HTML Table Report Using the SAS DATA STEP

Cyril Ravindra, U.S. Bureau of the Census, Washington, DC

ABSTRACT

Web publishing has quickly become an important delivery media for information. SAS Institute provides various macros for web publishing. These macros create reports similar to the reports produced by SAS output procedures. While these reports are still powerful in presenting complex information, some times they have their own limitations and disadvantages. When it comes to creating customized reports, the SAS data null step programming is the best solution. With some basic knowledge of HTML tags and the SAS data null steps, one can format a regular SAS output into a HTML document, that can be viewed in a web browser (see Figure 1 at the end of this paper).

INTRODUCTION

In the American Community Survey (ACS) at the U.S. Census Bureau, we conduct various survey operations daily. Some of them include keying the mail returns, telephone and person interviews and telephone follow-up. These activities are updated in a control file daily. Each morning we create a report called 'ACS Status Report' to view the statistics at various survey sites. This report creates tables by panel months and geographic summary levels.

The ACS report program contains two data steps. In the first step , we create a temporary data set from the control file that contains various tally counts, which are stored into array cells. Tallies are counted using Boolean logic. The second step uses above created data set, and, using put statements and HTML tags, creates a HTML formatted document.

STEP 1

```
libname   statements;
proc format statements;

/* CREATE DATE-TIME STAMP FOR REPORT HEADER */
    data _null_ ;
         call
symput('cdate',put(date(),mmddyy10.)||" "||
              put(time(),time8.));
    run;

/* SORT DATA BY CLASS VARIABLES */
    proc sort data = ctl.ctrlf1 out=acsdata;
         by descending panel ro st cn ;
    run;

/* CREATE THE SUMMARIZED DATA */
/* w1 - w46 are column variables in the tables
*/
    data acsdata1( keep = slevel panel st stname
cn ctyname ro w1 - w46);
         set acsdata;
         by descending panel ro st cn ;

/* 2-d array to hold tallies by summary levels
*/
         array t{4,46};
/* Initialize the array cells */
/* STATE TALLIES */
         if first.st then do;
              do j= 1 to 46;
                   t{1,j} = 0;
              end;
         end;
/* REGIONAL OFFICE (RO) TALLIES */
         if first.ro then do;
              do j= 1 to 46;
                   t{2,j} = 0;
              end;
         end;
/* STATE, RO AND COUNTY TALLIES */
         if (first.st or first.ro or first.cn)
         then do;
              do j= 1 to 46;
                   t{3,j} = 0;
              end;
         end;
/* BREAK EACH SET OF REPORTS CONTAINING THREE
GEOG. LEVELS
   BY PANEL MONTH */

         if first.panel then do;
              do j= 1 to 46;
                   t{4,j} = 0;
              end;
         end;
/* INCREMENT THE TALLIES USING BOOLEAN LOGIC  */
         do i = 1 to 4;
           t{i,1}  + (m1_st = '01')  ;
           t{i,2}  + (m1_str = '10') and (m1_str
           >= m2_str);
           t{i,3}  + (m1_st >= '50' or m2_st >=
           '50');
           ..
           ..
         end;
         drop i j;
 /* THIS MACRO WILL TRANSFER 2 DIMENSIONAL ARRAY
   CELLS INTO 1 DIMENSION COLUMN VARIABLES */

         %macro mlevel( sval,sl);
            slevel = &sval;
            w1  = t{&sl,1};
            w2  = t{&sl,2};
            w3  = t{&sl,3};
           ..
           ..
            output;
         %mend mlevel;

  /* INVOKE THE MACRO TO OUTPUT THE INDIVIDUAL
TALLY TOTALS AS COLUMN VARIABLES IN THE TABLE */
         if last.st then do;
            %mlevel(1,1)
```

```
        end;

        if last.ro then do;
           %mlevel(2,2)
        end;

        if (last.st or last.ro or last.cn) then
        do;
           %mlevel(3,3)
        end;

      /* GRAND TOTAL OUTPUT, NUMBERED SUCH THAT
WHEN SORTED, SUB TOTALS FALL UNDERNEATH THE
DETAIL DATA FOR EACH OF THE GEOG. LEVELS AND
BREAKS BY PANEL */

        if last.panel then do;
           %mlevel(1.1,4)
           %mlevel(2.1,4)
           %mlevel(3.1,4)
        end;
    run;

    proc sort data = acsdata1;
         by descending panel slevel;
    run;
```

STEP 2

In the following example only for one table of code is shown to save the space.

```
data _null_;

        set acsdata1;
        by descending panel slevel;

    file repfill print notitle lrecl = 175;
         if _n_ = 1 then do;
           pgcount=0;
    /* REPORT HEADING */
           put  @1 '<HTML>' /
                @1 '<HEAD> ACS STAT
                REPORT</HEAD>' /
                @1 '<BODY bgcolor=pink>'     /
                @1 '<PRE><H4>'/
                @b8      'U.S Department of
                Commerce' /
                @(b8+6) 'Bureau of the Census' /
                @(b8+1) 'American Community
                Survey' /
                @(b8+7)"&cdate" /
                @1 '</H4></PRE>';
        end;

    /* PAGE COUNT */
         if first.slevel  then do;
              if slevel in (1,2,3) then do;
                   pgcount+1;
              end;
    /* PAGE SUBHEADERS AND COLUMN TITLES BY
        LEVEL */
     if slevel = 3 then do;
       put @1 '<PRE>' /
          / @1 'SAMPLE MONTH:'  @15 panel  @138
              'Page:' @144 pgcount /
            / @1 'MAIL-CATI-CAPI' //
              @1 '</PRE>'
              @1 '<TABLE BORDER=1 WIDTH=100%
                  ALIGN=CENTER  CELLPADDING=1
                    CELLSPACING=1>' /
              @1 '<TR>' /
                    @1 '<TH ALIGN=CENTER
                    VALIGN=MIDDLE ROWSPAN
                    ="2">STATE </TH>'/
              @1 '<TH ALIGN=CENTER VALIGN=MIDDLE
                  ROWSPAN="2"> COUNTY </TH>'/
              @1 '<TH ALIGN=CENTER VALIGN=MIDDLE
                    ROWSPAN="2"> RO </TH>'/
              @1 '<TH ALIGN=CENTER VALIGN=MIDDLE
                    COLSPAN="4"> MAIL </TH>'/
              ..
              ..

              @1 '</TR>' /
                 ;
              end;
        /* ADDITIONAL SUBSETTED COLUMN TITLES BY
           LEVEL */
           if slevel in (1,2,3) then put
                       @1 '<TR>' /
                       @1 '<TH ALIGN=CENTER
                                VALIGN=MIDDLE >
                                UNMAILABLE </TH>'/
                        @1 '<TH ALIGN=CENTER
                                VALIGN=MIDDLE >
                                MAILED   </TH>'/
                       ..
                       ;
            end;

           /* WRITE THE DATA VALUES FOR GEOGRAPHIC
VARIABLES */

           if slevel = 3 then do;
                put    @1 '<TR>' /
                       @1 '<TD ALIGN=CENTER
                                VALIGN=MIDDLE
                                >' stname
                                '</TD>'/
                       @1 '<TD ALIGN=CENTER
                              VALIGN=MIDDLE
                              >' ctyname
                              '</TD>'/
                       ..
                       @;
           end;

/*WRITE THE TALLY DATAS INTO THE TABLE*/
           if slevel in (1,2,3) then do;
           put @1 '<TD ALIGN=CENTER VALIGN=MIDDLE
                 >' w1 '</TD>' /
               @1 '<TD ALIGN=CENTER VALIGN=MIDDLE
                 >' w2 '</TD>' /
               ..
               ..
               @1 '<TD ALIGN=CENTER VALIGN=MIDDLE
                  >' w10 '</TD>' /
               @1 '</TR>'
           ;
          end;
         /* WRITE THE SUBTOTALS INTO THE TABLE */
           if slevel in (1.1,2.1,3.1) then do;
           put @1 '<TD ALIGN=CENTER VALIGN=MIDDLE
                  >' w1 '</TD>' /
               @1 '<TD ALIGN=CENTER VALIGN=MIDDLE
                  >' w2 '</TD>' /
               ..
               ..
               @1 '<TD ALIGN=CENTER VALIGN=MIDDLE
                  >' w10 '</TD>' /
               @1 '</TR>'
               @1 '</TABLE>'
           ;
          end;
      run;
```

Below is a sample screen capture of the HTML report produced using this code (Figure 1).

CONCLUSION

With the little knowledge of HTML TAGS and data null step programming we can create HTML tables in a more efficient and flexible way.

ACKNOWLEDGMENTS

SAS is a registered trademark or trademark of SAS Institute, Inc. in the USA and other countries. Other brand and product names are registered trademarks or trademarks of their respective companies.

The author would like to thank Amanda Sardenberg and Bula Ghose for their input and assistance.

REFERENCES

O'Reilly and Associates, Inc., Internet in a Nutshell, 1997.

SAS Institute, Inc., SAS Report Writing: A Programming Approach, 1996.

CONTACT INFORMATION

Mr. Ravindra can be contacted by mail at the Continuous Measurement Office, Demographic Statistical Methods Division, Bureau of the Census, Washington DC 20233, or by telephone at (301) 457-8052. His e-mail address is cyril.ravindra@census.gov.
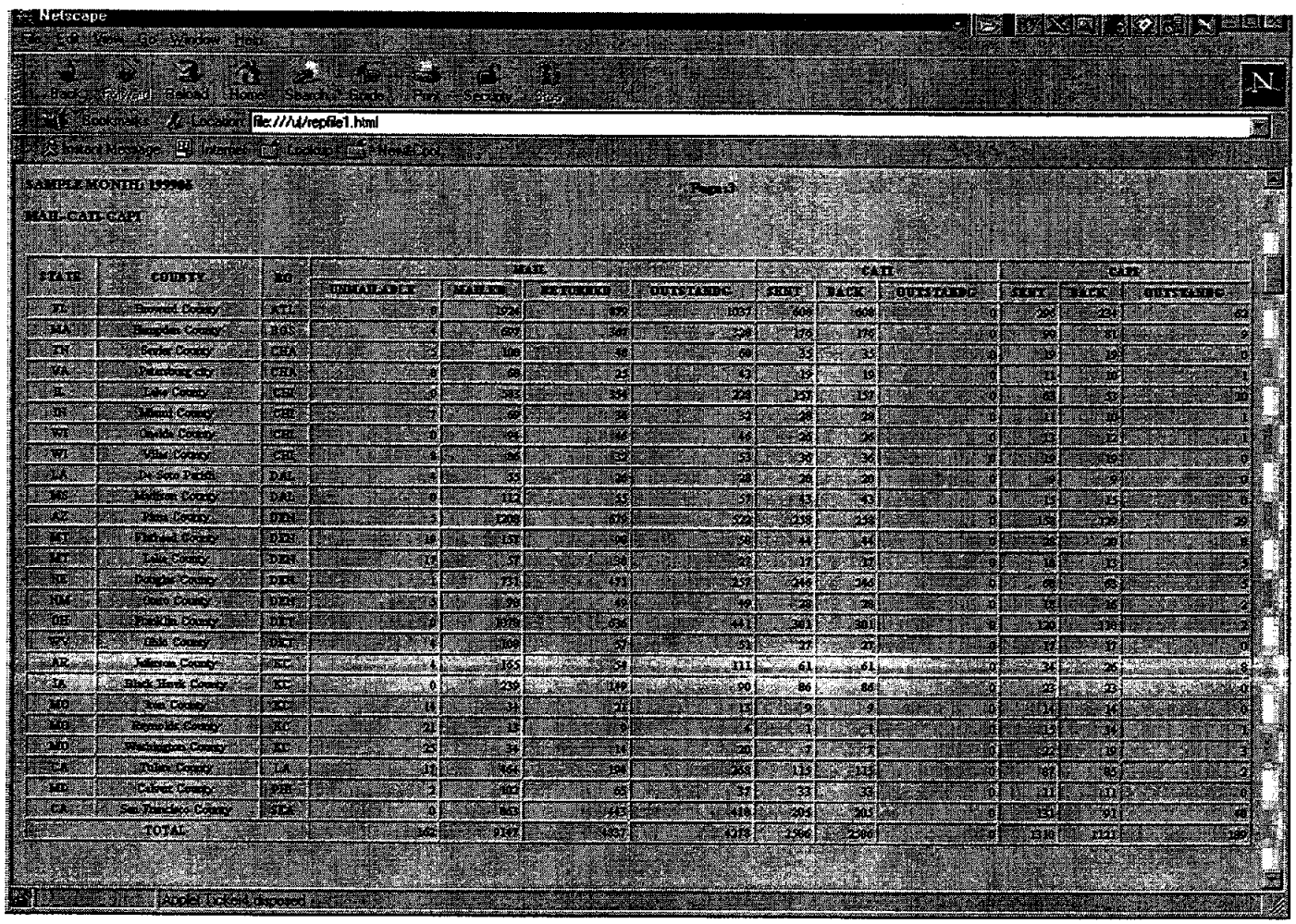
Figure 1.