85	518.4	46.5

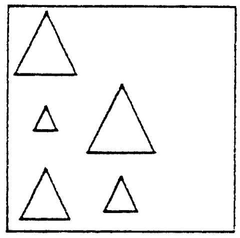
```
** TOO FEW FLIGHT LINES
—$ now look at frequency distributions of % cover from photographs
—class cl5 3

CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
 1        0.000       16.667        3      ***
 2       16.667       33.333        0
 3       33.333       50.000        6      ******
EACH * DENOTES    1 OBSERVATION(s)
—class cl6 3

CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
 1        0.000        9.583        3      ***
 2        9.583       19.167        2      **
 3       19.167       28.750        4      ****
EACH * DENOTES    1 OBSERVATION(s)
—class cl7 3

CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
 1        0.000       12.083        3      ***
 2       12.083       24.167        2      **
 3       24.167       36.250        4      ****
EACH * DENOTES    1 OBSERVATION(s)
—class cl8 3

CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
 1        0.000       33.333        6      ******
 2       33.333       66.667        0
 3       66.667      100.000        3      ***
EACH * DENOTES    1 OBSERVATION(s)
—class cl9 3

CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
 1        0.000        1.042        3      ***
 2        1.042        2.083        2      **
 3        2.083        3.125        4      ****
EACH * DENOTES    1 OBSERVATION(s)
—class c20 3

CLASS  LOWER-BOUND  UPPER-BOUND  FREQUENCY
 1        0.000        2.083        4      ****
 2        2.083        4.167        2      **
 3        4.167        6.250        3      ***
EACH * DENOTES    1 OBSERVATION(s)
—$ now select grids with more than 50 cattle observed and 25% cult
—make c46 if(c36 gt 50 and cl5 gt 25) then (1) else (0)
—$ now you can draw some maps using 'specimen.map'
—$ first put plotter on line by typing DEVICE
—$ then enter appropriate number on request
—
```

LISTING OF SPECIMEN.MAP

```
$ NB this will only work if plotter has been put on line (DEVICE)
$ set map origin at bottom left corner of paper
origin 0 0 0 0
$set scaling factor (otherwise each grid will be 1mm square!)
scale 10
$ plot grid squares to check that all are present and correct
mark c1 c2 size=10 marker=5 colour=1
$ move map origin 100mm to right
origin 0 0 100 0
$ plot a map of cattle numbers
$ NB you must first locate key, and define plotting classes
key 150 10
class c22 3
$ define map symbols
symbol 1 3 5 1 1
symbol 2 6 5 1 1
symbol 3 9 5 1 1
$ draw border around survey site
$ but first check you have written the string file
$ called 'specimen.str' using editor in MS-DOS operating system
$ LINE 1: 1 2 10
$ LINE 2: 0.5 0.5 0.5 3.5 3.5 3.5 3.5 0.5 0.5 0.5
string 'specimen.str'
point c1 c2
$ move map origin up 50mm
origin 0 0 100 50
$ map millet values without classes
mark c1 c2 marker=1 size=c10 maxsize=9 minsize=0 colour=2
$ put in a title for millet map
$ draw survey border
string 'specimen.str'
text 100 45 'Millet values'
$ move map origin 100 mm to left
origin 0 0 0 50
$ plot grids with more than 50 observed cattle and 25% cultivation
mark c1 c2 marker=c46 size=9
$ draw survey boundary
string 'specimen.str'
```

THE MAPS DRAWN USING 'SPECIMEN.MAP'





Millet values