PRODUCTION OF AN ANNUAL OPERATIONAL STATISTICS REPORT USING SAS

Wayne Cornelius, Southeastern Cooperative Fish and Game Statistics Project

Introduction

In 1979 the Southeastern Association of Fish and Wildlife Agencies decided to compile a document containing operational statistics of interest to the 16 member agencies, and to revise it annually. The 1979 edition had 41 tables in 109 pages.

The Tennessee Wildlife Resources Agency compiled the 1979 report from each State’s responses to a mail survey questionnaire, using manual techniques and the labor of many typists to produce the draft copy. After the 1979 report was circulated, the Southeastern Cooperative Fish and Game Statistics Project agreed to produce all future reports using available computer resources, especially the SAS system, to reduce the human labor required and to allow more data summary (e.g., averages) to be included than had been attempted.

To produce a draft copy of the 1980 report, after reorganizing the report so as to contain two additional tables, I wrote 43 short SAS programs to read data stored in an OS dataset, modify the data and produce summary statistics when necessary, then copy the information to a print file. This paper describes these programs.

Considerations

Several guidelines for preparing this report came from the experience of people involved with the 1979 report. These are listed as follows:

* reproduce the style of the 1979 report as closely as possible
* revise contents of the report as suggested by the users
* produce totals or averages of data where appropriate
* simplify the process of annually updating the data

In addition to these guidelines, as they reviewed drafts of the 1980 tables I prepared, other staff members offered specific suggestions too numerous to be listed here; often these suggestions motivated special programming which I will describe in this paper.

Style

In the 1979 report almost all tables are laid out across the long dimension of a page. A portion of a representative table (Table 5 in the document) is reproduced in Figure 1. The title is centered at the top of the page. Two lines below the title is a heading, followed by the body of the table. If the table requires more than one page, "(CONTINUED)" is added to the title on the additional pages. A centered page number is placed on the bottom lines. At the end of the table a horizontal line is printed, followed by any required footnotes.

To improve the table illustrated in Figure 1, I added a column of averages. This made the table clearly too wide to fit on one page, so I split the table in half, and I identified the first half as Table 5A and the second half as Table 5B. Splitting the table permitted me to spell out the State names in the headings, which is preferable to abbreviating them, and to add dollar signs to the numbers in the body. Figure 2 illustrates the revised table. I added similar improvements to other tables in the document at every opportunity.

Program Structure

The contents of the 43 tables varied so much that it was impossible to write one program to print all the tables. Instead I wrote 43 programs and collected common features of these programs into a "common" section of source code. The "common" section contains MACROs, SAS options, a MISSING statement and a null TITLE statement.

For each table, I wrote separate sections of source code to name variables, to modify, total or average them, and to specify titles and formats for printing. Though varying from one table to the next, this source code followed a pattern which was essentially the same for all programs.

Figure 1. Representative Table From 1979 Report

<table>
<thead>
<tr>
<th>Resident License Types</th>
<th>AL</th>
<th>AR</th>
<th>FL</th>
<th>GA</th>
<th>KY</th>
<th>LA</th>
<th>MS</th>
<th>MO</th>
<th>NC</th>
<th>SC</th>
<th>VA</th>
<th>TX</th>
<th>WA</th>
<th>WY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resident Fishing (Rod &amp; Reel or Artificial Bait)</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fishing (County Hunt &amp; Line)</td>
<td>1.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fishing (Eligible)</td>
<td>.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sportmen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC Lic</td>
<td>25.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resident 3-Day Fishing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonresident 1-Day Live Bait</td>
<td>25.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

344
The "Common" Program

The source statements for the "COMMON" section are reproduced in Figure 3, and their purposes described below.

An OPTIONS statement specifies NODATE, NONUMBER and page size.
MISSING Y N X are data values used to represent "YES", "NO", and "PRESENT" in certain tables.
MACRO HORIZ prints a horizontal line across a page.
MACRO TOPMARG provides several lines of blank space and is used at the beginning of each new page.
MACRO ENDPAGE prints a blank line followed by a centered page number.
MACROs SOURCE and TNUM together identify respectively the file name and the member name of a data library.
MACROs YEAR and YY specify the report year.
MACROs HEADER, PAGEINIT, and PAGECOUNT are page counters.
MACROs PROLOG contains DATA, RETAIN, INFILE and FILE statements with options and parameters used by most programs.
MACROs STARTUP and ENDOPT are used at the end of each program to print the end of the table.
MACROs SPELOUT expands State abbreviations to full words, and expands divisions from one-letter abbreviations to full words, so that a $STATE format can be used for printing the names. Alternatively, when a fixed format is acceptable, I use a $STATE format stored in a format library (and described later) for this purpose.
MACROs EDTMIN and EDTMAX similarly list one of one-letter abbreviations to seven-letter words; six tables use EDTMAX to list job requirements for employment by the State agencies.
MACROs ISDAVID and NOTDAVID are logical expressions which determine whether a datum is a meaningful number or a footnote reference or remark.

Figure 2. Table Revised for 1980 Report

<table>
<thead>
<tr>
<th>REGION</th>
<th>LICENSE TYPE</th>
<th>NORTH CAROLINA</th>
<th>OKLAHOMA</th>
<th>SOUTH CAROLINA</th>
<th>TENNESSEE</th>
<th>TEXAS</th>
<th>VIRGINIA</th>
<th>NORTH DAKOTA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOTAL HUNTING</td>
<td>613.00</td>
<td>59.00</td>
<td>510.20</td>
<td>57.00</td>
<td>58.70</td>
<td>58.60</td>
<td>80.60</td>
</tr>
<tr>
<td></td>
<td>COYOTE HUNTING</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
</tr>
<tr>
<td></td>
<td>MINK HUNTING</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
</tr>
<tr>
<td></td>
<td>FISHING</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
</tr>
<tr>
<td>SPECIAL FISHING</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td>9.00</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Source Statements in the "Common" Program

OPTIONS NODATE NONUMBER PS=85;
TITLE 'PRELIMINARY DRAFT';
MISSING Y N X;
MACRO HORIZ PUT OVERPRINT '32*';
MACRO TOPMARG PUT III; %
MACRO ENDPAGE PUT '7';
MACRO YEAR '1980' % MACRO YY '1980-81' %
MACRO PROLOG DATA NULA; RETAIN SEITE PAGEINIT FIRSTPAGE 0; INFILE SOURCE ( THUM ) LINE=19 COLUMN=69 END=99; FILE PRINT HEADER=99 L=99 COLUMN=80; %
MACRO STARTUP HORIZ; PUT II; %
MACRO ENDFOOT FS: IF 99 GT 7 THEN DO: PUT;
DO TO 99; END: ENDPAGE; %
MACRO SPELOUT IF STATE='AL' THEN FULLST='ALABAMA';
IF STATE='AR' THEN FULLST='ARKANSAS';
IF STATE='WV' THEN FULLST='WEST VIRGINIA';
IF DIV = 'F' THEN FULLDIV = 'FISH';
IF DIV = 'W' THEN FULLDIV = 'WILDLIFE';
LENST = LENGTH(FULLST); %
MACRO EDIEXPAND
INFORMAT FULLEDI-FULLEDI6 $7.1;
ARRAY FULLEDI(F) FULLEDI-FULLEDI16;
ARRAY LENGDI(J) LENGDI-LENGDI6;
DO J = 1 TO 16;
IF EDUJ = 'A' THEN FULLED = 'HS';
IF EDUJ = 'E' THEN FULLED = 'HS';
IF EDUJ = 'L' THEN FULLED = 'LICENSE';
IF EDUJ = 'P' THEN FULLED = 'ED & WR';
LENED = LENGTH(FULLED);
END: %
MACRO ISDAVID (Z=991X=981X=971)
X=11X=21X=31X=41X=11 X=11 X=21X=31X=41 X=11 %
MACRO NOTDAVID X NS & NOT ISDAVID %

345
Specially created print formats are not really part of the "common" program, but belong to the same discussion, since they are features used by several different programs. The special formats which I made frequent use of were the following.

$STATE prints postal abbreviations as full State names;

$PEE prints the special missing values, _X as "NO" and _Y as "YES";

$YESNO prints the character values (Y,N,S,J) as (YES,NO,SOME,UNKNOWN);

DAVID causes negative integers to print as select

A few other special formats found application in only one or two tables and are not listed here.

Figure 4. Source Statements for Creating Special Formats

PROC FORMAT DNAME=FORMLIST;
VALUE $STATE AL-AlABAMA AR-ARKANSAS
FL-FLORIDA GA-GEORGIA
VA-VIRGINIA WV-WEST VIRGINIA;
VALUE $PEE .N='NO',.Y='YES',.X='YES',.Z='Z';
VALUE $YESNO(MIN=3) Y='YES' N='NO' S='SOME'
UNKNOWN;
VALUE DAVID(MIN=4) -1='(1) -99='VARIIES'
-2='(2) -99='NA'
-3='(3) -99='APPROX';

Program Flow

Although no single program uses every feature I developed, all those features may be combined in a program in the way I am about to describe.

(1) There are sort sequence numbers attached to the data records because we need to have the revised table list items in a different order than the way they are stored. To use these sequence numbers, the program begins with a DATA step and a PROC SORT.

(2) A DATA _NULL_ step calls in the data and identifies it. If the data do not require two page widths for printing, the PROCLOG macro can do this after the macros SOURCE and TXNS identify the stored data.

(3) Further source code modifies data, computes totals or averages and PUTS the results of doing so in a PRINT file. I will describe this feature in more detail in a separate section of the paper.

(4) When a page is nearly full the program must end the page, write the page number and begin the next page.

(5) A LINK section is needed to print headings. This is called by the HEADER parameter in the INFILE statement.

(6) At the end of the input file data, the program needs to end the last page and write footnotes. Another LINK section contains the necessary source code.

(7) For multi-page width tables, the program has to do some extra work. Instead of writing to the PRINT file immediately, it first writes each page width to a separate TEMP file, disregarding tops and bottoms of pages; then it copies each TEMP file to the PRINT file, using the PUT INFILE statement. It uses a separate TITLE for each page width, and a page number increment equal to the number of page widths in the table.

(8) In one program, because of extensive footnote references, I concatenated the footnotes to the data, separating the footnotes from the data with a row of asterisks, and I wrote additional source statements to handle this method of storage. This novel method facilitates the annual revision of footnote references better than including the footnotes in a LINK section in the source statements, so I plan to convert other programs to this method in the future.

Description of the Data

I use one of two patterns for data storage, according to whether the printed table would have State names as entries in the first column or as column headings. When State names are in the first column, the table usually contains quite little information about the State, and this information is easily fitted into the remaining columns of an 80 byte record. There are sixteen records in the data for such a table, except when some States describe different characteristics for their fish and wildlife divisions, or when totals across states are stored as data.

When State names are headings of the table, there is usually a lot of information for each State, such as a list of all its license types. I store data for such tables in 100 byte records, using the first 15-20 columns of a record for the name of the item tabulated, and 60-65 of the remaining columns to store each State's values. For some tables I have to use two such records to store the data for all 16 States, and sometimes I have to use multiple pairs of records to accommodate items with very long names. I use the remaining 20 columns of each record for sort sequence numbers and for line numbers; I maintain sort sequence numbers in columns 85-91 and line numbers in columns 92-100.

Summarizing the Data

Data summarization for this document was limited to calculating totals and averages (means) of data. To average variables across columns I began by RETAINing sum and counter variables with an initial value of 0, then for non-missing values, I incremented each of them "retained" variables. I used the footnote section of the program to compute and print the averages at the bottom of the table.

To compute an average across a row, I formed sums of variables for each observation. Source statements shown in Figure 5 furnish an example. Here the columns are the 16 States, and I used an ARRAY statement and a DO group to handle the summation. Figure 5 also illustrates usage of the ISDAVID and NOTDAVID macros. Often the output format required for a data item depends on the item's value, so I used several PUT statements with "IF" conditions to write these values.
The printing of within-State totals in a few tables is another problem. An example is illustrated in Figure 6 (Table 22 of the document). Some State agencies provide us with their own totals, but some may not. When totals were provided, I stored them in the dataset, but I decided to print computed State totals instead. For these tables I had to calculate row and column statistics simultaneously. Figure 6 illustrates an interesting problem because of its missing cells. These occur because different agencies do not have exactly the same line items in their budgets, though they do have the same items of expense; State values that are missing in one row have been included as part of the values on another row, so that many row averages pertain to fewer than 16 States. Because of missing cells, the average of the "TOTAL" row is not equal to the total of the "AVERAGE" column. The appropriate value to place at the intersection of the "TOTAL" row and the "AVERAGE" column is the average of the "TOTAL" row, so the source program is written to use this value.

Acknowledgements

Preparation of the operational statistics report was supported by the Southeastern Association of Fish and Wildlife Agencies, who will publish the report with the title "Vital Statistics Southeastern Association of Fish and Wildlife Agencies." I thank Dr. Don W. Hayne and Mr. Gary Meyers for their guidance in its preparation.

Figure 5. Source statements for averaging data across a row and selecting output formats

```
DATA NULL;
INFILE CABLE5A  LINES=19 END=9;
INPUT #1 @1 LICTYPE $CHAR20.
#1 @2 (ST1-ST8) (7.2 9.2)
#2 @2 (ST9-ST16) (7.2 9.2);
ARRAY ST(I) ST1-ST16;
TOTAL=0; COUNT=0;
MACRO X ST %
 DO I=1 TO 16;
    IF NOTDAVID THEN DO;
        TOTAL = TOTAL + ST;
        COUNT = COUNT + 1;
    END;
    IF TOTAL ^= 0 THEN
        AVRGE = TOTAL / COUNT;
    END;
FILE TEMP1;
PUT #1 LICTYPE $CHAR20.
MACRO X ST % IF NOTDAVID THEN PUT #23 ST1 DOLLAR10.2 6;
    ELSE IF ISDAVID THEN PUT #27 ST1 DAVIDS. 6;
MACRO X ST % IF NOTDAVID THEN PUT #23 ST2 DOLLAR10.2 6;
    ELSE IF ISDAVID THEN PUT #27 ST2 DAVIDS. 6;
MACRO X ST % IF NOTDAVID THEN PUT #23 ST9 DOLLAR10.2 6;
    ELSE IF ISDAVID THEN PUT #27 ST9 DAVIDS. 6;
    PUT;
FILE TEMP2;
PUT #1 LICTYPE $CHAR20.
MACRO X ST10 % IF NOTDAVID THEN PUT #24 ST10 DOLLAR10.2 8;
    ELSE IF ISDAVID THEN PUT #28 ST10 DAVIDS. 8;
MACRO X AVRGE % IF AVRGE ^= 0 THEN PUT #112 AVRGE DOLLAR10.2 8;
    ELSE IF NOTDAVID THEN PUT #109 AVRGE DOLLAR10.2 8;
    ELSE IF ISDAVID THEN PUT #112 AVRGE DAVIDS. 8;
```

347
Figure 6. A Table With Row Averages and Column Totals

**PRELIMINARY DRAFT. ALL DATA UPDATED EXCEPT FOR MARYLAND**

<table>
<thead>
<tr>
<th>STATE</th>
<th>NORTH CAROLINA</th>
<th>OKLAHOMA</th>
<th>SOUTH CAROLINA</th>
<th>TENNESSEE</th>
<th>VIRGINIA</th>
<th>WEST VIRGINIA</th>
<th>AVERAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL EXPENDITURES</td>
<td>51,112</td>
<td>29,018</td>
<td>51,132</td>
<td>51,118</td>
<td>51,112</td>
<td>51,118</td>
<td>51,118</td>
</tr>
<tr>
<td>PERSONAL SERVICES</td>
<td>6,172</td>
<td>4,800</td>
<td>6,192</td>
<td>6,192</td>
<td>6,172</td>
<td>6,192</td>
<td>6,192</td>
</tr>
<tr>
<td>CIVILIAN EMPLOYMENT</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
</tr>
<tr>
<td>COMMUNITY DEVELOPMENT AND HOUSING</td>
<td>999</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
<td>56,364</td>
</tr>
<tr>
<td>TRAVEL AND VEHICLE CREATION</td>
<td>5,260</td>
<td>6,750</td>
<td>5,260</td>
<td>5,260</td>
<td>5,260</td>
<td>5,260</td>
<td>5,260</td>
</tr>
<tr>
<td>PROFESSIONAL SERVICE AND DUES</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
</tr>
<tr>
<td>BONDS FOR LOANS</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
<td>2,250</td>
</tr>
<tr>
<td>LAND ACQUISITION</td>
<td>6,172</td>
<td>6,172</td>
<td>6,172</td>
<td>6,172</td>
<td>6,172</td>
<td>6,172</td>
<td>6,172</td>
</tr>
<tr>
<td>LOCAL BARS</td>
<td>516,760</td>
<td>516,760</td>
<td>516,760</td>
<td>516,760</td>
<td>516,760</td>
<td>516,760</td>
<td>516,760</td>
</tr>
<tr>
<td>CONSTRUCTION</td>
<td>514,418</td>
<td>514,418</td>
<td>514,418</td>
<td>514,418</td>
<td>514,418</td>
<td>514,418</td>
<td>514,418</td>
</tr>
<tr>
<td>INCOME FROM OTHER SOURCES</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
<td>575</td>
</tr>
<tr>
<td>TRANSFERS</td>
<td>5,260</td>
<td>6,750</td>
<td>5,260</td>
<td>5,260</td>
<td>5,260</td>
<td>5,260</td>
<td>5,260</td>
</tr>
<tr>
<td>TOTAL</td>
<td>62,918</td>
<td>62,918</td>
<td>62,918</td>
<td>62,918</td>
<td>62,918</td>
<td>62,918</td>
<td>62,918</td>
</tr>
</tbody>
</table>

**TABLE 195 (CONTINUED). 1980-91 SOUTHEASTERN REGION AGENCY BUDGETS GROWN DOWN BY LINE ITEMS (IN THOUSANDS OF DOLLARS)**