Education and Training in the SAS System at the Central Statistical Office in Harare

Sten Bäcklund

SCB

R & D Report
Statistics Sweden
Research - Methods - Development
1988:13

INLEDNING

TILL

R & D report : research, methods, development / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1988-2004. – Nr. 1988:1-2004:2.
Häri ingår Abstracts : sammanfattningar av metodrapporter från SCB med egen numrering.

Föregångare:

Metodinformation : preliminär rapport från Statistiska centralbyrån. – Stockholm : Statistiska centralbyrån. – 1984-1986. – Nr 1984:1-1986:8.

U/ADB / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1986-1987. – Nr E24-E26

R & D report : research, methods, development, U/STM / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1987. – Nr 29-41.

Efterföljare:

Research and development : methodology reports from Statistics Sweden. – Stockholm : Statistiska centralbyrån. – 2006-. – Nr 2006:1-.

R & D Report 1988:13. Education and training in the SAS system at the Central Statistical Office in Harare / Sten Bäcklund.
Digitaliserad av Statistiska centralbyrån (SCB) 2016.

urn:nbn:se:scb-1988-X101OP8813

Education and Training in the SAS System at the Central Statistical Office in Harare

Sten Bäcklund

SCB

R & D Report
Statistics Sweden
Research - Methods - Development
1988:13

Published September, 1988
Printer STATISTICS SWEDEN, Dept of Research and Development, EDP Systems Unit
Publisher Staffan Wahlström
Questions answered by Bo Sundgren

ISSN 0283-8680

Printed in Sweden
Garnisonstryckeriet, Stockholm 1988

During April 29 - May 13 a course in Statistical Analysis System (SAS) was given to some of the staff at the Central Statistical Office (CSO) in Harare. This course is to be seen as one of the important steps towards an integration of EDP and statistical theory at CSO. The new MVAX-2 has so far SAS installed as the only statistical software. Some other software is for the moment being used for data entry and validation, but it should in the future be replaced with designated SAS modules.

SAS is built up from modules. Each module covers specific fields of data processing. The modules consists mainly of **pre-written procedures** which make data processing easy, also for people that have a limited experience in computer programming.

There is though, a programming language within SAS, which is used to **create datasets** of information. These datasets are later to be used as **input** in the procedures. An important set of procedures are the reporting and tabulation procedures. Thus, the course was organized in two theory parts:
- how to create SAS datasets from registred data
- how to report information from the SAS datasets using pre-written procedures

This is all done with the **SAS Base module**, which is one of two modules currently installed on the MVAX2. The other module is **SAS Graph**.

Theory lessons were always immediately followed up by hands-on training at the VAX terminals. Fortunately, all of the participants were used to the keyboard and the operating system (VMS), which made all much easier. The only thing outside SAS that had to be learnt was the SAS Display Manager, which serves as a tool for on-line program developing and testing.

As case study for the course, the Intercensal Demographic Survey 1987/88 was used. Some of the forms from the ICDS was registred resulting in around 450 data records on individuals, 100 households, 30 deceased and 90 women for fertility studies. As basic system design, the mission report by Mrs B Lagerlöf was taken (See reference list).

Theory and practice were integrated during the hours from 9 a.m. to 12.15 p.m. Afternoons and early mornings were often used by the participants for practice. All material for the course was created in Harare during the mission using Lotus Freelance Plus for overhead slides on a HP pen plotter, WordPerfect for text writing and SAS for program testing.

General comments and suggestions for the future

There was a 100 % participation during the course which is remarkable. All of the students were good and some excellent. If the moral and skill stays that high after the course, there should, without any discussion, be a strong EDP unit after some time of further practice. This in turn will enable the CSO to process and report information gained from future surveys and censuses without delays.

SAS is a powerful tool for statistical data processing. At the same time it sometimes seems to make life difficult for the VAX system. SAS is memory-demanding and when run in an interactive environment using the display manager there can be unwanted break-downs of the VMS system. This can be avoided if every SAS user shows a certain degree of responsibility and disciplin versus VMS. Therefore, rules should be stated and followed.

There will be a need of, if possible, a continuous follow-up during the first year in order to keep the SAS know-how high and also for further penetration of the SAS system. There is still a lot more to be achieved.

It is strongly recommended that the module **SAS/FSP** is bought and installed. This module will make it possible to use SAS the **whole way** from data entry to tabulation. SAS/FSP is an interactive data base handler for SAS datasets which can be used for data editing. It can also be used for designing **customized** data entry screens. A specially made screen control language **(SCL)** can be applied for crossvalidation of entry fields and other kinds of validation.

APPENDICES

Appendix 1: SAS Course: Theory

Appendix 2: SAS Course: Training

Appendix 3: SAS Display Manager

Appendix 4: PROC TABULATE

Appendix 5: Sample SAS programs for the ICDS

Appendix 6: Lecture minutes made from overhead slides

Appendix 7: Participation list

Appendix 8: Certificate

Appendix 9: Course evaluation form

REFERENCES

SAS User's Guide: Basics (Version 5 edition)

SAS System under VMS

SAS Guide to TABULATE processing

B Lagerlöf: Development of System Design for National Household Surveys
ZIMSTAT 1988:2, 1988-01-28

Appendix 1

SAS COURSE

THEORY

CSO, Harare

May, 1988

Sten Backlund, Statistics Sweden

SAS DATASET

A SAS dataset is a table or matrix of information, where

- rows are observations
- columns are variables

A SAS dataset also contains information (metadata) on itself like

- on what file it is based
- when it was created
- how it was created (the source code)

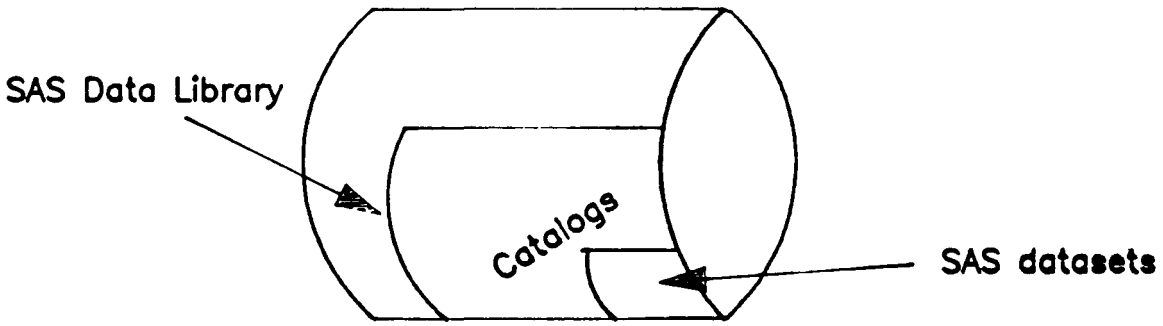
etc

SAS DATA LIBRARY

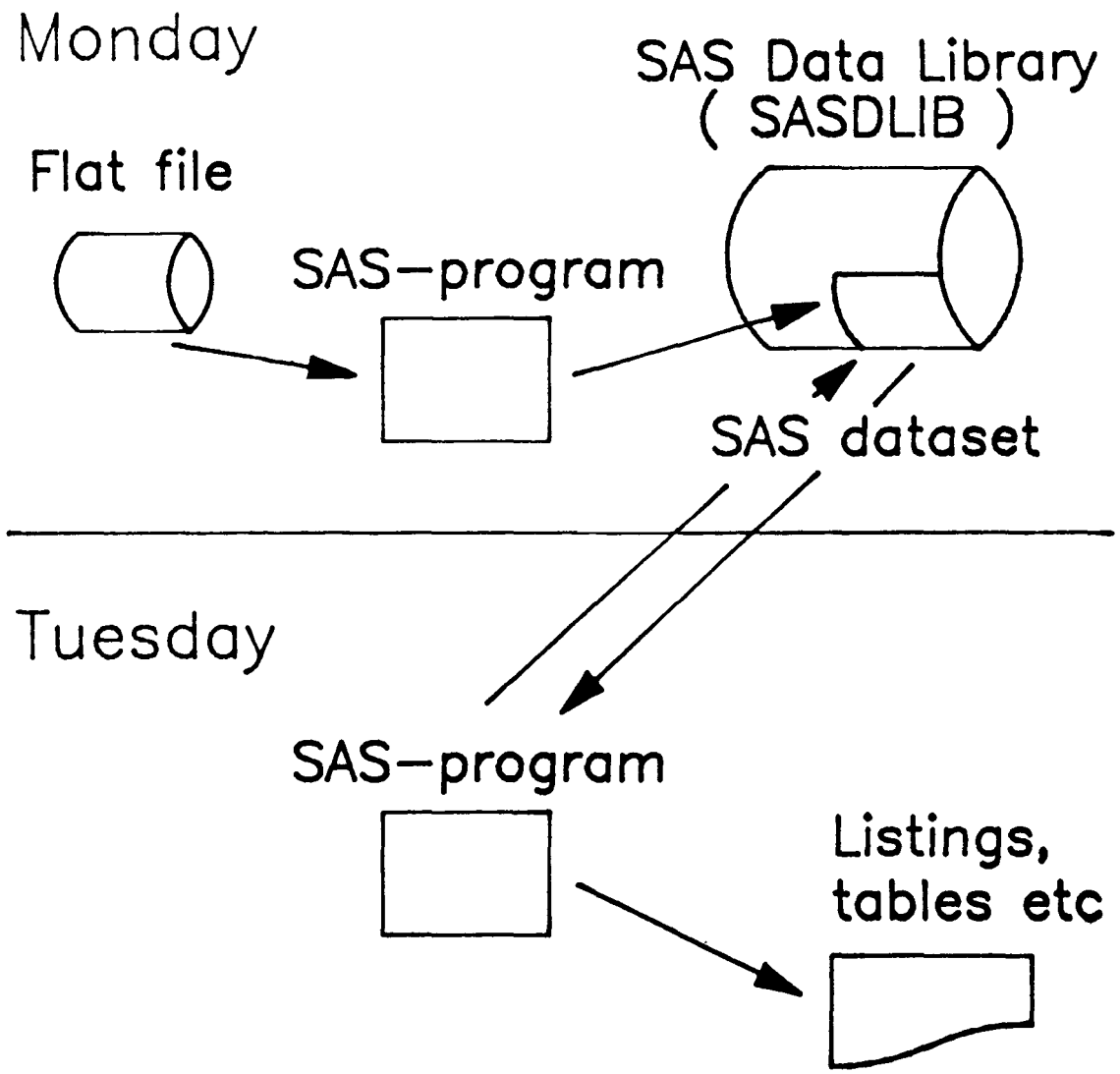
A SAS Data Library (SASDLIB) can contain information of different kind.
The SASDLIB is administrated by SAS and only SAS can read and write this information.

The same kind of information is stored in a catalog.

A SAS dataset will be stored in a catalog of the type DATA



HOW TO USE A SASDLIB



NAMING RULES

A variable name or a SAS dataset name consists of 1 to 8 letters or digits.

Underscore _ is allowed.

Itmust start with a letter or underscore.

Examples:

Right:	VAR1	_id	SEX
Wrong:	2A	NATIONALITY	CASH@

THE SAS LANGUAGE

The SAS language elements are keywords and statements.
Statements are separated by semicolons ;
A statement has to start with a keyword or a variable name.

Examples:

```
IF X=9 THEN Z=12;
NAME='STANLEY JOHNSON';
Y=X**2;
```

Character strings are always enclosed in single apostrophes.

SAS DATASET NAMES

SAS starts building a new SAS dataset when it reads the DATA statement.

```
DATA d1;
```

This means that a dataset with the name d1 will be created and stored in a SASDLIB named WORK. The DATA statement actually reads

```
DATA WORK.d1;
```

When giving a two–level name like i.e.

```
DATA census.d1;
```

the SAS dataset d1 will be stored permanently in the SAS Data Library census .

TWO IMPORTANT KEYWORDS: DATA AND PROC

The keywords DATA and PROC have a special meaning in SAS.

The DATA keyword tells SAS to start building one or more SAS datasets. All statements following DATA up to the next DATA or PROC (or RUN) keyword will together form a data step.

The PROC keyword tells SAS to use one of the pre-written procedures to process just one SAS dataset. All statements following PROC up to the next DATA or PROC (or RUN) keyword will together form a procedure step.

THE FIRST SAS–PROGRAM

```
DATA censlib.house;
   INFILE  censdata;
   INPUT    @1  area  $CHAR3.
            @4  division  $CHAR2.
                subdiv  $CHAR2.
            @10  hhnr  $CHAR4.
            @8  ednr  $CHAR2.
            @14  intd  2.  intm  2.
            ;
   LM == area !! division !! subdiv !! ednr !! hhnr;
RUN;
```

DATA	tells SAS to start build a permanent SAS dataset named house in the SAS Data Library censlib
INFILE	opens the file censdata
INPUT	specifies where and how the dataset variables should be read from the infile
@	is the pointer
$CHARw.	is an informat (instruction)
RUN	ends the data step

THE INPUT STATEMENT

When SAS sees the INPUT keyword SAS will act as follows:

- One record from the infile is fetched and put into a buffer
- SAS reads values into variables given in the statement
- SAS uses the pointer @ to find the starting position for the value
- The values are read according to the informat (instruction) given

Note: Every INPUT reads a new record.

VARIABLE ATTRIBUTES

A variable in SAS has 6 attributes

– name	the name of the variable
– type	character or numeric (C or N)
– label	replaces the variable name in printouts
– length	the number of bytes used for storing the variable's values in memory
– informat	how the input data are to be read from i.i. a flat file
– format	how the variable's values are to be written

Attributes given in a data step will belong to the metadata of the SAS dataset and the variable

LENGTH

SAS automatically assigns 8 bytes for storing a numeric variable.

Character variables are assigned

- the number of bytes given in the w parameter when the $CHARw. informat is used in an INPUT statement
- 8 bytes if the character variable is ≤ 8 characters and to the left in an assignment statement
- the number of characters of the variable value in an assignment statement if it is more than 8 characters

Examples:	VARIABLE		LENGTH
INPUT	sex	1. ;	8
INPUT	area	$CHAR2. ;	2
	f_name = 'JOHN';		8
	l_name = 'Andersson';		9

LABEL

A label will replace the variable name in tables and reports

A label is given in a LABEL statement

Example:

```
DATA d1 ;
   LENGTH pid 2 intdate $ 4;
   INFILE censdata;
   INPUT @14 intd $CHAR2. intm $CHAR2.
         @24 pid 2.;
  intdate = intd !! intm;
  LABEL intdate = 'Interview date'
        pid = 'Household member no';
RUN;
PROC CONTENTS;
RUN;
PROC PRINT LABEL;
RUN;
```

FORMAT

A format will replace the variable value in tables and reports

A format is given in a FORMAT statement

Suppose there is an instruction already defined which translates the numeric values 1,2 for the variable sex into the labels 'Male','Female'

This instruction is then a format
A format needs a name
Let us call this format SEXF (or any valid SAS name)

Example:

```
DATA d1;
  LENGTH sex 2;
  INFILE censdata;
  INPUT @28 sex 1.;
  LABEL sex = 'Sex';
  FORMAT sex sexf.;
RUN;
PROC CONTENTS; RUN;
PROC PRINT; RUN;
```

TITLES, FOOTNOTES AND OPTIONS

There can be up to and including 10 title lines and 10 footnote lines specified

```
TITLE; /* suppresses the default line */
TITLE2 'This is title line 2';
FOOTNOTE 'This is first footnote line';
```

Titles and footnotes are in effect until next title or footnote statement with the same number

System options are used to supervise the SAS program

```
OPTIONS NODATE LS=120;
OPTIONS PAGESIZE=56 NONUMBER;
```

TITLE, FOOTNOTE and OPTIONS statements can be put anywhere in a SAS program.

They do not belong to data– or procedure steps!

SET

The SET statement is used when creating a new SAS dataset from an existing SAS dataset

```
DATA  new;
   SET old;
   totage = yeard - yearb;
RUN;
```

OUTPUT

When SAS reads the OUTPUT statement, SAS will add an observation to the specified dataset. All current values for the obser-vation are taken.

```
DATA  new;
   SET old;
   OUTPUT;
   X=2;
RUN;
```

SUBSETTING IF

The subsetting IF statement is used to select certain observations from a SAS dataset

```
DATA women;
  SET person;
  IF sex = '2';
RUN;
```

IF – THEN – ELSE

Syntax: IF condition THEN statement ;
ELSE statement ;

```
DATA men
     women;
  SET person;
  IF sex = '2' THEN OUTPUT women;
  ELSE OUTPUT men;
RUN;
```

DELETE

Has inversely the same effect as the subsetting IF statement

```
DATA women;
  SET person;
  IF sex = '1'
  THEN DELETE;
RUN;
```

MISSING VALUES

Before executing the data step, the values of all defined variables are set to missing, meaning

. for numeric variables (period)
' ' for character variables (blank)

The variables will get their values during the execution of the data step. When all statements in the datastep are executed, the observation is added to the data set.
Variables that by then have not received a value will retain the missing value.

MISSING VALUES (cont.)

There are two exceptions where the variable values are not set to missing namely

- the sum statement
- the RETAIN statement

Example:

The infile MISSDATA contains 9 records, each with just one variable, X
The values are: 1,2,3, ,5,6, ,8,9

```
DATA test;
  INFILE missdata;
  INPUT x  1.;
  RETAIN y;
  y = x;
  z + x;
RUN;
```

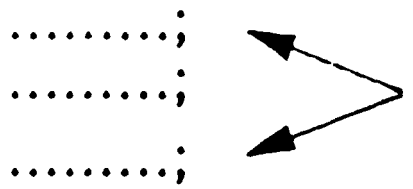
SAS DATA SET OPTIONS

KEEP = list of variables
DROP = list of variables
FIRSTOBS = n start at obs no. n
OBS = n stop at obs no. n
RENAME = (oldname=newname ....)
READ = password read protecting
WRITE = password write protecting

```
DATA headhh (KEEP=sex age citship  READ=secret)
     children (KEEP=sex age relth school WRITE=[illegible]);
  SET censlib.person (OBS=500);
  IF relth = '1' THEN OUTPUT headhh;
  IF age<11 THEN OUTPUT children;
RUN;
PROC PRINT DATA=children(FIRSTOBS=20);
RUN;
PROC PRINT DATA=headhh(READ=secret DROP=age);
RUN;
```

DO - END

Syntax: DO;



END;

GOTO

Syntax: GOTO label;

Example:

```
DATA ..... ;
     ......  ;
     IF age < 12 THEN GOTO stop;
     ......  ;
     ......  ;
 stop: RETURN;
     ......  ;
 RUN;
```

DO - TO - BY

Syntax:

```
DO indexvariable=start TO
   stop  BY  increment;
```

Example:

```
DATA d1;
  DO i=1 TO 20 BY 2;
    j=i**2; OUTPUT;
  END;
RUN;
```

DO - UNTIL

Syntax:

```
DO UNTIL (expression);
     .... ;
     .... ;
END;
```

PROCEDURES

All procedures in the SAS system follow
a specific pattern in the manuals

Abstract	what the procedure is used for
Introduction	brief description
Specifications	PROC–statement no. 1 ; PROC–statement no. 2 ; .....; PROC–statement no. n ;
Details, Examples	examples, in which the procedure is applied
References	theoretical background

SORTING A SAS DATASET

Syntax:

```
PROC SORT options ;
   BY option variable option variable ...;
```

Options in the PROC statement :

```
DATA =   SAS dataset
OUT =    SAS dataset
NODUPLICATES
```

Option in the BY statement

```
DESCENDING
```

Example:

```
PROC SORT DATA=censlib.person
                OUT=persort;
   BY hid DESCENDING age;
RUN;
```

LISTING OF A SAS DATA SET

```
PROC PRINT options  ;
  VAR       variables ;
  ID        variables ;
  BY        variables ;
  PAGEBY    byvariable ;
  SUM       variables ;
  SUMBY     byvariable;
```

Options in the PROC statement can be:

DATA =	SAS dataset
N	the number of obs will be printed
UNIFORM	all pages have the same layout
DOUBLE	double-spaced output
LABEL	variable labels are used as headings
SPLIT	gives a split-character for labels

Example:

```
PROC PRINT DATA=censlib.person(OBS=10) LABEL;
  VAR sex usmem age;
  ID pid;
  LABEL  sex='Sex'
         usmem = 'Usual member'
         age='Age in years'
         pid='Household member no.'
         ;
  TITLE 'Listing of censlib.person';
RUN;
```

FREQUENCY TABLES

Syntax:

```
PROC FREQ  options ;
  TABLES  requests / options ;
  WEIGHT  variable ;
  BY  variables ;
```

Options in the PROC statement :

```
DATA =  SAS dataset
FORMCHAR(1,2,7) =  'string'
```

Options in the TABLES statement :

MISSING	if missing values are included
LIST	list output requested
OUT =	SAS dataset
NOPERCENT	no cell percentages
NOCOL	no column percentages
NOROW	no row percentages
NOPRINT	no printout wanted

Table requests like

```
A B   A*B   A*(B C)   (A B)*(C D)
```

Example:

```
PROC FREQ DATA=censlib.person;
  TABLES sex relth sex*relth / norow;
  BY area;
RUN;
```

USER DEFINED FORMATS

Syntax:

```
PROC FORMAT options ;
  VALUE    name
           range = label
           .... ;
  PICTURE  name
           range = label
           .... ;
```

Options in the PROC statement :

LIBRARY = libref

Range can be:

- a list of values
- a range of values
- lists of ranges and values

Labels should always be enclosed in single quotes

USER DEFINED FORMATS (cont.)

Example:

```
PROC FORMAT LIBRARY = censlib;
  VALUE $usmemf
        '1'   = 'Present'
        '2'   = 'Absent'
        OTHER = 'Miscoded'
        ;
  VALUE agef
        0 - 6 =' 0- 6'
        7 -14 =' 7-14'
        15-54 ='15-54'
       55-HIGH='55-  '
      OTHER = 'Miscoded'
        ;
RUN;

LIBSEARCH censlib;
PROC FREQ DATA=censlib.person;
  TABLES usmem*age / NOCOL NOROW NOPERCENT;
  FORMAT usmem $usmemf. age agef. ;
RUN;
```

BY

The BY statement gives access to two important variables :

FIRST. byvariable
LAST. byvariable

They are to be considered as Boolean variables and can either be 0 or 1

When the byvariable changes its value
FIRST. byvariable = 1

If the observation is the last one with the current byvariable value, then
LAST.byvariable = 0

Example:

OBS	ID	FIRST.ID	LAST.ID
1	11	1	0
2	11	0	0
3	11	0	1
4	12	1	1
5	13	1	0
6	13	0	1
7	14	1	1
8	15	1	?

MERGE

The MERGE statement is used for joining observations from two or more SAS–datasets into single observations in a new dataset.

Syntax:

```
MERGE SAS dataset [(dsoptions IN=name)]
      SAS dataset [(dsoptions IN=name)]
      ...
      [ END= name)] ;
```

IN = name	creates a variable which will have the value of 1 if the dataset contributed data to the current observation; 0 otherwise
END=name	creates a variable which will have the value of 1 when end–of–file is reached

MERGE is almost exclusively used together with a BY statement
This in turn means that the merging SAS datasets must be sorted on the BY variable

MERGE (cont.)

Suppose there is a SAS dataset containing the variables HID and PID ; the dataset name is D1 :

OBS	HID	PID
1	101	1
2	101	2
3	101	3
4	423	1
5	515	1
6	515	2
7	601	1
8	601	2
9	601	3
10	601	4

OBS	HID	PID	SOH
1	101	1	3
2	101	2	3
3	101	3	3
4	423	1	1
5	515	1	2
6	515	2	2
7	601	1	4
8	601	2	4
9	601	3	4
10	601	4	4

It is, as can be seen, sorted on HID. Now the procedure FREQ can be use to calculate the household size :

```
PROC FREQ DATA=d1;
  TABLES hid / NOPRINT
           OUT = f1 ;
```

The variable COUNT in f1 will contain the hh size

Next thing is to merge :

```
DATA m1(KEEP=HID PID
        RENAME=(COUNT=soh) );
  MERGE d1(IN=ind1)
        f1(IN=inf1);
  BY hid;
  IF ind1=1 & inf1=1;
```

LIBRARY HANDLING

Syntax:

```
PROC DATASETS NOFS LIBRARY= libref;
  DELETE SAS dataset(-s) ;
  SAVE SAS dataset(-s) ;
  CHANGE oldname=newname ..... ;
  MODIFY SAS dataset ;
    FORMAT variable format. variable format. ... ;
    LABEL variable=label variable=label ... ;
    RENAME variable=newname variable=newname ...;
```

NOFS means no-full-screen editing

WRITING FILES

```
FILENAME children 'external name';
DATA _NULL_;
  SET censlib.person;
  FILE children;
  IF years>12 THEN RETURN;
  PUT      hid       $CHAR13.
           stratum   $CHAR1.
           years 8. ;
  RUN;
```

NULL	no dataset to be created
FILE	opens a file for writing
PUT	actually writes to the file

WRITING FILES (cont.)

```
DATA _NULL_;
  SET censlib.person; BY area;
  FILE PRINT HEADER=newpage;
  IF FIRST.area THEN PUT _PAGE_;
  PUT    hid $CHAR13.  /
         @20 pid $2.
         'Writing' ;
  RETURN;
  newpage:
    PUT 'Here starts a new page' //;
  RETURN;
```

FILE PRINT	opens a print file
HEADER	gives a name to a label
PAGE	new page
/	new line

Appendix 2

SAS COURSE

TRAINING

CSO, Harare

May, 1988

Sten Backlund, Statistics Sweden

GPAN 20111

Strictly confidential when completed

Identification

Intercensal Demographic Survey 1987/88

Round One, August 1987

CENTRAL STATISTICAL OFFICE,
P.O. BOX 8063, CAUSEWAY,
HARARE: TEL. 706681.

Starting time: .

Survey number	Round number	Administrative area	Division	Subdivision	E.A. number	Segment number	Sub-sample	Household number	Day of actual interview	Month of actual interview	Record type
1 2	3	4 5 6	7 8	9 10	11 12	13	14	15 16 17 18	19 20	21 22	23
1 2	1										1

Serial No	Usual member of the household, whether present or absent last night, and visitors who stayed here last night Name	Usual member present last night = 1 absent last night = 2 visitor = 3	Relationship to head of household (codes given below)	Sex M = 1 F = 2	Age at last birthday: Months	Age at last birthday: Years	Place of birth (State district or town in Zimbabwe. If born outside Zimbabwe, state country) Name	Place of birth: Office use only	Citizenship (codes given below)	Has this person changed his/her place of residence in the last 12 months (State district or town of person was in Zimbabwe or country if elsewhere) Name	Has this person changed his/her place of residence in the last 12 months: Office use only	Ethnic group: Black = 1 White = 2 Coloured = 3 Asian = 4 Other = 5	Marital status: Never married = 1 Married = 2 Divorced/ separated = 3 Widowed = 4	School attendance: At school = 1 Left school = 2 Never attended = 3	Highest level of academic education completed	Further vocational training (in years)	Persons aged 10 years and above: Main kind of work during the last 12 months: Description of work	Main kind of work: Office use only	If person did not work, reason for not working (codes given below)
24 25		26	27	28	29 30	31 32		33 34 35	36		37 38 39	40	41	42	43 44	45		46 47	48
01																			
02																			
03																			
04																			
05																			
06																			
07																			
08																			
09																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			

Column 27
Head = 1
Spouse = 2
Son/Daughter = 3
Son-in-law/Daughter-in-law = 4
Mother-in-law/Father-in-law = 5
Other relative = 6
Not related = 7

Column 36
Zimbabwe = 1
Mozambique = 2
Malawi = 3
Zambia = 4
Other African = 5
Other non-African = 6

Column 43 & 44
Grade 0 = 00
Grade 1 = 01
Grade 2 = 02
Grade 3 = 03
Grade 4 = 04
Grade 5 = 05
Grade 6 = 06
Grade 7 = 07
Form 1 = 08
Form 2 = 09
Form 3 = 10
Form 4 = 11
Form 5/M Level = 12
A Level = 13

Column 48
Unemployed = 1
Retired = 2
Pensioner = 2
Home duties = 3
Sick = 4
Handicapped = 5
Attending school = 6
Other = 7

Ending time

Enumerator

Checked by

Team leader

Date

Survey number		Round number	Administrative area			Division		Sub-division		E.A. number		Segment number	Sub-sample	Household number				Day of actual interview		Month of actual interview	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	2	1																			

23 Number of deaths in this household in the last twelve months [23]

CHARACTERISTICS OF THE DECEASED

Record type	Serial No.	Name of deceased	Relationship to head of household	Where did the deceased usually reside? Name of district or town	Office use only			Sex Male=1 Female=2	When was the deceased born: Day		Month		Year		Day, month and year of death: Day		Month		Year	
21	25 26		27		28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
2	01																			
	02																			
	03																			
	04																			
	05																			
	06																			

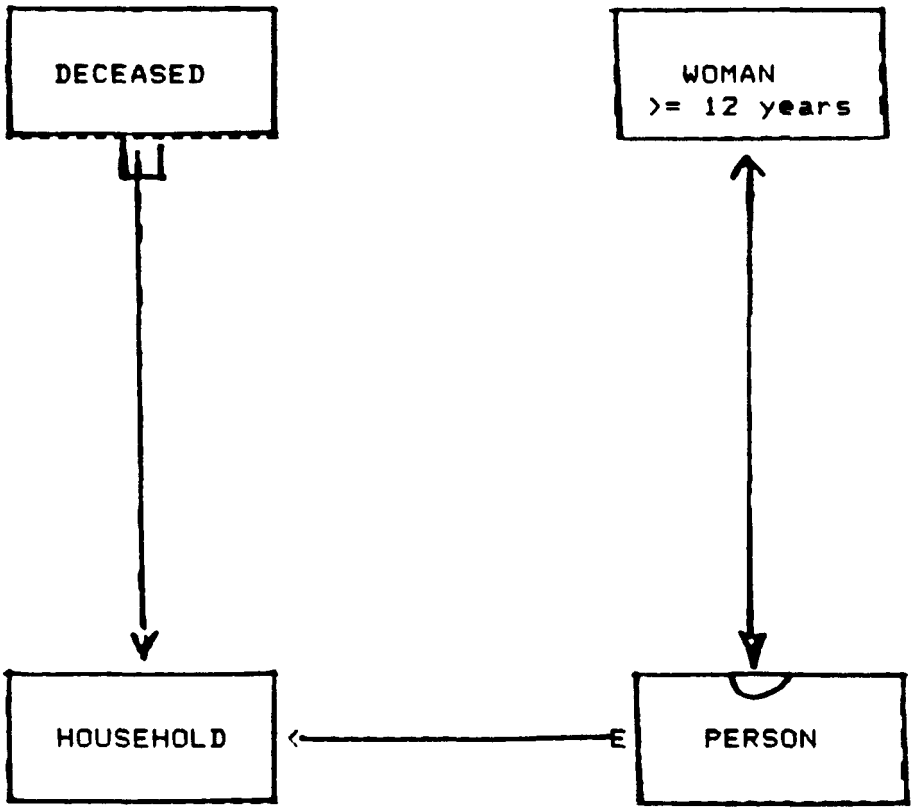
2

FERTILITY CHARACTERISTICS

FOR WOMEN AGED 12 YEARS AND OVER (The questions to be answered by the woman concerned if present)

Record type	Serial no. As per column 24, 25 of Record type 1	Who is the respondent woman Concerned=1 Proxy=2	Have you ever been pregnant Yes=1 No=2	Age at first pregnancy	Have you ever given birth to a live child Yes=1 No=2	Age at first live birth	Number of children of your own living in this household: Sons	Daughters	Number of children of your own living elsewhere: Sons	Daughters	Have you ever given birth to a child who later died: Sons	Daughters	Total number of children ever born: Sons	Daughters	Last live birth: When did you have your last live birth: Day	Month	Year	Sex of last live birth Male=1 Female=2	Is this child still alive Alive=1 Dead=2
23	24 25	26	27	28 29	30	31 32	33	34	35	36	37	38	39	40	41 42	43 44	45 46	47	48
3																			

2.1 OBJECT SYSTEM, OBJECT GRAPH



4.1

Record description for HOUSEHOLD - including derived variables

Field	Position	Bytes	Type	Dec	Comments
HID	1 - 13	13	C	0	Household identity
AREA	1 - 3	3	C	0	Admin. area
DIVISION	4 - 5	2	C	0	Division
SUBDIV	6 - 7	2	C	0	Subdivision
EANR	8 - 9	2	C	0	E.A.Number
HHNR	10 - 13	4	C	0	Household Number
--- HID is made up of AREA, DIVISION, EANR and HHNR -----					
INTD	14 - 15	2	C	0	Day of interview
INTM	16 - 17	2	C	0	Month of interview
STRATUM	18	1	C	0	Stratum
SRN	19 - 23	5	C	0	Serial number
SOH	24 - 25	2	(N)	0	Size of household

4.1

Record description for PERSON - excluding derived variables

Field	Position	Bytes	Type	Dec	Comments
HID	1 - 13	13	C	0	Household identity
AREA	1 - 3	3	C	0	Admin. area
DIVISION	4 - 5	2	C	0	Division
SUBDIV	6 - 7	2	C	0	Subdivision
EANR	8 - 9	2	C	0	E.A.Number
HHNR	10 - 13	4	C	0	Household Number
--- HID is made up of AREA,DIVISION,EANR and HHNR -----					
INTD	14 - 15	2	C	0	Day of interview
INTM	16 - 17	2	C	0	Month of interview
STRATUM	18	1	C	0	Stratum
SRN	19 - 23	5	C	0	Serial number
PID	24 - 25	2	C	0	Person number
USMEM	26	1	C	0	Usual members
RELTH	27	1	C	0	Relation to head
SEX	28	1	C	0	SEX(M=1,F=2)
AGE	29				Age- last birthday
MONTHS	29 - 30	2	(N)	0	Months
YEARS	31 - 32	2	(N)	0	Years
PLBIRTH	33 - 35	3	C	0	Place of birth
CITSHIP	36	1	C	0	Citizenship
RESID	37 - 39	3	C	0	Previous place of residence
ETHNIC	40	1	C	0	Ethnic group
MARSTAT	41	1	C	0	Marital status
SCHOOL	42	1	C	0	School attendance
HLEVAC	43 - 44	2	C	0	Level of education
VOCTR	45	1	(N)	0	Vocational train.
WORK	46 - 47	2	C	0	Main kind of work
NOWORK	48	1	C	0	Reason for not working

4.1

Record description for WOMEN)= 12 YEARS

Field	Position	Bytes	Type	Dec	Comments
HID	1 - 13	13	C	0	Household identity
AREA	1 - 3	3	C	0	Admin. area
DIVISION	4 - 5	2	C	0	Division
SUBDIV	6 - 7	2	C	0	Subdivision
EANR	8 - 9	2	C	0	E.A. Number
HHNR	10 - 13	4	C	0	Household Number
--- HID is made up of AREA, DIVISION, EANR and HHNR -----					
INTD	14 - 15	2	C	0	Day of interview
INTM	16 - 17	2	C	0	Month of interview
STRATUM	18	1	C	0	Stratum
SRN	19 - 23	5	C	0	Serial number
PID	24 - 25	2	C	0	Person number
WOMAN	26	1	C	0	Respondent
PREG	27	1	C	0	Ever pregnant
AGEP	28 - 29	2	N	0	Age at 1st pregnacy
BIRTHLC	30	1	C	0	Birth - live child
AGELC	31 - 32	2	N	0	Age at 1st live birth
BOYSIH	33	1	N	0	Sons in household
GIRLSIH	34	1	N	0	Daughters in household
BOYSE	35	1	N	0	Sons elsewhere
GIRLSE	36	1	N	0	Daughters elsewhere
BOYSD	37	1	N	0	Sons deceased
GIRLSD	38	1	N	0	Daughters deceased
BOYSTOT	39	1	N	0	Sons - total born
GIRLSTOT	40	1	N	0	Daughters - total number born
BIRTHL	41				Last live birth
BDAYL	41 - 42	2	C	0	Day
BMONTHL	43 - 44	2	C	0	Month
BYEARL	45 - 46	2	C	0	Year
SEXC	47	1	C	0	Sex of last live birth
ALIVE	48	1	C	0	Child still alive

Record description for DECEASED - excluding derived variables

Field	Position	Bytes	Type	Dec	Comments
HID	1 - 13	13	C	0	Household identity
AREA	1 - 3	3	C	0	Admin. area
DIVISION	4 - 5	2	C	0	Division
SUBDIV	6 - 7	2	C	0	Subdivision
EANR	8 - 9	2	C	0	E.A.Number
HHNR	10 - 13	4	C	0	Household Number
--- HID is made up of AREA,DIVISION,EANR and HHNR -----					
INTD	14 - 15	2	C	0	Day of interview
INTM	16 - 17	2	C	0	Month of interview
STRATUM	18	1	C	0	Stratum
SRN	19 - 23	5	C	0	Serial number
DID	24 - 25	2	C	0	Deceased's serial number
RELTHD	26	1	C	0	Relation to head
PLACED	27 - 29	3	C	0	Place of residence
SEXD	30	1	C	0	Sex (M=1,F=2)
DATEB	31				Date of birth
DAYB	31 - 32	2	C	0	Day
MONTHB	33 - 34	2	N	0	Month
YEARB	35 - 36	2	N	0	Year
DATED	37				Date of death
DAYD	37 - 38	2	C	0	Day
MONTHD	39 - 40	2	N	0	Month
YEARD	41 - 42	2	N	0	Year

2.2 VARIABLE DESCRIPTION

OBJECT GROUP: HOUSEHOLD

NAME OF VARIABLE	DESCRIPTION/ROLE	VALUES, CODES etc
HID	Household identification	
AREA	Administrative area (see codes for PERSON.PLBIRTH)	000-999
DIVISION	Division	
	pos 1	0-6
	pos 2	0-9
SUBDIV	Subdivision	01-33
EANR	E. A. number	
	pos 1	0-7
	pos 2	0-9
HHNR	Household number	0001-9999
INTD	Day of actual interview (18/8 - 17/9 ??)	01-31
INTM	Month of actual interview	08-09

DERIVED VARIABLES:

SOH	Size of household = number of persons excluding visitors	01-40

2.2 VARIABLE DESCRIPTION

2:10

OBJECT GROUP: PERSON 1(2)

NAME OF VARIABLE	DESCRIPTION/ROLE	VALUES, CODES etc
HID	Household identification	
AREA	Administrative area (see codes for PERSON.PLBIRTH)	000-999
DIVISION	Division	
	pos 1	0-6
	pos 2	0-9
SUBDIV	Subdivision	01-33
EANR	E. A. number	
	pos 1	0-7
	os 2	0-9
HHNR	Household number	0001-9999
PID	Serial number which identifies person within the household	01-40
USMEM	Usual members of the household	
	Present last night	= 1
	Absent last night	= 2
	Visitors	= 3
RELTH	Relationship to head of household	
	Head	= 1
	Spouse	= 2
	Son / Daughter	= 3
	Son / Daughter-in-law	= 4
	Mother / Father-in law	= 5
	Other relative	= 6
	Not related	= 7
SEX	Sex	
	Male	= 1
	Female	= 2
AGEM	Age in months if AGE < 1 Number of months above AGE if 1 <= AGE <= 4 00 if AGE >= 5 (see Interviewer's manual page 10)	00-11
AGE	Age at last birthday in years	00-99
PLBIRTH	Place of birth (State district if born in Zimbabwe. State country if born outside Zimbabwe. See Coding and editing manual.)	000-999
CITSHIP	Citizenship	
	Zimbabwe	= 1
	Mozambique	= 2
	Malawi	= 3
	Zambia	= 4
	Other African	= 5
	Other non-African	= 6

2:11

NAME OF VARIABLE	DESCRIPTION/ROLE	VALUES, CODES etc
HID	Household identification	
AREA	Administrative area (see codes for PERSON.PLBIRTH)	000-999
DIVISION	Division	
	pos 1	0-6
	pos 2	0-9
SUBDIV	Subdivision	01-33
EANR	E. A. number	
	pos 1	0-7
	os 2	0-9
HHNR	Household number	0001-9999
PID	Serial number which identifies person within the household	01-40
WOMAN	Who is the respondent woman	Concerned = 1 Proxy = 2
PREG	Have you ever been pregnant	Yes = 1 No = 2
AGEP	Age at first pregnancy	12-49,bb
BIRTHLC	Have you ever given birth to a live child	Yes = 1 No = 2 If never pregnant = b
AGELC	Age at first live birth	15-49,bb
	Number of children of your own living in this household:	
BOYSIH	- Sons	0-9
GIRLSIH	- Daughters	0-9
	Number of children of your own living elsewhere:	
BOYSE	- Sons	0-9
GIRLSE	- Daughters	0-9
	Have you ever given birth to a child who later died:	
BOYSD	- Sons	0-9
GIRLSD	- Daughters	0-9
	Total number of children ever born:	
BOYSTOT	- Sons	0-9
GIRLSTOT	- Daughters	0-9

2.2 VARIABLE DESCRIPTION

OBJECT GROUP: WOMAN >= 12 years 2(2)

NAME OF VARIABLE	DESCRIPTION/ROLE		VALUES, CODES etc
BIRTHL	When did you have your last live birth		
BDAYL	Day		01-31
BMONTHL	Month		01-12
BYEARL	Year		00-87
		If BIRTHLC = 2	= bbbbbb
SEXC	Sex of last live birth	Male	= 1
		Female	= 2
		If BIRTHLC = 2	= b
ALIVE	Is this child still alive	Alive	= 1
		Dead	= 2
		If BIRTHLC = 2	= b

2.2 VARIABLE DESCRIPTION

OBJECT GROUP: DECEASED

NAME OF VARIABLE	DESCRIPTION/ROLE	VALUES, CODES etc
HID	Household identification (see HOUSEHOLD)	
DID	Serial number which identifies person deceased within the household during the last twelve months	01-12
RELTHD	Relationship to head of household (see PERSON.RELTH)	1-7
PLACED	Where did the deceased usually reside (same codes as PERSON.PLBIRTH)	000-999
SEXD	Sex	Male = 1 Female = 2
DATEB	Date of birth	
DAYB	Day	01-31
MONTHB	Month	01-12
YEARB	Year	00-87
DATED	Date of death (18/8-87 - 17/8-88)	
DAYD	Day	01-31
MONTHD	Month	01-12
YEARD	Year	00-87

DREIVED VARIABLES:

AGED	Age at death : YEARD - YEARB	00-87
AGEMD	For infants only: If AGED = 0: MONTHD - MONTHB If AGED = 1: MONTHD + (12 - MONTHB) If AGED > 1: 0	00-23

2:14

PROCESS	FUNCTION /PROGRAM	RECORD-DESCR.	SYSTEMS FLOW	DESCRIPTION
		PERSON	o	
A60	SELECT AGGREGATE		--- | | ---	Select USMEM = 1,2 Aggregate by HID to get SOH
		HOUSE	o	
		PERSON	o | |----|	
A62	JOIN PROJECT		--- | | ---	Join PERSON, HOUSE where HID=HID. Project to get PERSONH
		PERSONH	o	
A64	SELECT PROJECT		--- | | ---	Select PERSON with SEX=2 and AGE>=12. Project to record-description incl just variables needed
		&PW	o	Females>=12 years
		WOMAN	o | |----|	
A66	OJOIN		--- | | ---	Join &PW,WOMAN where HID,PID= HID,PID. Create a record for every record in &PW completed with - WOMAN-variables; if corresponding - blanks; else
		WOMANP	o	All females >= 12
A68	DEFINE		--- | | ---	If PREG = blank: Create WOMANP-records like those with PREG=2
		WOMANP	o	

2.4 INFORMATION NEEDS

1.00 Total population.

Nr	OBJECTS	for OBJECT with	give	by
1.01	PERSON, HOUSEHOLD (SOH)	for PERSON with USMEM = 1,2	give number	by AGE (gr) * SOH (gr) * SEX
		and		
1.01.1		STRATUM=0		
1.01.2		STRATUM=4		
1.02-04				
	PERSON	for PERSON	give number	by AGE (gr) * (see below) * SEX
1.02				MARSTAT
1.03				ETHNIC
1.04				CITSHIP
1.05	PERSON	for PERSON	give number	by AGE * ETHNIC * SEX

2.4 INFORMATION NEEDS

2.00 Household characteristics.

Nr	OBJECTS	for OBJECT with	give	by
2.01	PERSON, HOUSEHOLD (SOH)	for PERSON with RELTH = 1	give number	by AGE (gr) * SOH (gr) * SEX
		and		
2.01.1		STRATUM=0		
2.01.2		STRATUM=4		
2.02-07				
	PERSON	for PERSON with RELTH = 1	give number	by AGE (gr) * (see below) * SEX
2.02				MARSTAT
2.03				ETHNIC
2.04				CITSHIP
2.05				SCHOOL
		and		
2.05.1		STRATUM=0		
2.05.2		STRATUM=4		
2,06				HLEVAC (gr)
		and		
2.06.1		STRATUM=0		
2.06.2		STRATUM=4		
2.07				WORK (gr)
		and		
2.07.1		STRATUM=0		
2.07.2		STRATUM=4		
2.1	PERSON	for PERSON	give number	by AGE (gr) * RELTH * SEX
2.2-4				
	PERSON	for PERSON with USMEM = 1,2	give number	by AGE (gr) * (see below) * SEX
2.2				ETHNIC
2.3				CITSHIP
2.4				MARSTAT

Appendix 3

SAS Display Manager

3:2

Color and Highlighting Commands

Specify color and highlight codes (see **Table 7.5**) in the following commands to set and change color and highlighting:

CBANNER | CBA *color* [*highlight*]
: changes color or highlighting of the screen border, line numbers (if any), and screen description of the active screen. This command is used with display manager, but not full-screen procedures.

CPROT *color* [*highlight*]
: changes color or highlighting of the protected fields of the active screen. In display manager this command affects the text area of the SAS log and SAS output screens. CPROT can also be used with full-screen procedures.

CSOURCE | CSO *color* [*highlight*]
: changes the color or highlighting for SAS source lines on the SAS log screen. This command is not used in full-screen procedures.

CUNPROT | CUN *color* [*highlight*]
: changes the color or highlighting for unprotected fields on full-screen procedure and display manager screens.

Changes made with color and highlighting commands are also saved, but these changes do not override color and highlighting changes made with the ESC key.

KEYBOARD LAYOUTS FOR SUPPORTED TERMINALS

This section contains a figure for each terminal on which the use of full-screen products is supported. Each figure shows the position of function keys and editing keys on the terminal keyboard.

In the figure for your terminal each editing key is labeled with the editing command that it is defined to execute. (Refer to **EDITING KEYS** earlier in this chapter for a list of editing commands.) Function keys are labeled F*n* where *n* is the function key number. Be aware that the actual key on your keyboard may contain a different label.

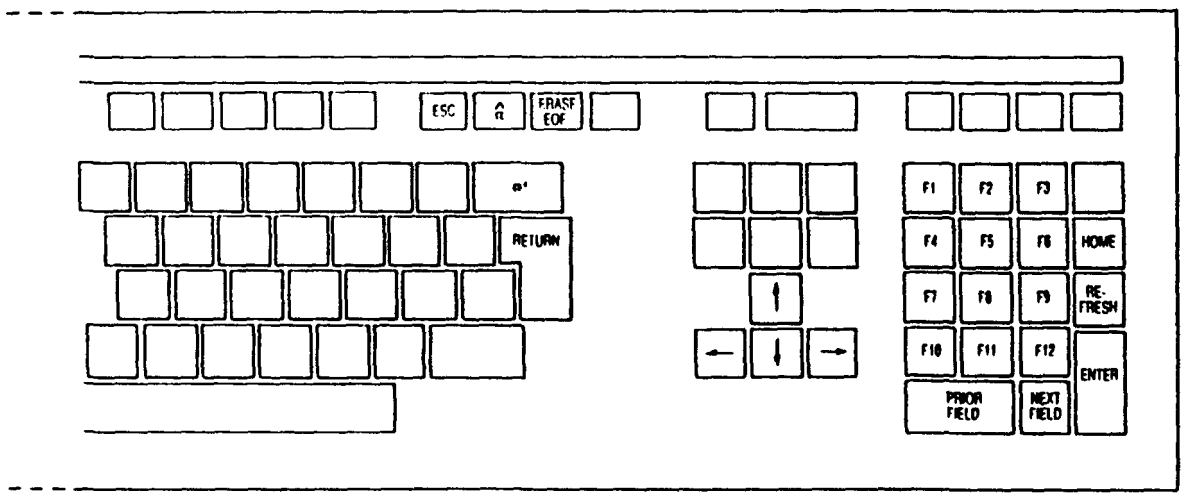


Figure 7.1 Keyboard Layout Representing Digital VT220, VT240, and VT241

Note that function keys may be defined to execute different commands from different screens. Enter the KEYS command to see the list of current function key definitions for the active screen. When the KEYS command executes, the active screen clears and the function key definition screen displays a column of function key numbers and a column of commands. **Screen 7.4** shows a portion of the function key definition screen.

Screen 7.4 indicates that F3 issues the END command during PROC FSEDIT execution, so to exit from the function key definition screen, you press F3. Press F8 to scroll forward and view the remainder of the function key definition screen.

The SAS System allows twenty-four function key definitions; however, the number of function keys that you can define is determined by your terminal. Many terminals have twelve or fewer function keys. If a function key is not defined to issue a command, you must either define a key to issue the command or type the command on the command line and press ENTER.

Many function keys are predefined at the Institute. **Table 7.4** shows the predefined (or default) settings provided by the Institute for the display manager primary screens.

Table 7.4 Default Function Key Definitions for Display Manager

Function Keys	Program Editor	SAS Log	SAS Output
1	HELP	HELP	HELP
2	PRINT	PRINT	PRINT
3	SUBMIT	END	END
4	RECALL		COMMAND
5	RFIND	RFIND	RFIND
6	RCHANGE		
7	BACKWARD	BACKWARD	BACKWARD
8	FORWARD	FORWARD	FORWARD
9	SPLIT	SPLIT	SPLIT
10	LEFT	LEFT	LEFT
11	RIGHT	RIGHT	RIGHT
12	CURSOR	CURSOR	CURSOR

You can define or redefine function keys to execute any valid command for a particular screen. For display manager's program editor screen, you can also define function keys to execute line commands.

For convenient reference, **Table 7.3** provides a list of available line commands, grouped by function. These line commands are defined in the "SAS Display Manager" chapter in the *SAS User's Guide: Basics* and the "SAS Display Manager" appendix in SAS user's guides for each SAS software product.

Table 7.3 Display Manager Line Commands

Single Commands

A[*n*],B[*n*]
C[*n*]
COLS
D[*n*]
I[*n* | A[*n*] | B[*n*]]
M[*n*]
MASK
O[*n*]
P[*n*]
R[*n*]
TF
TS

Block Commands

CC
DD
MM
OO
PP
RR[*n*]

Special Shift Commands

>[*n*]
<[*n*]
>>[*n*]
<<[*n*]
)[*n*]
([*n*]
))[*n*]
(([*n*]

INVOKING A FULL-SCREEN PROCEDURE

If you attempt to invoke a full-screen procedure before you have identified your terminal, the SAS System prompts for the device name. Refer to **Table 7.1** to find the device name for your terminal. **Screen 7.3** illustrates how to invoke the FSEDIT procedure during a line-mode session.

Table 7.2 Display Manager Command-Line Commands

Program Editor	SAS Log	SAS Output	Command
X	X	X	ASSIGN *filename* | *fileref*
X			AUTOADD [ON | OFF]
X	X	X	AUTOEXEC [SASEXEC | *fileref*]
X	X	X	AUTOSHOW OFF | ON
X	X	X	BACKWARD | BAC [*n* | MAX]
X	X	X	BOTTOM | BOT
X	X	X	BYE
X	X	X	CAPS OFF | ON
X *	X *	X *	CBANNER | CBA *color* [*highlight*]
X	X	X	CHANGE *str1 str2* [ALL | WORD | SUFFIX | PREFIX]
		X	COMMAND | CMD
X *	X *	X *	CPROT *color* [*highlight*]
	X *		CSOURCE | CSO *color* [*highlight*]
X *	X *	X *	CUNPROT | CUN *color* [*highlight*]
X	X	X	CURSOR [*rownumber* | *rownumber columnnumber*]
X	X	X	FIND *str1 str2* [WORD | SUFFIX | PREFIX]
X	X	X	FORWARD | FOR [*n* | MAX]
X	X	X	HELP [*topic*]
X	X	X	HSCROLL HALF | PAGE | *n*
X			INCLUDE *fileref* | *filename* | *linenumber* | *linenum1* | *linenum2*
X	X	X	KEYS
X	X	X	LEFT [*n* | MAX]
X			LINESIZE *n*
X	X	X	LOCATE | LOC *n*
X		X	LOG
X			NODMS
X			NULLS ON | OFF
X			NUMBER | NUMS ON | OFF
X	X	X	OUTPUT [ON | OFF | TOP]
X	X	X	PRINT [ALL]
X	X	X	PROGRAM
X			RCHANGE | RC
X			RECALL
X			RESET
X	X	X	RFIND | RF
X	X	X	RIGHT [*n* | MAX]
X	X	X	RULE OFF | ON
X	X	X	SAVE *fileref* | *'filename'*
X	X	X	SCREEN OFF | ON
X	X	X	SPLIT
X			SUBMIT ['*SAS statement;*']
X	X	X	TOP
X	X	X	VSCROLL HALF | PAGE | *n*
X	X	X	X *hostcommand* | *'hostcommand'*

* These commands are effective only if your terminal has extended color or highlighting attributes. (See **EXTENDED COLOR AND HIGHLIGHTING** later in this chapter.)

Appendix 4

PROC TABULATE

```
PROC FREQ DATA=censlib.person;
  TABLES
       relth *
       sex
       /NOCOL NOROW NOPERCENT;
    ;
  TITLE2  Example 1: Frequencies using PROC FREQ';
RUN;
```

Example 1: Frequencies using PROC FREQ

TABLE OF RELTH BY SEX

RELTH(Relationship to head)

	SEX(Sex)		
Frequency	Male	Female	Total
Head	60	20	80
Spouse	5	58	63
Child	163	131	294
S/D in law	6	6	12
F/M in law	8	11	19
Other	51	45	96
Not related	1	1	2
Total	294	272	566

```
PROC TABULATE DATA=censlib.person;
  CLASS sex relth;
  TABLE
       relth ALL ,
       sex ALL
    ;
  TITLE2 'Example 2: Frequencies using PROC TABULATE';
RUN;
```

Example 2: Frequencies using PROC TABULATE

	Sex		
	Male	Female	ALL
	N	N	N
Relationship to head			
Head	60.00	20.00	80.00
Spouse	5.00	58.00	63.00
Child	163.00	131.00	294.00
S/D in law	6.00	6.00	12.00
F/M in law	8.00	11.00	19.00
Other	51.00	45.00	96.00
Not related	1.00	1.00	2.00
ALL	294.00	272.00	566.00

```
PROC TABULATE DATA=censlib.person;
  CLASS sex relth;
  TABLE
       relth ALL ,
       sex ALL
    ;
  KEYLABEL ALL='Total' N='Frequency';
  TITLE2 'Example 3: Frequencies using PROC TABULATE';
  TITLE3 'Using the KEYLABEL statement';
RUN;
```


Example 3: Frequencies using PROC TABULATE
Using the KEYLABEL statement

	Sex		
	Male	Female	Total
	Frequency	Frequency	Frequency
Relationship to head			
Head	60.00	20.00	80.00
Spouse	5.00	58.00	63.00
Child	163.00	131.00	294.00
S/D in law	6.00	6.00	12.00
F/M in law	8.00	11.00	19.00
Other	51.00	45.00	96.00
Not related	1.00	1.00	2.00
Total	294.00	272.00	566.00

```
PROC TABULATE DATA=censlib.person;
  CLASS sex relth;
  TABLE
       relth ALL='Total'      ,
       ( sex ALL='Total')     *
       (N='Frequency'*F=12. )
       /RTS=15
    ;
  TITLE2 'Example 4: Frequencies using PROC TABULATE';
  TITLE3 'Labels and cell sizes';
RUN;
```

Example 4: Frequencies using PROC TABULATE
Labels and cell sizes

	Sex		
	Male	Female	Total
	Frequency	Frequency	Frequency
Relationship to head			
Head	60	20	80
Spouse	5	58	63
Child	163	131	294
S/D in law	6	6	12
F/M in law	8	11	19
Other	51	45	96
Not related	1	1	2
Total	294	272	566

```
PROC TABULATE DATA=censlib.person
              MISSING
              FORMCHAR='            ';
  CLASS sex relth;
  TABLE
       relth ALL='Total'      ,
       ( sex ALL='Total')     *
       (N='Frequency'*F=12. )
       /RTS=15
    ;
  TITLE2 'Example 5a: Frequencies using PROC TABULATE';
  TITLE3 'Labels and cell sizes';
  TITLE4 'Borders removed. Missing values valid';
RUN;
```

Example 5a: Frequencies using PROC TABULATE
Labels and cell sizes
Borders removed. Missing values valid

	Sex		
	Male	Female	Total
	Frequency	Frequency	Frequency
Relationship to head			
Head	60	20	80
Spouse	5	58	63
Child	163	131	294
S/D in law	6	6	12
F/M in law	8	11	19
Other	51	45	96
Not related	1	1	2
Total	294	272	566

```
PROC TABULATE DATA=censlib.person
              MISSING
              NOSEPS ;
  CLASS sex relth;
  TABLE
       relth='RELATIONSHIP TO HEAD' ALL='TOTAL'
       ( sex='SEX' ALL='TOTAL')    *
       (N=' '*F=12. )
       /RTS=15
    ;
  TITLE2 'Example 5b: Frequencies using PROC TABULATE';
  TITLE3 'Labels and cell sizes';
  TITLE4 'NOSEPS option. Missing values valid';
RUN;
```

Example 5b: Frequencies using PROC TABULATE
Labels and cell sizes
NOSEPS option. Missing values valid

	SEX		
	Male	Female	TOTAL
RELATIONSHIP TO HEAD			
Head	60	20	80
Spouse	5	58	63
Child	163	131	294
S/D in law	6	6	12
F/M in law	8	11	19
Other	51	45	96
Not related	1	1	2
TOTAL	294	272	566

```
PROC TABULATE DATA=censlib.person
              MISSING
              ;
  CLASS sex relth;
  TABLE
       relth=' ' ALL='TOTAL'      ,
       ( sex ALL='TOTAL')         *
       (N=' '*F=12. )
       /RTS=15 MISSTEXT='-' BOX=relth CONDENSE
    ;
  TITLE2 'Example 6: Frequencies using PROC TABULATE';
  TITLE3 'Labels and cell sizes';
  TITLE4 'MISSTEXT, BOX, CONDENSE options';
RUN;
```

Example 6: Frequencies using PROC TABULATE
Labels and cell sizes
MISSTEXT, BOX, CONDENSE options

Relationship to head	Sex		TOTAL
	Male	Female	
Head	60	20	80
Spouse	5	58	63
Child	163	131	294
S/D in law	6	6	12
F/M in law	8	11	19
Other	51	45	96
Not related	1	1	2
TOTAL	294	272	566

```
PROC TABULATE DATA=censlib.person
              MISSING
              ;
  CLASS sex relth;
  VAR years;
  TABLE
       relth=' ' ALL='TOTAL'      ,
       ( sex ALL='TOTAL')         *
       years                      *
       (N='N'*F=6.
        MEAN='Mean'*F=8.1
        STD='Std dev'*F=8.3 )
       /RTS=15 MISSTEXT='-' BOX=relth CONDENSE
    ;
  TITLE2 'Example 7: Statistics using PROC TABULATE';
  TITLE3;
  TITLE4;
RUN;
```

Example 7: Statistics using PROC TABULATE

Relationship to head	Sex						TOTAL
	Male			Female			
	Age in years			Age in years			Age in years
	N	Mean	Std dev	N	Mean	Std dev	N
Head	60	45.1	12.996	20	40.1	11.849	80
Spouse	5	48.6	7.403	58	32.9	10.819	63
Child	163	10.7	7.177	131	11.6	7.637	294
S/D in law	6	21.8	15.198	6	7.8	10.962	12
F/M in law	8	9.5	8.246	11	29.2	21.692	19
Other	51	20.8	17.027	45	25.6	20.783	96
Not related	1	60.0	-	1	20.0	-	2
TOTAL	294	20.5	17.820	272	21.2	16.191	566

(CONTINUED)

Example 7: Statistics using PROC TABULATE

Relationship to head	TOTAL	
	Age in years	
	Mean	Std dev
Head	43.8	12.830
Spouse	34.1	11.377
Child	11.1	7.386
S/D in law	14.8	14.597
F/M in law	20.9	19.686
Other	23.1	18.937
Not related	40.0	28.284
TOTAL	20.8	17.045

```
PROC TABULATE DATA=censlib.person
              MISSING
              ;
  CLASS area sex relth;
  TABLE
       (area=' ' ALL='ALL AREAS') *
       (relth=' ' ALL='TOTAL')          ,
       (sex ALL='TOTAL')           *
       (N=' '*F=12. )
       /RTS=30 MISSTEXT='-' CONDENSE
        BOX='Administrative area * Relationship to head'
    ;
  TITLE2 'Example 8: Frequencies using PROC TABULATE';
  TITLE3;
  TITLE4 '3 CLASS variables';
RUN;
```

Example 8: Frequencies using PROC TABULATE

3 CLASS variables

Administrative area * Relationship to head		Sex		
		Male	Female	TOTAL
Miscoded	Head	30	8	38
	Spouse	2	29	31
	Child	72	68	140
	S/D in law	1	3	4
	F/M in law	2	1	3
	Other	16	10	26
	TOTAL	123	119	242
Region A	Head	30	12	42
	Spouse	3	29	32
	Child	91	63	154
	S/D in law	5	3	8
	F/M in law	6	10	16
	Other	35	35	70
	Not related	1	1	2
	TOTAL	171	153	324
ALL AREAS	Head	60	20	80
	Spouse	5	58	63
	Child	163	131	294
	S/D in law	6	6	12
	F/M in law	8	11	19
	Other	51	45	96
	Not related	1	1	2
	TOTAL	294	272	566

```
OPTIONS NONUMBER NODATE LS=78 PS=60;
TITLE;

PROC TABULATE DATA=censlib.person
              MISSING
              ;
  CLASS sex relth;
  TABLE
       relth=' ' ALL='TOTAL'                ,  /* Row dimension    */
       (sex ALL='TOTAL')                    *  /* Column dimension */
       (PCTN<relth ALL>='%'*F=10.1)
       /RTS=15 MISSTEXT='-' BOX=relth CONDENSE
    ;
  TITLE2 'Example 9a: Percentages on categories for the';
  TITLE3 'variable RELATIONSHIP TO HEAD OF HOUSEHOLD';
  TITLE4;
RUN;

PROC TABULATE DATA=censlib.person
              MISSING
              ;
  CLASS sex relth;
  TABLE
       relth=' ' ALL='TOTAL'                ,  /* Row dimension    */
       (sex ALL='TOTAL')                    *  /* Column dimension */
       (PCTN<sex ALL>='%'*F=10.1)
       /RTS=15 MISSTEXT='-' BOX=relth CONDENSE
    ;
  TITLE2 'Example 9b: Percentages on categories for';
  TITLE3 'the variable SEX';
  TITLE4;
RUN;

PROC TABULATE DATA=censlib.person
              MISSING
              ;
  CLASS sex relth;
  TABLE
       relth=' ' ALL='TOTAL'                ,  /* Row dimension    */
       (sex ALL='TOTAL')                    *  /* Column dimension */
       (N*F=6. PCTN<relth ALL>='%'*F=10.1)
       /RTS=15 MISSTEXT='-' BOX=relth CONDENSE
    ;
  TITLE2 'Example 10: Frequencies and percentages on categories';
  TITLE3 'for the variable RELATIONSHIP TO HEAD OF HOUSEHOLD';
  TITLE4;
RUN;
```

Example 9a: Percentages on categories for the variable RELATIONSHIP TO HEAD OF HOUSEHOLD

Relationship to head	Sex		
	Male	Female	TOTAL
	%	%	%
Head	20.4	7.4	14.1
Spouse	1.7	21.3	11.1
Child	55.4	48.2	51.9
S/D in law	2.0	2.2	2.1
F/M in law	2.7	4.0	3.4
Other	17.3	16.5	17.0
Not related	0.3	0.4	0.4
TOTAL	100.0	100.0	100.0

Example 9b: Percentages on categories for the variable SEX

Relationship to head	Sex		
	Male	Female	TOTAL
	%	%	%
Head	75.0	25.0	100.0
Spouse	7.9	92.1	100.0
Child	55.4	44.6	100.0
S/D in law	50.0	50.0	100.0
F/M in law	42.1	57.9	100.0
Other	53.1	46.9	100.0
Not related	50.0	50.0	100.0
TOTAL	51.9	48.1	100.0

Example 10: Frequencies and percentages on categories for the variable RELATIONSHIP TO HEAD OF HOUSEHOLD

Relationship to head	Sex					
	Male		Female		TOTAL	
	N	%	N	%	N	%
Head	60	20.4	20	7.4	80	14.1
Spouse	5	1.7	58	21.3	63	11.1
Child	163	55.4	131	48.2	294	51.9
S/D in law	6	2.0	6	2.2	12	2.1
F/M in law	8	2.7	11	4.0	19	3.4
Other	51	17.3	45	16.5	96	17.0
Not related	1	0.3	1	0.4	2	0.4
TOTAL	294	100.0	272	100.0	566	100.0

Appendix 5

APPENDIX 5: Sample SAS programs for the ICDS

```
TITLE;
LIBNAME censlib '[STEN.ICDS.CENSLIB]';
FILENAME person '[STEN.ICDS]ICDS_PERSON.DAT';

DATA censlib.person(LABEL='ICDS household members');
  INFILE person OBS=10;
  LENGTH
         hid $ 13
         months years 2
         ;
  INPUT
         area      $CHAR3.
         division  $CHAR2.
         subdiv    $CHAR2.
         eanr      $CHAR2.
         hhnr      $CHAR4.
         intd      $CHAR2.
         intm      $CHAR2.
         stratum   $CHAR1.
         srn       $CHAR5.
         pid       $CHAR2.
         usmem     $CHAR1.
         relth     $CHAR1.
         sex       $CHAR1.
         months          2.
         years           2.
         plbirth   $CHAR3.
         citship   $CHAR1.
         resid     $CHAR3.
         ethnic    $CHAR1.
         marstat   $CHAR1.
         school    $CHAR1.
         hlevac    $CHAR2.
         voctr     $CHAR1.
         work      $CHAR2.
         nowork    $CHAR1.
         ;
   hid=area || division || subdiv || eanr || hhnr;
/*=============================================*/
/*   Associating labels  to the variables in  */
/*   SAS Dataset PERSON                        */
/*=============================================*/
 LABEL  area    = 'Administrative area'
        division= 'Division'
        subdiv  = 'Subdivision'
        eanr    = 'Enumeration area'
        hhnr    = 'Household number'
        hid     = 'Household identification'
        pid     = 'Serial number'
        usmem   = 'Usual member'
        relth   = 'Relationship to head'
        sex     = 'Sex'
        agem    = 'Age in months'
        age     = 'Age in years'
        plbirth = 'Place of birth'
```

```
        citship = 'Citizenship'
        resid   = 'Previous place of residence'
        ethnic  = 'Ethnic group'
        marstat = 'Marital status'
        school  = 'School attendance'
        hlevac  = 'Highest academic level'
        voctr   = 'Vocational training years'
        work    = 'Main kind of work'
        nowork  = 'Reason for not working'
        ;
RUN;

/*=============================================*/
/*   Printing metadata for the SAS Dataset     */
/*   PERSON                                     */
/*=============================================*/
PROC CONTENTS DATA=censlib.person;
TITLE2 'Metadata for the SAS Dataset PERSON';
RUN;

/*=============================================*/
/*   Printing the first 6 observations in      */
/*   the SAS Dataset PERSON without formats    */
/*=============================================*/
PROC PRINT DATA=censlib.person(OBS=6) LABEL;
TITLE2 'Listing of the first 6 observations in the SAS Dataset
PERSON';
RUN;
```

```
/*=============================================*/
/*   Defining formats for the variables in     */
/*   SAS Dataset WOMEN                         */
/*=============================================*/
title;
title2 'Formats for the data set WOMEN';

PROC FORMAT library=censlib;
 VALUE $womanf
    '1'='Concerned'
    '2'='Proxy'
    OTHER='Missing'
    ;
 VALUE $pregf
    '1'='Yes'
    '2'='No'
    OTHER='Missing'
    ;
 VALUE $birthlc
    ' '='Never pregnant'
    '1'='Yes'
    '2'='No'
    OTHER='Miscoded'
    ;
 VALUE $sexcf
    ' '='Not applicable'
    '1'='Male'
    '2'='Female'
    OTHER='Miscoded'
    ;
 VALUE $alivef
    ' '='Not applicable'
    '1'='Alive'
    '2'='Dead'
    OTHER='Miscoded'
    ;
run;

/*=============================================*/
/*   Associating formats to the variables in   */
/*   SAS Dataset WOMEN                         */
/*=============================================*/

LIBSEARCH censlib;

PROC DATASETS library=censlib;
 MODIFY women;
 FORMAT AREA $aREAF.
        woman $womanf.
        preg $pregf.
        birthlc $birthlcf.
        sexc $sexcf.
        alive $alivef.
        ;
run; quit;
```

```
/*============================================*/
/*   Associating labels  to the variables in  */
/*   SAS Dataset WOMEN                        */
/*============================================*/
PROC DATASETS library=censlib;
 MODIFY women;
 label  preg    = 'Pregnancy'
        agep    = 'Age at first pregn'
        birthlc = 'Birth to live child'
        agelc   = 'Age at first live birth'
        boysih  = 'No. sons in hh'
        girlsih = 'No. daughters in hh'
        boyse   = 'No. sons elsewhere'
        girlse  = 'No. daughters elsewhere'
        boysd   = 'No. sons dead'
        girlsd  = 'No. daughters dead'
        boystot = 'Total no. sons ever born'
        girlstot= 'Total no. daughters ever born'
        bdayl   = 'Day when last live birth'
        bmonthl = 'Month when last live birth'
        byearl  = 'Year when last live birth'
        sexc    = 'Sex of last live birth'
        alive   = 'Last child still alive'
        ;
run; quit;

/*============================================*/
/*   Printing metadata for the SAS Dataset    */
/*   WOMEN                                    */
/*============================================*/
proc contents;
title2 'Metadata for the SAS Dataset WOMEN';
run;

/*============================================*/
/*   Printing the first 6 observations in     */
/*   the SAS Dataset WOMEN without formats    */
/*============================================*/

proc print data=censlib.women(obs=6) split=' ';
  format _all_;
title2 'Listing of the first 6 observations in the SAS
Dataset WOMEN';
run;
```

```
/*============================================*/
/*   Defining formats for the variables in    */
/*   SAS Dataset DECEASED                     */
/*                                            */
/*   Note: Additional formats not needed      */
/*============================================*/

/*============================================*/
/*   Associating labels  to the variables in  */
/*   SAS Dataset DECEASED                     */
/*============================================*/
LIBSEARCH censlib;

PROC DATASETS library=censlib;
 MODIFY deceased;
 label  dayb    ='Day of birth'
        monthb  ='Month of birth'
        yearb   ='Year of birth'
        dayd    ='Day of death'
        monthd  ='Month of death'
        yeard   ='Year of death'
        ;
run; quit;

/*============================================*/
/*   Printing metadata for the SAS Dataset    */
/*   DECEASED                                 */
/*============================================*/
proc contents data=censlib.deceased;
title2 'Metadata for the SAS Dataset DECEASED';
run;

/*============================================*/
/*   Printing the first 6 observations in     */
/*   the SAS Dataset DECEASED without formats    */
/*============================================*/
proc print data=censlib.deceased(obs=6) split=' ';
  format _all_;
title2 'Listing of the first 6 observations in the SAS
Dataset DECEASED';
run;
```

```
LIBNAME censlib '[STEN.ICDS.CENSLIB]';
/*=============================================*/
/*  PROCESS A60                      SELECT    */
/*      Select USMEM='1','2'                   */
/*=============================================*/
DATA a60sel;
  SET censlib.person;
  *FORMAT _ALL_;              /* Program runs faster */
  IF usmem='1' | usmem='2';   /* Subsetting IF       */
RUN;
/*=============================================*/
/*  PROCESS A60                     AGGREGATE  */
/*      Aggregate on HID to get SOH            */
/*=============================================*/
 * Alternative 1 ;
 DATA a60sum (KEEP=hid soh);
   SET a60sel;
   BY hid;                    /* Provided that a60sel
   LENGTH soh 2;                 is sorted on hid    */
   RETAIN soh;                /* Not needed when using
   IF FIRST.hid THEN soh=0;      a sum statement:
   soh=soh+1;                    soh+1;              */
   IF LAST.hid;
 RUN;
 * Alternative 2;
 PROC FREQ DATA=a60sel;
   TABLES hid/NOPRINT OUT=
     a60sum(KEEP=hid count RENAME=
          (count=soh) );
 RUN;
/*=============================================*/
/*  PROCESS A60                       JOIN     */
/*      Merging A60SEL and A60SUM by HID       */
/*=============================================*/
 DATA censlib.personh;
   MERGE a60sel
         a60sum;              /* The larger dataset to
   BY hid;                       the left            */
 RUN;
/*=============================================*/
/*  PROCESS A60                      REPORT    */
/*      Printing 10 observations from PERSONH  */
/*=============================================*/
 PROC PRINT DATA=_LAST_(OBS=10) DOUBLE LABEL;
   TITLE2 'Dataset PERSONH';
 RUN;
/*=============================================*/
/*  PROCESS A60                      MDSHOW    */
/*      Printing metadata for PERSONH          */
/*=============================================*/
 PROC CONTENTS DATA=censlib.personh;
   TITLE2 'Dataset PERSONH metadata';
 RUN;
```

Appendix 6

APPENDIX 6: Lecture minutes made from overhead slides

Correcting errors

This can be done like:

```
* Alternative 1: ;

DATA censlib.person_c;
  SET censlib.person;

  IF hid='10111023100' & pid='04'    /*** SAY ***/
  THEN DO;
     work='01';
     years='39';
  END;

  /** Then repeat for all observations using the editor */

  IF hid='12111023100' & pid='06'
  THEN DO;
     ..... ;
     ..... ;
  END;

  /** ETC. **/

RUN;


* Alternative 2: ;

DATA _NULL_;
  SET censlib.person;
  FILE '[STEN.ICDS]PERSONW.DAT';
  /*** Then write a flat file using the PUT statement ***/
  PUT  .....  ;
RUN;
```

After this the data might be read into the Datatrieve system and updated.

Finally, just run the SAS input program for creating censlib.person and do another error detecting to make sure that all is correct.

Duplicates

I will define two different kinds of duplicates:

1) "True" duplicates, meaning repeated observations
2) "Partial" duplicates, which repeats some set of identification variables.

True duplicates can be removed using PROC SORT:

```
PROC SORT DATA=censlib.person NODUPLICATES;
RUN;
```

Partial duplicates can be detected and reported using a data step provided that the dataset is sorted on the set of identification variables:

```
DATA dup;
  SET censlib.person;
  BY hid pid;
  IF NOT(FIRST.pid & LAST.pid); /* Subsetting IF stmt */
RUN,

PROC PRINT;
RUN;
```

Error structures

We differ between

- errors within an observation, i.e. a woman has her first child at the age of 8

- errors between observations in the same dataset, i.e. there are more than one household head in the household or a child is older than its parents

- errors between different datasets, i.e. a women age 36 with children alive in the SAS dataset CENSLIB.WOMEN, who does not exist in a household in the SAS dataset CENSLIB.PERSON

Example:

```
* Finding children who are too old in respect to the
  household head;

PROC SORT DATA=censlib.person;
  BY hid relth pid;
RUN;
DATA hh_err1(KEEP=hid relth pid f1 f2);
  SET censlib.person;
  BY hid relth;
  FORMAT _ALL_;
  RETAIN agehh;
  IF relth='1' THEN agehh=years;
  IF relth='1' | relth='2'
  THEN IF NOT(FIRST.relth & LAST.relth)
       THEN f1=1;
  IF relth='3'
  THEN IF agehh-years<12
       THEN f2=1;
  IF f1>. | f2>. THEN OUTPUT;
RUN;
DATA hh_err2;
  SET hh_err1;
  BY hid;
  IF FIRST.hid;
RUN;
DATA hh_err3;
  MERGE censlib.person(IN=in1)
        hh_err2(IN=in2);
  BY hid;
  FORMAT _ALL_;
  IF in2;
RUN;
PROC PRINT LABEL;
RUN;
```

Phases of statistical data processing

1. Data capture
When enumerators fill in the questionnaire in the field

2. Editing/coding
Manually done - centralized at the office or decentralized in the field

3. Data entry
This consists of two parts
- entry with field validation
- correcting errors (sometimes using computerized automated correcting)

4. Organizing data
In this case this means building SAS datasets stored in SAS Data Libraries

---

5. Editing

Editing in general can be split into 3 actions to be taken:

a) Detecting errors
This means do find the overall error structure (macro-editing). It can in SAS be done using the FREQ procedure for single frequency tables on all values there exist in the SAS dataset or for crosstables including 2 or more variables. On pages we can find errors in i.e. the crosstable of "school attendance" (SCHOOL) and "highest level of academic studies" (HLEVAC). The error structure in the table is marked.

SAS-program:

```
PROC FREQ DATA=censlib.person;
  FORMAT _ALL_; /*** To suppress all formats ***/
  TABLES school*hlevac
         work*nowork
         years*relth
         /NOCOL NOROW NOPERCENT;
RUN;
```

TABLE OF SCHOOL BY HLEVAC

SCHOOL HLEVAC

Frequency	00	01	02	03	04	05	Total
1	15	15	11	21	19	14	177
2	0	4	9	8	13	13	119
3	130	0	0	1	0	0	135
Total	145	19	20	30	32	27	431

(Continued)

TABLE OF SCHOOL BY HLEVAC

SCHOOL HLEVAC

Frequency	06	07	08	09	10	11	Total
1	22	27	16	11	6	0	177
2	15	25	3	11	1	17	119
3	1	2	0	0	0	1	135
Total	38	54	19	22	7	18	431

TABLE OF WORK BY NOWORK

WORK NOWORK

Frequency	0	1	3	5	6	Total
00	118	10	13	4	139	284
01	1	0	0	0	0	1
13	9	0	0	0	0	9
40	1	0	0	0	0	1
49	2	0	0	0	0	2
59	1	0	0	0	0	1
61	125	0	2	0	1	128
72	1	0	0	0	0	1
75	2	0	0	0	0	2
77	1	0	0	0	0	1
83	1	0	0	0	0	1
Total	262	10	15	4	140	431

TABLE OF YEARS BY RELTH

6:8

YEARS RELTH

Frequency	1	2	3	4	6	Total
0	0	0	6	0	2	8
1	0	0	10	0	0	10
2	0	0	11	0	3	14
3	0	0	9	0	1	10
4	0	0	6	0	3	9
5	0	0	14	0	1	15
6	0	0	6	0	4	10
7	0	0	18	0	5	23
8	0	0	3	0	5	8
9	0	0	14	0	1	15
10	0	0	10	0	7	17
11	0	0	14	0	1	15
12	0	0	10	0	3	13
13	0	0	14	0	4	18
14	1	0	9	0	3	13
15	1	0	13	0	2	16
16	1	0	8	0	0	9
17	0	0	5	0	0	5
18	0	0	10	0	4	14
19	3	1	0	1	0	5
39	2	1	1	0	0	4
Total	8	2	191	1	49	251

b) Reporting errors
Reporting errors means to select and list all observations that contains any of the errors found above.

SAS-program:

```
* Selecting;

DATA p_error(DROP=f1-f6);
  SET censlib.person;
  LENGTH  f1 f2 f3 f4 f5 f6 $ 1 errflag $ 6;
  /*** Errors in variables YEARS,WORK,NOWORK,RELTH,
       SCHOOL,HLEVAC  ***/
  IF years>=10 THEN DO;
    IF work>'00' & nowork>'0' THEN f1='1';
    IF work='00' & nowork='0' THEN f2='1';
  END;
  IF school='3' & hlevac='00' THEN f3='1';
  IF (relth='1' | relth='2') & years<=15 THEN f4='1';
  IF (relth='3' | relth='4') & years>=40 THEN f5='1';
  IF relth='5' & years<=30 THEN f6='1';
  errflag=f1 || f2 || f3 || f4 || f5 || f6;
  /*** Selecting observations with errors ***/
  IF errflag>' ';
RUN;

* Listing the selected observations;
PROC PRINT DATA=p_error DOUBLE LABEL;
  ID hid pid;
  VAR errflag years relth school hlevac work nowork;
  TITLE 'Errors in SAS dataset CENSLIB.PERSON';
RUN;
```

c) Correcting errors

Firstly the filled-in forms must be found. It is therefore important that a system for storing i.e. boxes or folders is created to facilitate this operation. Then, after writing the correct data on the error list the error correcting can be done at the terminal.
Sometimes automated error correction can be done in the computer using error correcting computerized schemes.
The best way to handle this is by using the SAS full-screen editor SAS/FSP interactively. Therefore it is recommended that this SAS module is bought as soon as possible

Note: The ations above must always be iterated until no more errors can be found in a)

6. Reporting

Reports, tables and results from statistical analysis could be the last of the phases for statistical data processing

Match merging and non-matches

Here is an example that shows how to find the different types of matches/non-matches:

```
* Sort on the variables used for matching;

PROC SORT DATA=censlib.person;
  BY hid pid;
RUN;

PROC SORT data=censlib.women;
  BY hid pid;
RUN;

DATA match /*** (KEEP= list of variables )  ***/
     nonm_p
     nonm_w
     ;
  MERGE censlib.person(IN=inperson)
        censlib.women(IN=inwomen);
  BY hid pid;
  IF inperson & inwomen
  THEN OUTPUT match;
  IF inperson & NOT(inwomen)
  THEN OUTPUT nonm_p;
  IF NOT(inperson) & inwomen
  THEN OUTPUT nonm_w;
RUN;
```

Datastep or procedure?

1. Organize information into SAS datasets	Data step
2. Editing	PROC FREQ PROC PRINT Data step
3. Metadata handling	PROC CONTENTS PROC DATASETS PROC FORMAT Data step
4. Reporting	PROC PRINT PROC TABULATE etc. Data step

Reporting can be done using a lot of different procedures:

- printing SAS datasets	PROC PRINT Data step
- Crosstables	PROC TABULATE PROC FREQ
- Aggregates	PROC SUMMARY
- statistics	PROC TABULATE PROC MEANS PROC SUMMARY PROC UNIVARIATE PROC CORR
- analysis	PROC GLM ... and more
- graphics	Procedures in SAS/GRAPH

Note: It is always possible to use a common word processor or editor to enhance text or layout from SAS procedures before final printing. All reporting from SAS can be stored in ASCII files.

PROC TABULATE (comments)

There are three dimensions

1) page dimension
2) row dimension
3) column dimension

They are marked using **the comma sign**

Variables can be related thru
- nesting using the **asterisk ***
- concatenation using **one or more blanks**

Example nesting : **RELTH*SEX**

```
                        RELTH
          1         2         3         4         5
SEX   1     2   1     2   1     2   1     2   1     2
```

Example concatenation : **RELTH SEX**

```
          RELTH                SEX
  1   2   3   4   5        1       2
```

There are two kinds of variables:

- classification variables, i.e. sex,relth
- analysis variables, i.e. age,income

Note: Statistics are always nested within a classification or an analysis variable; cells (with formats or witdhs) are regarded as nested within the requiered statistic; both cases in the column dimension, thus

```
classification variable,   /* row dimension */
classification variable    /* column dimension */
         *                 /* nesting */
analysis variable
         *                 /* nesting */
statistic
         *                 /* nesting */
F=10.                      /* say */
```

Table 1.01 from the Lagerlof paper

```
PROC TABULATE DATA=censlib.personh
              MISSING
              NOSEPS
              ;
  CLASS years sex soh;
  FORMAT years years5f.
         sex $ sexff.;
  LABEL soh='Size of Household';
  TABLE
       years ALL='All ages',
       (soh ALL='All hh') * (sex=' ' ALL='Both sexes')
                 *
       (N=' '*F=5.)
       /RTS=10 MISSTEXT='-' CONDENSE BOX=years;
  TITLE1 'Table 1.01: Total population by agegroup, size of';
  TITLE2 'household and sex';
  TITLE3 'Frequencies';
  TITLE4;
  FOOTNOTE1 'Visitors are excluded';
RUN;
```

Appendix 7

SAS COURSE May 1988

Participants

Name	Day 29/4	3/5	4/5	5/5	6/5	9/5	10/5	11/5	12/5	13/5
G. MAZARIRE	X	X	X	X	X	X	X	X	X	X
R. SIBANDA	X	X	X	X	X	X	X	X	X	X
ALEX P. MARUFU	X	X	X	X	X	X	X	X	X	X
C. GOVORE	X	X	X	X	X	X	X	X	X	X
M. JAMBWA	X	X	X	X	X	X	X	X	X	X
J. DIRAWO	X	X	X	X	X	X	X	X	X	X
M. [illegible]	X	X	X	X	X	X	X	X	X	X
[illegible]	X	X	X	X	X	X	X	X	X	X
N. MUHWAVA	X	X	X	X	X	X	X	X	X	X
KRN SHONHIWA	X	X	X	X	X	X	X	X	X	X

Appendix 8

Certificate

This is to certify that

Ms M. Magwaza

has successfully completed the SAS course held at the Central Statistical Office 29 April – 13 May 1988

The course covered basic theory in the Statistical Analysis System with hands–on training

THEORY:
- SAS language
- SAS reporting and tabulation procedures

TRAINING;
- Creating SAS datasets from external files
- Data validation and error correction
- Tabulation
- Report writing

The course was arranged by
Central Statistical Office and Statistics Sweden

Harare, 16 May 1988

Dr G.M. Mandishona
Director

Mr S Bäcklund
Systems Analyst
Statistics Sweden

Appendix 9

CENTRAL STATISTICAL OFFICE 12 May, 1988

SAS course evaluation

1. What do you think about the contents of the SAS course? (Useful? easy/difficult? too many different things? too little/too much theory? confusing/easy to follow ? etc..)

2. Is there any topic from the course that you want to know more about?

3. Should some topic be deleted or given less attention in the course? Should something be added or expanded?

4. Any comments about the practical arrangements? (Time schedule? conference rooms? breaks?)

5. How about the balance between theory and practice?

6. Any comments about the teacher? (Possible negative comments will be accepted with equanimity if they are courteously phrased!)

7. Do you think that you will use what you have learnt in the course in the near future. If so, what part(s)?

8. What was best in the course?

9. What was worst in the course?

10. Any other comments:

Note: You can use the backside of the paper as well!

Latest R & D Reports (area ADB) published by Statistics Sweden:

1988:3 Base Operators as a Tool for Systems Development (Bo Sundgren)

1988:4 Development of Systems Design for National Household Surveys - Report from a short-term mission to Harare, Zimbabwe, 12th-28th January, 1988 (Birgitta Lagerlöf)

1988:11 Design of the User Interface for an Object-Oriented Statistical Data Base (Erik Malmborg)

Copies of these reports may be ordered from Statistics Sweden, att. Ingvar Andersson, S-115 81 Stockholm.