Standard Software Applications for Statistics in Developing Countries

One possible Solution
to a case study

created during the seminar
on micro—computer applications
in
Freetown, Sierra Leone
03/05/1388 - 27/05/1988

Authors: Klaus Roder
Herbert Girkes

Li - Standard software for statisticians in DC

Table of Contents

CDG- Standard software for statisticians in DC

1. Introduction

This booklet is the resume of a 3 weeks computer course in Freetown/Sierra Leone.

The author of this report has elaborated the course content and together with Herbert Girkes consultant at the EGA in Addis Abbaba documented the possible solution the case study.

The views expressed hereafter are personal opinions of the authors and should not be attributed to any organization .

The Munich Center for Advanced Training has been engaged in training statisticians of developing countries for the past 15 years. Microcomputer applications have increasingly become an important section of these training activities.

It was felt however that classical computer training including programming and applications for statistics in analysis etc. deals generally only with one aspect of computer use in statistical offices. The main bulk of routine work is data entry for surveys and presentation of survey results.

Although microcomputers, as they are defined today are far from replacing mainframes in all areas and never will, there are numerous cases where these machines readily available in many offices can substitute mainframe solutions, bypass bottlenecks due to limited computer access and enhance presentation and user-friendliness of statistical applications.

A major difficulty of computer training is to jump the first fences of computer access, these hurdles have increasingly diminished their size by modern software design. Though some basic knowledge of computer handling is still required the tedious process of starting with a programming language can be avoided for the non-expert user, especially the statistician and still achieve valuable work on the microcomputers.

Thus workshop did not try to discourage the programmers who participated in this seminar, on the contrary, but it certainly tried to encourage the non-programming approach. It remains obvious that elaborated applications especially of programmable data-bases requires some computer background, but it seems unnecessary to scare the users with too much computer lingo, uncountable business applications with microcomputers should be the example to follow.

These reasons and the fact that in most offices not only the industry standard microcomputers are available but also some software packages which have become a de-facto standard, like the ones used in this workshop, have let to the design of this workshop. There certainly are valuable options for the statistician like statistics packages including or incorporating tasks handled by the standard software, either purely commercial ones like SPSS/PC or SAS or tailored to measure ones, offered by research, training and foreign cooperation agencies and institutions.

The advantage using standard software application will be that the user is not restricted to one specific package, e.g. can replace one spreadsheet by another and can be sure to be able to rely on dependable and rather well documented software. Apart from these advantage the modular approach is not limited to surveys only but can lead to numerous other applications depending on the experiences the user might acquire using these software packages.

The actual survey treated in principal was carried out in Sierra Leone in 1984 to 1986 and processed on minicomputers. The purpose of the workshop was by no means to criticize the the approach of the CSO of Sierra Leone. There the avalability of microcomputers did not permit in any case the processing, it is rather a general alternative suggestion to enhance further applications.

Some 30 questionnaires were entered out of 194, only a section of each questionnaire was treated in order to avoid the rather tedious data entry procedure to be repeated too many times.

The choice for the three software packages, dBase III+, Lotus 1-2-3 Vers.2 and .Microsoft Chart Vers.3 used is thus rather arbitrary, however a preference was given to package which are not copy protected. This was not true yet for Lotus 1-2-3 but eventually this was scheduled to change in close future.

Klaus Roeder

2 The workshop program

The Seminar:" Use of microcomputers and application of standard-software in a statistical service " was held from 9.5.88 to 28.5.88 in Freetown/Sierra Leone in the Institute of Public Administration and Management (IPAM) .

The Munich Center was responsible for the organization of the seminar.Klaus Rfider, former Computer lecturer of the Munich Center, was responsible for content, lecturing, preparation and implementation in Freetown. Herbert Girkes, statistics and computer consultant at the Economic Commission for Africa was also responsible for lecturing and organization.

The seminar was meant for professional statisticians from government services.Academic background and at least two years working experience as well as computer knowledge and access to a microcomputer were other preconditions.

The participants: 18 participants from Gambia(2), Ghana(4), Liberia(4), Nigeria(4) and Sierra Leone(4). All participants work in the statistical services of their respective countries.

2.1 Timetable of the seminar

Day 1:
Administration
Target of the seminar
Fundamentals of microcomputer handling
Fundamentals of the operating system MS-DOS
Presentation of the software to be dealt with:
- dBase 111 +
- Lotus 1-2-3
- MS-Chart

Introducing dBase III+:
Use of the ASSISTANT Create a dBase-File Retrieve and update of data

Day 2:
Editing and appending data
Indexing of files Data entry screens
* On-your-own exercise:
Create a dBase-file
Create a data entry screen
Write a data entry program with the ASSISTANT

Day 3:
Command programs
Programming fundamentals: Loops (dc.enddo)
Conditional branches (if..endif)
* On-your-own exercise:
Programming of an error-retrieval program

Day 4:
Menus with the help of the Application Generator

Modular Program design
Subprograms
Converting dBase-files to ASCII-files

Day 5: *On-your-own exercise:
Programming of a data entry program in dBase with menu technique and the following subprograms:
- Data entry with error control
- Data correction,deleting and appending of data
- Data output on screen and printer
- Data transfer to an ASCII file

Day 6: Accompanied on-youi—own exercises (Saturday)

Day 7: Fundamentals of the spreadsheet program Lotus 1-2-3

Cells and cell contents
Use of functions and formulas
Create a table and modify the table
Printout

Day 8: Accompanied on-your-own exercises (Holiday)

Day 9: Ranges in Lotus 1-2-3
Defining names for ranges
Copying and moving of ranges
Absolute and relative cell-references
Graphics in Lotus 1-2-3
Transfer of dBase files to Lotus 1-2-3
* On-your-own exercise:
Create a table with the data of the dBase data entry-program of day 5

Day 10: The elements of the graphics program package MS-Chart List-Entry-Chart screens

Day 11: Editing of data in Chart
Format editing of
Graphics
Transfer of Lotus 1-2-3 files to MS-Chart
Printing of Graphics

* Afternoon: Visit of the Central Statistics Office in Freetown

Day 12: Accompanied on-your-own exercises (Saturday)

Day 13-16:

Case study:

Processing of data of a survey in Sierra Leone:

"Census of Large Manufacturing Establishments 84-84 & 85-86"

using microcomputers and standard-software, 38 completed questionnaires were treated. The number of questions to evaluate was limited to 55.

The following tasks had to be solved by the participants in groups of two or three and as much unassisted as possible:

-Programming of a data-entry-program in dBase in menus-technique
-Transfer of data to Lotus 1-2-3 and aggregate in two tables -Transfer of data to MS-Chart, creating three
presentation graphics

Day 17: Printing and documentation of programs
of the case study by the participants.
Presentation of model solution, valuation
Closing ceremony

2.2. The tasks of the workshop case study

Case study:

Authors: Klaus Roeder
Herbert Girkes

17.5.1988

What to do :

I. Write a data entry program for the
"Sierra. Leone - Establishment survey"

The following fields should be contained in the data file:
1. Questionnaire number (1—100)
2. ISIC Code (4 digits)
3. Questions to serial numbers 5.1 to 5.9 (6 questions each)
4. Questions to serial numbers 6.1 to 6.3 (2 questions each)

II. The data entry program should be menu-like containing:
1. data entry with checks
2. data editing
3. data listing
4. data transfer to ASCII file
5. exit from entry program

III. The following checks should be applied during data entry:
1. The numbers in line 5.7 must contain the sum of lines 5.3 to 5.6 for each column
2. The numbers in line 5.9 must contain the sum of lines 5.1 to 5.6 for each column
3. The numbers in line 6.3 must contain the sum of lines 6.1 to 6.2 for each column

IV. Transfer the data file after data entry to Lotus 1—2-3 and present data in three tables:

1. Table 1: Number of establishments, numbers of persons engaged and labor costs by 3 employment size groups (10-19 employees, 20-100, above 100 and all classes) for the two periods 84-85 and 85-86. (see example Table 1)

2. Table 2: Number of establishments and number of persons engaged by enterprise groups (31,32,33,34,35,36,38+39,9512 and 9513) and all industries for the two periods 84-85 and 85-86. This table should have 10 rows and 4 columns.(see example Table 2)

Table 3: Number of persons engaged, number and percentage and skilled percentage by the enterprise groups (31,32,33,34,35,36, 38+39,9512 and 9513) and all industries for the two periods 84-85 and 85-86. This table should have 10 rows and 10 columns.(see example Table 3)

Use Chart to present at least three graphics of the three tables to emphasize : non-Sierra Leoneans in workers in number and

1. For Table 1: Distribution of number of establishments, employees and labor costs in the two periods of survey by employment size groups.

2. For Table 2: Distribution of number of establishments and employees in the two periods of survey by employment groups.

3. For Table 3: Distribution of different groups of employees in the two periods of survey by employment groups (see example charts)

4. Try to use your own program, table and chart design it

2.3 Specimen of questionnaire

ANNEX

GOVERNMENT OF SIERRA LEONE

Census of Large Manufacturing Establishments
1984-85 & 1985-86

(Authorised under Statistics Act, 1963)

Please read carefully the Manual of Instructions (enclosed) before completing this form. For any clarification, please contact Mr. E. Webber, Senior Statistician or Mr. C.G. Richards, Assistant Statistician, Central Statistics Office (CSO), Tower Hill, Freetown. Telephone : 24267.

BLOCK 1 - Identification particulars

1.1. Name of establishment... The I.N.T. Chemical Corporation

1.2 Full address including post/mail box, if any. The INT Chemical Corporation 21 Wellington Street P.O. Box 317, Freetown Sierra Leone

1.3 Is this establishment part of a multi-establishment enterprise?

Yes [ ] 1 No [✓] 2

1.4 If the answer to 1.3 is 'Yes', write the name and address of headquarters

1.5 Accounting Year..... 1982

BLOCK 2 - Type of Ownership

1 [ ] Foreign 2 [ ] National-Government 3 [✓] National-private

4 [ ] Others (specify)..... Code (Leave blank) [ ]

BLOCK 3 - Type of legal organisation

1 [ ] Individual proprietor s 2 [✓] Partnership 3 [ ] Registered Compan

4 [ ] Cooperative Society 5 [ ] Others (specify) .....
Code (Leave blank) [ ]

BLOCK 4 - Kind of Activity

Manufacturing of Chemicals

Code (Leave blank) 3 5 1 0

BLOCK 5 - Average annual and quarterly employment

S. No.	Category	1984-85 Male	1984-85 Female	1984-85 Total	1985-86 Male	1985-86 Female	1985-86 Total
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
5.1	Working proprietors & partners	50	30	80	55	45	100
5.2	Unpaid workers	40	20	60	35	20	55
5.3	Paid professional and managerial staff	10	10	20	20	25	45
5.4	Paid skilled operatives	15	10	25	20	15	35
5.5	Other paid operatives	16	14	30	10	10	20
5.6	Other paid employees	12	13	25	10	12	22
5.7	Total paid employees	53	47	100	60	62	122
5.8	Total non-Sierra Leone employees	5	5	10	10	15	25
5.9	Total persons engaged	196	144	340	210	189	399
5.10	Persons engaged - I quarter						
5.11	Persons engaged - II quarter						
5.12	Persons engaged - III quarter						
5.13	Persons engaged - IV quarter						

BLOCK 6 - Compensation to employees during the accounting year

S. No.	Item	Amount paid (Le.) 1984-85	Amount paid (Le.) 1985-86
(1)	(2)	(3)	(4)
6.1	Wages and salaries of employees	708,000	750,000
6.2	Employers' contribution to social security pension funds etc.	220,000	215,000
6.3	Total	928,000	965,000

BLOCK 7 - Quantity and cost of materials and supplies received/consumed

S. No.	Description	Unit of Quantity	Code	1984-85 Quantity	1984-85 Value (Le)	1985-86 Quantity	1985-86 Value (Le)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7.1							
7.2							
7.3							
7.4							
7.5							
7.6	Others	x	x	x		x	

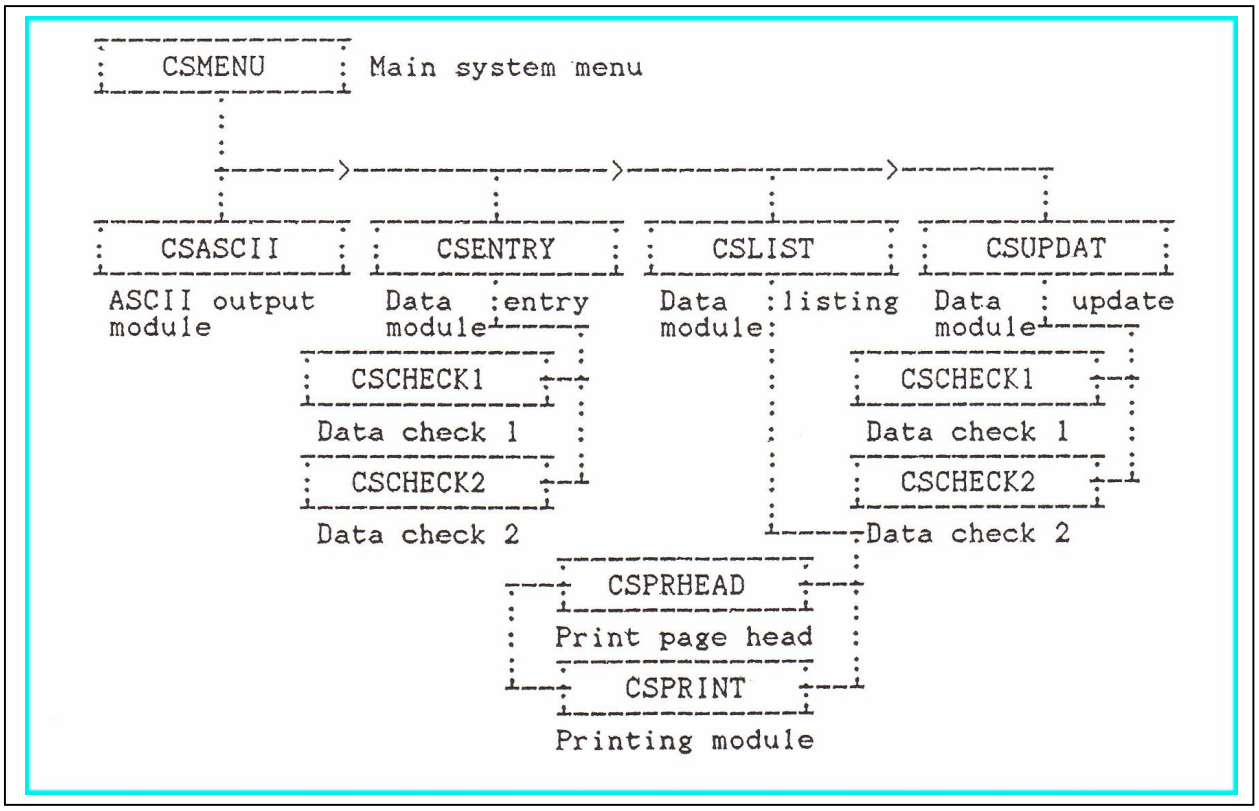
3. The Data Entry System

3.1 Basic layout

Any data entry system has to contain the following modules to be useful:

- one data entry module,
- one module which allows to update or correct data,
- one module which allows to print out the data, and
- one module which allows to prepare the stored data in such a way that other programs (for instance a data analysis program like SPSS) can read the ente red data for further processing.

In the light of these basic requirements the case study data entry system has been designed (see graph 1). To demonstrate the power for on-line data checking while entering data, special modules have been created to check the entered data. As indicated during the lessons, dBASE III plus provides some range checking facilities for numeric data and data of the DATE type. In some cases this might not be enough for the "real" world. Due to this the modules CSCHECK1.PRG and CSCHECK2.PRG should be taken into consideration as a guide on how to create even more complex checking programs. It is obvious that not all the possibilities to establish checking programs could be demonstrated. Among the possibilities not mentioned are for instance databases where valid values can be looked up or the facility to cross-check field values where logical connections are existing. All this is possible with only decent programming knowledge, raph 1:
Basic layout of the case study Data Entry System



3.2 Main system menu

The main menu will be the first screen to be exposed to the user of the system. Four choices are offered to the user as defined by the basic system layout. It should be remembered that this main menu was created using the program APPSGEN.PRG provided with the dBASE III plus system.

The use of the applications generator APPSGEN.PRG facilitates the programming of such type of menus and the resulting .PRG file can easily be amended by using a wordprocessor or the dBASE III plus editor (REMARK: It must be taken into account that this in—build editor allows only for 6000 bytes of program. If bigger programs; have to be written, either several small programs, not exceeding 6000 bytes, have to be written, calling one after the other, or one external wordprocessing system has to be used to write the program).

The following is the description of the program APPLEX.PRG in detail, according to the printout given in the Annex.

The lines 1 to 31 are comments.They describe the program name, author, creation date, date of last revision, and describe in the notes the purpose of the program. Lines 33 to 37 define the dBASE III plus environment for the execution of the program (all these settings and their meaning can be recalled via the HELP facility inside dBASE III plus).

Lines 38 to 40 define the program path, the data path and database file to be used. The two lines 41 and 42 are containing messages which are stored to memory variables. The purpose of this is that dBASE III plus does not allow ordinary setting of screen attributes as possible by other programs using the ANSI.SYS driver (it circumvents the loaded screen drivers and implements a generic one). If one wants to highlight text by displaying this text in reverse video, the only way is to "GET" this text on the screen because dBASE III plus inverts fields to get automatically. dBASE I'll plus offers one command (the "CLEAR GETS" command) which enables the user to clear out all "GET's" set since the last "READ". This means that the system after a "CLEAR GETS" no longer remembers that values have to be read and the desired effect of inverting just a message has taken place. Now new "GET's" can be set. Line 43 fills a memory variable with the data path name plus the database file name. This memory variable "iMainFile" will be used to call in the desired database file. This technique is often used in cases where the data or program path might be changed. It allows to change the paths at a single location in the program system (usually the starting module) and to make sure that all over the system the appropriate definition is used.

Line46 establishes one endless loop because the expression (here ".T.") can never become false. This type of loops, otherwise not wanted, is mostly established to permit the display of the same menu over and over again until the selection of the exit option has been chosen.

Lines 43 to 117 are mainly created using the APPSGEN.PRG. The main difference is the elimination of the "RANGE"-clause in the "GET"-statement concerning the variable "iSelectnum". This was done to demonstrate one other feature of the "CASE"-structure, the "OTHERWISE"-statement. This statement will be executed if all other "CASE"-statements were not fulfilled. As can be seen, only for the values 1 to 4 other "CASE"-statements are existing. Any other value will execute the "OTHERWISE"-statement.

In addition to this change the programs CSENTRY.PRG, CSOPDAT.PRG, CSLIST.PRG, and CSASCII.PRG are not called directly using a command like "DO csnetry". This is due to the fact that all programs are store into a separate path called "C:cDBASEcFREETOWNcPROGSc". Because of this, the path information has to be added to the filename. Otherwise dBASE III plus would seek for the files in the path in use. For further explanation see the above discussion on paths.

3.3 Data entry screen

Major parts of the data entry screen have been created using the "CREATE FORMAT" command. Using this command from the systems dot prompt or by the ASSIST, provides the user with the so called "Blackboard" to design the layout of a screen (this was demonstrated and exercised during the seminar). Again some statements were added to create a complete, independent module.

Lines 1 to 22 contain again comments as discussed in 3.2. The "STORE" statements are used for the program path definition for the programs "CSCHECK1" and "CSCHECK2", and the expression used in the "DO WHILE"-loop. This time no indefinite loop was used but a loop which ends if the value of the memory variable "iMore" is no longer "Y", In order to allow the user to enter also the value "y" the value of "iMore" is first transferred to upper case letters before the condition check takes place- Otherwise the user would be forced to type the exact letter "Y" as the answer of the question asked at the end of the loop.

In order to avoid that unwanted data is entered into the database file all input is first stored into memory variables. These variables are preset zeroes inside the loop (lines 31 to 41) to avoid the carry on of previous values. The major difference between the standard "CREATE FORMAT" output and the program, as shown in the annex, is that all "GETS'"s have been placed together (the normal order is "SAY" followed immediately by the associated "GET"). This was done to enter a loop which will be in effect until the user declares that the entered data is correct. Lines 63 to 135 form this loop. The memory variable "iCorrect" is first set to ".F." (for false) and this loop is executed until the user answers with "Y" or "v" to the question "Data correct (Y/N) ?". This technique, where only the "GET"'s are used inside the loop, decreases the time delay caused by displaying the complete screen layout at any loop iteration.

In line 137 the program CSCHECK1.PRG is executed. As stated earlier, all programs are stored into a specific subdirectory (in the example into C:cDBASEcFREETOWNcPROGSc). The memory variable "iSubl" contains the path to this subdirectory plus the program name. As this should be "DO"ne, the use of a macro variable is necessary ("DO SiSubl"). After checking the data of the first data entry screen (for a description of the program CSCHECK1.PRG see below), a second data entry screen is displayed and the user has to enter the remaining data. The same definitions as described above are applying to the second data entry screen. After the second data check (for a description of the program CSCHECK2.PRG see below) a new, blank record is created in the data file ("APPEND BLANK"). Finally the record field contents of this new, empty record are "REPLACE'M by the values kept in the respective memory variables, and the user is asked whether he or she wants to enter more data. Once the user answers with anything else but "Y" or "y", the program returns control to the main menu.

In line 133 the clearing of the question "Data correct (Y/N)?" is executed using a simple "SAY". This "SAY" just prints a sequence of blank to the screen, erasing the line displayed before. It must be mentioned that the usual way to clear a line (and the rest of the screen) using "a row,col CLEAR" is not applicable if the "SET STATUS ON" is in effect. The execution of such a statement might have unwanted effects: the complete screen might be erased. This is why throughout the whole system the described technique erasing message lines is used. It seems to be more valuable to keep the status line on the screen than easing up the cleaning of lines.

3.4 Update module

The update module CSUPDAT.PRG is used to update or delete records in the existing data base file. Three options are available to the user: updating a single record, deleting a single record and using the dBASE III plus system's facility "BROWSE" to do both (and more: for instance adding new records to the file). This availability of choices demands for a menu which has been created again using the applications generator APPSGEN.PRG (the whole "DO WHILE .T."-loop, this time again an endless loop). Line 29 contains the setting of a memory variable which will be later used to determine whether the database file has to be "PACK"ed. Lines 24 to 52 follow the already discussed sequence. But instead of executing after the "CASE=" statement a "DO" statement (meaning the execution of one ".PRG"-file), the actual statements are executed inside the CSUPDAT.PRG.

Line 57 to 60 execute the question to the user on the record number to be updated. In line 62 the statement "ON ERROR STORE •F. TO iOk" appears. This is a special command in dBASE III plus to trap errors which might occur during execution. In this case it is used to check whether the entered record number is existing in the current data base file: if the command "GOTO iNum" is executed and the record is not existing, normally an error message will appear, indication "Record out of range", and the execution of the program is terminated. But because the "ON ERROR" statement is in effect, this message is bypassed and the statement "STORE .F. TO iOk" will be executed. Otherwise a programmed message is displayed that the record number was wrong and the CSUPDAT.PRG selection number reappears.

The statements which are executed if the variable "iOk" is true are almost the same as the data entry screen in CSENTRY.PRG. Actually the lines 122 to 218 and 225 to 289 were copied from the CSENTRY.PRG lines 42 to 137 and 139 to 204 respectively. The difference is, that this time the field-values of the actual record are used for filling the memory variables starting with "i". This means that the values of the chosen record are displayed and can be updated. The programs CSCHECK1.PRG and CSCHECK2.PRG are called within the updating process because obviously the user can update the record in a manner that the new values are not acceptable.

The second selection (deleting a specified record) uses again most of the above described statements. This time the display of the first data screen of the actual record is performed to allow the user to cross—check the selected record. This is necessary because the program provides one automatic pack feature which permanently removes deleted records.

If the user acknowledges that the chosen record is the record to be deleted, a memory variable "iToPack" is set to true (initially this variable is set to false).

The third selection enters the user into the full—screen dBASE III plus "BROWSE" facility. Inside "BROWSE" the user can delete records, update records or even add new records. "BROWSE" is terminated by pressing Crtl-END.

Due to the fact, that the user can delete records while in "BROWSE" it is necessary to check whether one record has been marked for deletion. This is done in lines 428 to 437. After detection of a deleted record the user can decide that this record must not be removed permanently. This is acknowledged in the value of "iToPack". If the user indicates that he or she wishes definitely to delete marked records the "PACK"ing is done.

3.5 Listing module

The listing module CSLIST.PRG starts with a simple menu to decide whether the data file will be listed on screen or on printer. In case the listing is done on the screen ("CASE iSelectnum=l"), the screen is cleared, the database file is set to the first record and the "DISPLAY" of each record in the file is performed. To reduce distortion on the screen the status line is switched off before this action and after the status line reappears. If listing to printer was chosen the user will be asked whether the printer is switched on. This is necessary because otherwise the program would be terminated with the "Printer not ready" message.

Due to the fact that the data in the records need more than just a single line on the printer (with a maximum of 132 characters) it was necessary to write to small programs to print the data in a readable format (programs CSPRHEAD.PRG and CSPRINT.PRG). These programs are described later. As can be seen in the program the page number (iPageNo) is first set to zero and the current date is set to the memory variable iPgDate. This two variables will be used to print the output heading on top of each page. After all records are printed the device is reset to screen, so that all further "a.." commands will be executed again on the screen.

3.6 Module to create an ASCII-file

As discussed during the lectures it is necessary to prepare dBASE III plus data for further use. dBASE III plus employs a special scheme to store .DBF files which is not recognized by other programs like CENTS 4, SPSS and the like. Some programs like lotus 1-2-3 and MS-Chart have their built in translation facilities for some of the most widely used programs, like dBASE III plus. This frees the user from any transformation inside dBASE III plus.But still it might be a useful feature to be able to create one ASCII file recognized by virtually any micro-computer program.

The program CSASCII.PRG allows in its present form to create either a file in delimited or undelimited format.Delimited files contain the contents of the database file in the following form:

- fields of type "CHARACTER" are enclosed in double quotes
- fields of type "NUMERIC" are transferred as they are

- fields of type "DATE" are transferred in the format yyyymmdd (year.month,day)

- fields of type "LOGICAL" are transferred as characters T or Y (for true or yes) and F or N (for false or no)

- fields of type "MEMO" are not transferred.

All fields are separated by commas.

Files of type SDF will take the following format:

- fields of type "CHARACTER" are printed in full length according to the defined field length without quotes (i.e. a character field defined 25 charachters long will occupy in the resulting ASCII-file 25 characters and the actual character string is filled with blanks if less than 25 characters)

- fields of type "NUMERIC" are transferred as they are

- fields of type "DATE" are transferred in the format yyyymmdd (year,month,day)

- fields of type "LOGICAL" are transferred as charac ters T (for true or yes) and F (for false or no). This means regardless of the value being T or Y the character T is written and for F or N the character F is output.

- fields of type "MEMO" are not transferred.

Lines 29 to 40 establish the menu screen for selecting the appropriate choice. In case the user wants to create an ASCII-file he or she has to enter the path and the name of the output file. It should be recalled that the standard extension of the output file (whether delimited or of type SDF) is ".TXT".

In linme 67 to 71 a check is done whether the path has been entered with a trailing "9" which is necessary to separate the path from the actual file name. The lines 73 to 77 add automatically the extension ".TXT" to the output file name if the extension ".TXT" was not found in the file name. The statement "AT(".TXT",iOutFile)" returns the position of the ".TXT" extension in the string stored in the variable "iOutFile" if existing. If no such character string is found the "AT"-function returns a value of zero. Lines 78 to 95 check the existance of the file. The function "FILE(filename)" returns a value of true if the file is existing. In this case the user will be asked whether the file should be overwritten. If the user answers with yes the file is overwritten otherwise a new path and filename can be entered. The same system applies to lines 120 to 150 for the selection of delimited file output.

The actual copy will take place at the and of the program in **lines** 170 to 179.

3.7 Data check programs

The programs CSCHECK1.PRG and CSCHECK2.PRG are called from two other programs: CSENTRY.PRG, the data entry program, and CSUPDAT.PRG, the data updating program. Beside the usual checks as demonstrated during the lectures and as mainly in CSCHECK1.PRG and CSCHECK2.PRG in lines 41 to 60 of program CSCHECK1.PRG a slightly more complex checking is done.

It was required that the ISIC code used for the case study should start with the figures 31, 32, 33 35, 36, 38, 39 or 95. To test this the numeric memory variable ilSIC has to be divided be 100 to isolate the first to digits. The "INK)" function returns the integer equivalent of the division and the "STR(>" function transfers this value into string type. This makes it easy to check with a single statement whether one of the acceptable ISIC codes is stored in ilSIC or not. The "AT()" function checks whether the first string (in this case the string STR(INT(iISIC/100),2)) is part of the second string (in this case "3132333536383995". This second string contains all the valid codes as defined above.

This technique has been chosen in order to avoid several checking: the acceptable codes are not in an unbroken sequence (for instance 34 is missing). Otherwise several "IF" statements would have to be executed.

3.8 Programs used with printout

Two programs are used for the printout of the database contents. The first program CSPRHEAD.PRG just adds 1 to the page number, advances the paper to the top of page ("EJECT"), prints the header and stores a value to the memory variable iLine. This variable will be used to position the print head in the correct line for printing the contents of the records.

The second program CSPRINT.PRG actually prints the contents of the records. It was calculated in a way that a maximum of 132 characters are printed on a line. Most printers installed are not of the wide carriage type, this means normally only 80 characters can be printed per line. But almost all matrix printers provide for the so-called condensed format, which actually allows for 17 characters per inch printing, enough for 132 characters. The program CSLIST.PRG contains the automatic switch to this printer mode in line 98 and sets back to normal printing mode in line 108 (ATTENTION: the used codes 15 and 18 might differ for your own printer. These codes are used by EPSON printers and work-alike).

In CSPRINT.PRG is checked whether a given record is the first printed on this page. If so the heading "Record //" is printed otherwise omitted. This was just done to show that it is possible to change with a few statements the printed headings while printing.

4. Data presentation

Quite obvious the interface to the user of statistics is the printout, the table, the graph, the intelligible computer screen, in brief the information required by the user.

The spreadsheet are not specially designed to contain statistical information but the ease of manipulation, modification of column width, headers, calculations and simple statistics make them a choice for tabulation and facilitate changes after first layout, which can be modified without much effort for changing requirements.

Although many spreadsheets notably 1—2—3 provides graphics facilities these are rudimentary compared to more flexible special graphics packages. Graphics should be incorporated as much as possible in presentation of data, although some experience is needed to reduce information in graphs to the essential.

Report writing would be, in any case, a complete training module to envisage, using recent possibilities of desk-top-publishing together with the mentioned packages. It can only be stressed again and again that without proper presentation of data, this means reliable, intelligible , designed to the needs of users and flexible enough to be adapted to changing demands, all efforts of statisticians will be eventually be lost and purposeless.

4.1 Spreadsheet solution for table design

For a certain time Lotus 1-2-3 has been a de-facto standard for spreadsheets. Other products may exceed its capabilities but the purpose of this workshop was to rely on a widespread product.

Lotus 1-2-3 like almost all other product import at least ASCII files if not special dBASE files. In order to remain as general and to allow the user to use part of the training modules with other products ASCII files were transferred from dBASE III+. These can be aggregated prior to transfer or afterwards in Lotus.

The availability of graphics character can be improved with the automatic replace facilities of a word processor to avoid the rather clumsy column and line separators. Lotus does not support semigraphic characters, while most word processors do.

The printouts can be improved by using other available software products like SIDEWAYS which allows prints with enlarged with and direct handling of Lotus format.

Another advantage of using Lotus 1-2-3 is the capability of many graphics packages to access without detour via ASCII files the spreadsheet format.

4.2 **Graphics with standard software**

The choice of a graphics package is less obvious than in the two other sections, MS—Chart is widespread but by no means the most brilliant. FREELANCE by Lotus allows easier access to the 1-2-3 spreadsheets, HARVARD GRAPHICS allows comfortable use of self- and pre-designed symbols, combination with freehand graphics like PAINTBRUSH by" Microsoft could further enhance the presentation.

Nevertheless MS-Chart is easy to learn, rather flexible and provides much better facilities than the 1-2-3 graphic. Furthermore it can access directly 1-2-3 spreadsheets and part of it although the use of this utility is rather clumsy. In addition Chart allows some elementary statistics analysis as does 1-2-3.

5. Conclusion

The example shown in the annex shows the possibility on how to use a standard micro-computer database program to create a data entry system. It has been shown during the seminar that all major functions may be processed using the built in facilities (namely ASSIST). The creation of a simple data, entry system has been demonstrated, showing that even non-programmers are able to create useful dBASE III plus program system with only a few additional statements.

Additional statements and some "tricks" provide a system which might be of general use to statistical offices. Replacing the data entry and update screen by the screen or screens of an actual survey, changing the path names for program and data files as well as the used database file name might do most of the job to create a data entry system. The CSCHECK1.PRG and CSCHECK2.PRG modules have to be replaced by a checking program reflecting the actual range or logical checks.

Even though dBASE III plus programs have to be tested as well before any final use, it seems to be less time consuming to test within a given orogram environment like dBASE III plus but to write the COBOL, BASIC or RPG II equivalent on a mini-computer or mainframe computer.

The handling of Spreadsheets and Graphics packages is easier still for the beginner, although one notes the limitations to be restricted to the user interface. Despite some shortcomings the advantages of these packages against programmed approaches are obvious, the flexibility of the modular approach seems to be another advantage.

Overall this is not the definite answer to the shortcomings and bottlenecks in data processing which are frequently encountered in Developing Countries. The authors would be happy though if this report could stir some initiative in the mentioned direction and complement, facilitate or enhance the existing tasks of the statistician. Whenever it is suitable (and if the appropriate number of micro-computers are available) this type of application should be given a try.