

Paper 225-2009

Automatic Generation of Titles, Footnotes, and Superscripts for Tables with Macros and PROC REPORT

Paul Cascagnette, Health Quality Council, Saskatoon, Canada

ABSTRACT

What do you do when you have 5,000 web ready output tables to produce in a short time? You panic! – Then you automate. We created a macro and input structure using SAS[®] ODS, PROC REPORT and Microsoft Excel that auto-generated the tables, web pages, directories, and HTML index pages. The footnotes and subscripts were all put in the right place, and with the right number. The macro can be easily modified for the needs of the user. Researchers, and persons with possibly little or no background in SAS can enter their metadata specifications for the output tables into the Excel template, and the worksheet can be subsequently modified without the need to make further changes to the macro.

INTRODUCTION

We developed a macro and metadata structure to handle different title, footnotes, and superscripts requirements for similar table layouts. Metadata for the output tables was entered into an Excel spreadsheet and imported into SAS with PROC IMPORT. Tables were generated from the macro using PROC REPORT and ODS HTML.

The macro and inputs have the following features, and can be easily adapted to suit the needs of the user:

- Allows superscripts to appear after titles and row or column headers.
- Automatically determines numbering for superscripts
- Allows for multiple superscripts to appear at the same position
- Generates the same superscript number when the same footnote is referred to at different positions within the table
- Allows a user with no knowledge of SAS to enter footnote and superscript requirements into an Excel document
- Easy specification of superscript position and footnote using a letter/number combination (e.g. 'a24')

The macro also demonstrates these additional features:

- Outputs are generated by major topic area and separate folders for the outputs are automatically created
- Tables are created in HTML using SAS ODS, PROC REPORT and PROC FORMAT
- HTML index pages with titles and hyperlinks to individual web pages containing the various tables are automatically created for each major topic area
- Traffic-lighting highlights particular cells within a column

The original macro has been simplified for the purpose of generalization.

INPUT AND OUTPUT STRUCTURES

Table 1 is typical of the output tables that we produced in HTML format for our health quality outcome and process indicators. PROC REPORT was used to create this table from a master analysis dataset. In this example there were eight footnotes with superscripts in the appropriate places. Values in one column have altered colors and fonts to make them stand out to the reader (i.e. traffic-lighting). Green and bold represent values that are statistically significantly better than the other categories combined within the grouping (e.g. age). Red and italics represent values that are worse. Values in normal font are not deemed significantly better or worse.

Table 1: Sample HTML output table with footnotes and superscripts

Percentage of people with asthma aged 6-55 years in Saskatchewan with poor asthma control, by demographic categories, FY2005/06^{1 2 3}

	Numerator (#)	Denominator (#)	Actual rate (%)	Actual rate 95% CI	Risk adjusted rate (%) ^{4 5 6}	Risk adjusted rate 95% CI
All SK	4,383	24,622	17.8	17.3 - 18.3	17.8	17.3 - 18.3
Income quintile⁷						
1 Lowest	961	4,368	22.0	20.8 - 23.3	<i>21.3</i>	20.1 - 22.5
2	1,008	5,300	19.0	18.0 - 20.1	<i>18.8</i>	17.7 - 19.8
3	809	4,836	16.7	15.7 - 17.8	16.7	15.7 - 17.8
4	755	4,454	17.0	15.9 - 18.1	17.0	15.9 - 18.1
5 Highest	813	5,429	15.0	14.0 - 16.0	15.5	14.5 - 16.5
Sex						
Female	2,006	12,331	16.3	15.6 - 16.9	15.4	14.8 - 16.0
Male	2,377	12,291	19.3	18.6 - 20.0	<i>20.8</i>	20.1 - 21.6
Location⁸						
Urban	3,000	16,888	17.8	17.2 - 18.3	17.7	17.2 - 18.3
Non-urban	1,372	7,687	17.8	17.0 - 18.7	18.3	17.4 - 19.2
Age						
06-09	167	3,110	5.4	4.6 - 6.2	5.3	4.5 - 6.1
10-14	323	3,601	9.0	8.1 - 10.0	8.9	8.0 - 9.8
15-19	385	3,022	12.7	11.6 - 14.0	12.8	11.6 - 14.0
20-29	857	4,117	20.8	19.6 - 22.1	<i>21.1</i>	19.9 - 22.4
30-44	1,433	5,907	24.3	23.2 - 25.4	<i>24.7</i>	23.6 - 25.9
45-55	1,218	4,865	25.0	23.8 - 26.3	<i>25.8</i>	24.6 - 27.1

¹ Fiscal year begins April 1 and ends March 31.

² People aged 5–54 years in the case definition year (the year prior to the measurement year) were aged 6-55 years in the measurement year.

³ Poor asthma control was defined as high SABA use (> or = 4 SABA inhalers dispensed in 1 year), admission to hospital because of asthma or death caused by asthma in the measurement year.

⁴ Age specific results adjusted for sex, sex specific results adjusted for age, other results adjusted for both age and sex, using the indicator denominator population.

⁵ Bold and green text indicates results that are statistically significantly better than the result across all other categories within the classification variable.

⁶ Italics and red text indicates results that are statistically significantly worse than the result across all other categories within the classification variable.

⁷ Income quintiles are based on 2001 Census average income adjusted for household size.

⁸ Postal codes of individuals or institutions falling within a census agglomeration (CA) or census metropolitan area (CMA) were classified as urban. The remainder were classified as non-urban.

The Excel metadata file, *titles and footnotes.xls*, consists of at least two sheets. The first sheet (see *table 2*) is named *database*, specifying major topic area (study), topic (indicator), titles, footnotes, hyperlink text (link), and superscript information for the various tables. We had five major topic areas as part of a single report that used the same table layouts. The second sheet is named *footnotes*, (see *table 3*) containing actual text for the footnotes (description) along with the footnote number (fn_number). For convenience of those filling out the templates, we also included a *legend* sheet (see *table 4*), which describes the location (title, column heading, or category heading) where the footnotes can go, and the letter code associated with the location.

The metadata file is imported into SAS using PROC IMPORT; however one could also create the metadata file directly in SAS. In our case the Researchers, some of whom did not regularly use SAS, preferred to use Excel to input the data.

Table 2: Portion of DATABASE sheet in Excel metadata file

STUDY	INDICATOR	TITLE	LINK	TABLE1	TABLE2
LTC	Chronic Use Benzos	Percentage of long-term care residents in Saskatchewan, aged 65 years and older, who were chronically dispensed benzodiazepines	Percentage of long-term care residents chronically dispensed benzodiazepines	a1,a2,f3,f5,f6,j10	a1,a2,f4,f7,f8,h9,i11
LTC	Above max for Chronic Use Benzos	Percentage of long-term care residents in Saskatchewan, aged 65 years and older, who were chronically dispensed benzodiazepines at average daily doses greater than the recommended maximum level	Percentage of long-term care residents chronically dispensed benzodiazepines above maximum recommended dose	a1,a2,f3,f5,f6,j10	a1,a2,f4,f7,f8,h9,i11

Table 3: Portion of FOOTNOTES sheet in Excel metadata file

FN_NUMBER	DESCRIPTION
1	Fiscal year begins April 1 and ends March 31.
2	Excludes Saskatchewan residents with Registered Indian status and persons with no prescription drug claims during the study period.
3	Result adjusted for both age and sex, using the indicator denominator population.
4	Age specific results adjusted for sex, sex specific results adjusted for age, other results adjusted for both age and sex, using the indicator denominator population.

Table 4: Portion of LEGEND sheet in Excel metadata file specifying superscript location

Positions b to g are column headers and positions h to m are row (category) headers.

PLACEMENT	DESCRIPTION
a	Title
b	Numerator
c	Denominator
d	Actual rate
e	Actual rate 95% CI
f	Risk adjusted rate
g	Risk adjusted rate 95% CI

h	Income quintile
i	RI Status
j	RHA
k	Sex
l	Location
m	Age

ENTERING DATA IN THE METADATA FILE

The positions we chose to correspond to the letters of the alphabet follow a logical spatial ordering. Title is first, followed by the column headings (left to right), then category headings appearing on the left side of the table (top to bottom). Each table would include some subset of the columns and categories that were available in the *legend* sheet.

The *table1*, *table2*, etc. columns of the *database* sheet allow users to enter codes corresponding to the position and footnote number of each footnote in each table e.g. 'a5' in *table1* means that footnote 5 will appear in the footnotes, and a superscript for that footnote will appear after position 'a' (the title), as well as in front of footnote 5 following table 1. The footnotes will be numbered in the order specified. The user should specify these in alphabetic order, with each letter/number combination separated by a comma.

In the *footnotes* sheet, we placed the most commonly used footnotes first. The columns *study* and *indicator* are the key variables to link to our master dataset, which contains all the data to appear in the tables. The *link* column is what the text on the HTML hyperlink appearing on the HTML index page for that major topic area should read.

MACRO FOR AUTOMATIC TABLE GENERATION

Here I will walk you through the macro `%web_rpt`. There are parts of the macro that will have to be modified by the user, including:

- Paths for input and output files
- Variables and formats for rows and columns to be displayed in the output tables, corresponding to the information found in the user's master analysis file and desired output table structure

In the first part of the macro, information in the Excel metadata sheets are imported into SAS using PROC IMPORT and merged with the SAS master data set using PROC SQL.

```
%macro web_rpt(ds_in=,metadata_locn=);

/*****
* PARAMETERS
* ds_in          Master dataset used for output table generation
*               PROC REPORT will use this data
* metadata_locn  Location of Excel metadata file e.g. C:\my_SASdata
*
* CONTAINS THE FOLLOWING MACRO:
* %byind(study=,ind=) Process tables by study (major topic area) and
*   indicator(topic)
*
*****/

** MODIFY this value - Number of different types of tables to be produced;
%let ntables=2;

* Import the metadata from the Excel file;

* First the database sheet;
PROC IMPORT OUT=Footnote_map
  DATAFILE= "&metadata_locn\titles and footnotes.xls"
```

```

        DBMS=EXCEL REPLACE;
        SHEET="database$";
        GETNAMES=YES; MIXED=NO; SCANTEXT=YES; USEDATE=YES; SCANTIME=YES;
RUN;

* Next the footnotes sheet;
PROC IMPORT OUT= footnotes
  DATAFILE= "&metadata_locn\titles and footnotes.xls"
  DBMS=EXCEL REPLACE;
  informat description $1024.; format description $1024.;
  SHEET="footnotes$";
  GETNAMES=YES; MIXED=NO; SCANTEXT=NO; USEDATE=YES; SCANTIME=YES;
RUN;

* Put the footnotes into macro variables;
DATA _null_;
  set footnotes;
  call symputx('fnote' || trim(left(put(fn_number,best3.))),description);
  call symputx('nfoot',trim(left(put(_n_,best3.))));
run;

* Merge master data file with footnote mapping file;
* Merging is based upon EXACT match of the study and indicator values (not case specific);
* Any non-matches are discarded;

PROC SQL noprint;
  create table data_in as
  select a.*, b.indicator,b.title,
  %do i=1 %to &ntables;
    b.table&i,
  %end;
  b.link
  from &ds_in a left join footnote_map b
  on upcase(a.study)=upcase(b.study) and upcase(a.indicator)=upcase(b.indicator)
  where a.study ne "" and a.indicator ne "" and
        b.study ne "" and b.indicator ne "";
quit;

```

Next, output directories and HTML index pages are created. There will be one directory created for each study or major topic area, which will contain all the HTML output table pages for the various topics within the study and one HTML index page with hyperlinks to each topic. We uploaded the entire directory structure to our web server. A main web page was additionally created by our web manager to showcase our report, with links to each of the auto-generated web index pages produced by the macro. Modify the path and directory names to suit your web environment.

```

* This creates a macro var list of studies and indicators including a web link;
* Cmpvar is the comparison var or category (e.g. Sex) and cmpval is the value of cmpvar (e.g. Male);

PROC SORT data=data_in out = inds NODUPKEY;
  by study indicator cmpvar cmpval;
run;

DATA _null_;
  set inds;
  by study indicator;
  retain scnt icnt;
  if _n_=1 then scnt=0;
  if first.study then do;
    icnt=0;
    scnt+1;
    call symputx('study' || trim(left(put(scnt,best2.0))),study);
    call symputx('nstudy',trim(left(put(scnt,best2.0))));* no. of studies;
  end;
  ** Indi_j is indicator (topic) j in study (major topic area) i;
  if first.indicator then do;
    icnt+1;
    * Number of indicators;
    call symputx('nind' || trim(left(put(scnt,best2.0))),put(icnt,best3.0));
    * Indicator name;
    Call symputx('ind' || trim(left(put(scnt,best2.0))) || '_' || trim(left(put(icnt,best2.0))),
    indicator);
  end;

```

```

* HTML hyperlink to indicator (change to your web output location);
indref="http://www.hqc.sk.ca/hqcQInsight/"||trim(left(study))|| "/" ||
trim(left(indicator))|| "/" ||trim(left(indicator))|| ".html";
call symputx('indref' ||trim(left(put(scnt,best2.0)))|| '_' ||
trim(left(put(icnt,best2.0))),indref);
* What the link should say;
call symputx('link' ||trim(left(put(scnt,best2.0)))|| '_' ||trim(left(put(icnt,best2.0)))
,link);
end;
run;

* Create output directory, if it does not already exist;
* MODIFY directory and path names;
DATA _null_;
  dirname="hqcQInsight";
  newdir=dcreate(dirname,"P:\"); *P:\ is parent for new dir;
run;

%do i=1 %to &nstudy; * Number of studies;
  * First for study (major topic area);
  DATA _null_;
    dirname="&&study&i";
    newdir=dcreate(dirname,"P:\hqcQInsight\"); * Parent dir for studies;
  run;
  * Now for indicator (topic) within study;
  %do j=1 %to &&nind&i; * Number of indicators for study;
    DATA _null_;
      dirname="&&ind&i._&j";
      newdir=dcreate(dirname,"P:\hqcQInsight\&&study&i");* Parent dir for topic;
    run;
  %end;
%end;

* Create formats for outputting HTML code with links - DO NOT MODIFY;
%do i=1 %to &nstudy;
  PROC FORMAT;
    value $r&i.tag
      %do j=1 %to &&nind&i;
        "&&ind&i._&j"="<a href=" "&&indref&i._&j" ">&&link&i._&j</a>"
      %end;
  ;
run;
%end;

* Generate HTML index pages for each study area;
* There will be links to the study topics (indicators) on each page;
footnote; title; *Reset footnotes and titles for web pages;

PROC SORT data=inds out=inds_nodup (keep=study indicator) NODUPKEY;
  by study indicator;
run;

* MODIFY titles for your studies (major topic areas) - 1 web page per study;
* ODS and PROC PRINT are used to create the web pages;
%do i=1 %to &nstudy;
  %if "&&study&i"="Asthma" %then %let stdy=Asthma;
  %else %if "&&study&i"="Com_Drugs" %then
  %let stdy=Drug management in community-dwelling seniors;
  %else %if "&&study&i"="Diabetes" %then %let stdy=Diabetes;
  %else %if "&&study&i"="LTC" %then %let stdy=Drug management in long term care;
  %else %if "&&study&i"="Post AMI" %then %let stdy=Post heart attack care;

  DATA tmpst;
    set inds_nodup(where=(study="&&study&i"));
  run;
  ODS listing close;
  ODS HTML file="P:\hqcQInsight\&&study&i\&&study&i...html"; *MODIFY path;
  PROC PRINT data=tmpst noobs l;
    label indicator="Indicators for &stdy";
    format indicator $r&i.tag.;
    var indicator;
  run;
  ODS HTML close;
  ODS listing;
%end;

```

Nested into the macro %web_rpt is another macro %by_ind which will process the outputs by study (major topic area) and indicator (topic). Macro variables created above will make it easy to loop through all of them.

First, the desired sort order is pre-determined by the user and data is sorted according to the row order of what the user would like to see for categories in the output tables. PROC REPORT will use this internal sort order when creating the tables. Then, the information from the Excel metadata sheets is put into macro variables. This includes the superscript locations and footnotes in each of the tables for each major topic area and topic.

```

%macro byind(study=,ind=);
** Process tables by study (major topic area) and indicator (topic);

* Determine sort order for display in table rows since PROC REPORT will use this internal
  order of the sorted file;
* MODIFY sort order for the table row categories you have in your data;
DATA bydata;
  set data_in(where=(upcase(study)=upcase("&study") and upcase(indicator)=upcase("&ind")));
  call symputx('title',title);
  if cmpvar='YEAR' then sort_order=1;
  else if cmpvar='incqtl' then sort_order=2;
  else if cmpvar='non_ri' then sort_order=3;
  else if cmpvar='rha' then sort_order=4; * Used in other table not shown;
  else if cmpvar='sex' then sort_order=5;
  else if cmpvar='urb_rur' then sort_order=6;
  else if cmpvar='age' then sort_order=7;
  else if cmpvar='hospcat' then sort_order=8;
  else if cmpvar='hospital' then sort_order=9;
run;

* Here we put the information from the Excel tables into macro variables for later use;
* NOTE: This is generic code--only MODIFY number of possible table entries (index for j loop);
DATA _null_;
  set inds(where=(upcase(study)=upcase("&study") and
    upcase(indicator)=upcase("&ind")));
  * Tentries_i = # of entries in table i;
  * Tpos_i_j = position of entry j in table i;
  * Tfoot_i_j = footnote for entry j in table i;

  ** Get number of footnotes for tables and put entries into macro variables;
  %do i=1 %to &ntables; *Loop for each table;
  call symputx("tentries_&i",'0');
  do j=1 to 30; * Scan entries - up to 30 footnotes per table (can MODIFY number);
  format jpos $2.;
  jpos=trim(left(put(j,best3.)));
  if SCAN(table&i,j,',') ne '' then do;
    call symputx("tpos_&i._"||trim(left(jpos)),substr(SCAN(table&i,j,','),1,1));
    call symputx("tfoot_&i._"||trim(left(jpos)),substr(SCAN(table&i,j,','),2,3));
    call symputx("tentries_&i",jpos);
  end;
  end;
  %end;
run;

*** NOTE: ONLY ONE TABLE (TABLE 2) IS SHOWN IN THIS CODE ;
** Table 2 - Indicator by demographic categories for most recent Fiscal Yr;

* Keep only the vars for row groupings you want to display in the output table - MODIFY;
DATA table2;
  set bydata(where=(cmpvar in ("incqtl" "non_ri" "sex" "urb_rur" "age" "hospcat"
    "hospital")));
run;

```

This is where the real magic happens. The work done previously in filling in a standardized metadata file and the process of creating macro variables (above) allows us to generate the superscripts and footnotes in a relatively small amount of code. Though difficult to read, you won't need to change anything. For those who want to understand exactly what is going on, I have included extensive comments in the code to explain. The code is currently limited to 26 table positions (corresponding to the letters a through z), but more symbols (e.g. digits 0 through 9) could be

added if more positions are needed.

```

* Put footnotes and superscripts into macro variables;
%let alpha=a b c d e f g h i j k l m n o p q r s t u v w x y z;
* Initialize table superscript variables;
%do i=1 %to &ntables; * tables ;
  %do k=1 %to 26; * letters of alphabet;
    %let tmp=%scan(&alpha,&k,' ');
    %let t&tmp.fmt_&i=;
  %end;
%end;

** Calculate superscripts based on positions in table;
** Allow for multiple footnotes on the same position;
%do i=1 %to &ntables; %*** TABLES ***;
  %let tfnnum&i=0; * Initialize footnote count for table i;
  %do j=1 %to &&tentries_&i; *Loop table entries in table i;
    * Check for the existence of the footnote in a previous position;
    %let prev_supscr=; * Num of assigned superscript where same footnote found;
    %do m=1 %to %eval(&j-1);
      %if &prev_supscr= and &&tfoot_&i._&j=&&tfoot_&i._&m %then %do;
        %let prev_supscr=&&tsupr&m;
        %let tsupr&j=&prev_supscr;
      %end;
    %end;

    %if &prev_supscr= %then %do; *Repeated footnote;
      %let tfnnum&i= %eval(&&tfnnum&i+1); *Incr footnote (superscript) counter;
      %let tsupr&j=&&tfnnum&i; *Assign new footnote # to current footnote;
      ** Footnote for position j in table i;
      %let txfoot_&&tfnnum&i._&i=~{super &&tfnnum&i} &&&fnote&&tfoot_&i._&j;
    %end;

    *Check for the letter code and assign format superscript;
    %do k=1 %to 26;
      * Assign a new superscript if the footnote does not already exist in the table,
      else assign the superscript previously assigned;
      %let tmp=%scan(&alpha,&k,' ');
      %if &&tpos_&i._&j=&tmp %then %do; * Position found;
        %if &prev_supscr ^= %then %do; * Repeated footnote;
          %let t&tmp.fmt_&i=&&t&tmp.fmt_&i._&prev_supscr%str( );
        %end;
        %else %let t&tmp.fmt_&i=&&t&tmp.fmt_&i._&&tfnnum&i%str( ); * Superscript for
        position alpha(k) in table i;
      %end;
    %end;
  %end;
  %if &&smnum&i=Y %then %let tfncnt&i=%eval(&&tfnnum&i+1);
  %else %let tfncnt&i=&&tfnnum&i;
%end;

*Here we add the ~super in front of the list of footnote superscripts for a given position in
the table - this tells SAS to make it a superscript when ~ is the ODS escapechar (see below);

%do i=1 %to &ntables; * Assign superscripts - TABLES;
  %do k=1 %to 26;
    %let tmp=%scan(&alpha,&k,' ');
    %if %length(&&t&tmp.fmt_&i) %then %do;
      %let t&tmp.fmt_&i=~{super &&t&tmp.fmt_&i} ; * Reference code for position alpha(k)
      in table i;
    %end;
  %end;
%end;

```

Here is where we assign the superscripts to the pre-determined table row locations using PROC FORMAT e.g. *&thfmt_2* is the superscript associated with position 'h' on table 2. Normally, one would assign these letters of the alphabet to correspond to row positions that move down the table. That way superscript numbers would increase as you move down the row headings. Do this in a similar way moving across the column headings. The title should correspond to the letter 'a' and the letters corresponding to the column headings should come before those corresponding to the row headings for the best aesthetics.

One function this macro does not do is to assign superscripts to individual cell values within the table, as this is very much data driven. To incorporate this feature, refer to (Tungala and Montgomery, 2006).

```

* Create formats that change cmpvar (var for row categories) to names acceptable for tables
and add footnote numbers;

* NOTE that &tXfmt_Y is the actual superscript code for predetermined position X (row header)
in table Y (e.g. ~{super 1}). Only table 2 is shown in this example;
PROC FORMAT;
  value $cmpvar
    "incqtl"="Income quintile&thfmt_2"
    "non_ri"="Registered Indian (RI)&tifmt_2"
    "sex"="Sex&tkfmt_2"
    "urb_rur"="Location&tlfmt_2"
    "age"="Age&tmfmt_2"
    "hospcat"="Hospital Category&tnfmt_2"
    "hospital"="Selected Hospitals&tofmt_2"
    "YEAR"=" "; *We will have a blank row heading for this var;
run;

* Use this format for all cmpvals (sub-categories);
* NOTE the original prefix values 01, 02, etc. help to sort the data in the order desired;
PROC FORMAT;
value $cmpval
  "01 Urban"="Urban"
  "02 Rural"="Non-urban"
  "01 Registered Indian"="Registered Indian"
  "02 All Other"="All Other Saskatchewan Residents"
  "01 Community and Northern"="Community and Northern"
  "02 District"="District"
  "03 Regional"="Regional"
  "04 Provincial"="Provincial"
;
run;

** Sort to get lines in the right order before running PROC REPORT;
** A format can then be applied (in PROC REPORT) to get the names appearing as desired;
%do i=1 %to &ntables;
  PROC SORT data=table&i;
    by time sort_order cmpvar cmpval;
  run;
%end;

```

Now, we call on PROC REPORT to generate our tables. We use ODS HTML to create our tables in HTML format. In PROC REPORT, we are not actually doing any summarizing in this example because that had been done earlier before we created our master data table. As was done earlier in PROC FORMAT for row locations, we assign the macro variable associated with the predetermined column location using the letter location codes e.g. *&tbfmt_2*.

Traffic lighting of the values in one column is accomplished through the use of the *compute* block. In our example, the format of the cell value is altered depending on the value of two other variables *signif_better* (significantly better) and *signif_worse* (significantly worse) found in our master data table.

```

* MODIFY path;
ODS listing close;
ODS HTML file="P:\hqcQInsight\&study\&ind\&ind.html" style=BarrettsBlue;
ODS escapechar = '~'; * Important for indicating superscript font format;

** Table 2 - Indicator (topic) by various classifications;

footnote; title; *Reset footnotes and titles;

* Here is where the report is created;
PROC REPORT data=table2 nowindows split='*'
  style(report)=[font=(Arial, 10pt)]
  style(column)=[font=(Arial, 10pt)]
  style(header)=[font=(Arial, 10pt) foreground=white font_weight=bold]
  style(summary)=[font=(Arial, 10pt)];

title f='Arial' h=12pt bold "&title, by demographic categories&tafmt_2";

```

```

column cmpvar cmpval observed n_denom crude_rate crude_ci ra_rate ra_ci
signif_better signif_worse;
define observed / sum "Numerator*(#)&tbfmt_2";
define n_denom / sum "Denominator*(#)&tcfmt_2";
define crude_rate / display "&tdfmt_2";
define crude_ci / display "&tefmt_2" style(column)={just=right};
define ra_rate / display "&tfmt_2";
define ra_ci / display "&tgfmt_2" style(column)={just=right};
* Show row vars in the order found in the dataset;
define cmpvar / group "" noprint order=data ;
define cmpval / display "" style(column)={just=right} format=$cmpval.;
define signif_better /display noprint;
define signif_worse /display noprint;

* Here we are traffic lighting the values: Green and bold for better and red and italics for
worse;
compute signif_better/char;
if signif_better = '*' then do;
  call define ("ra_rate", "style", "style=
    [font_weight=bold foreground=green font=(Arial, 12pt)]");
end;
endcomp;
compute signif_worse/char;
if signif_worse = '*' then do;
  call define ("ra_rate", "style", "style=
    [font_style=italic font_weight=medium foreground=red font=(garamond, 14pt)]");
end;
endcomp;
* Row headings for cmpvar (category);
compute before cmpvar / style={font_weight=bold font=(Arial, 10pt)};
line @1 cmpvar $cmpvar.;
endcomp;

/* Now generating the footnotes (with preceding superscripts) is easy
The macro var tfncntX is the number of footnotes for table X.
The macro var txfootI_J is footnote I in table J.
MODIFY for the number of the current table */
%do i=1 %to &&tfncnt2;
  footnote&i f='Arial' h=9pt "&&txfoot_&j._2";
%end;
run;
quit;

ODS HTML close;
ODS listing;

%mend byind;

```

Finally, the macro %web_rpt cycles through all the major topic areas and topics.

```

%global ix jx;
* Loop through each study (major topic area) and indicator (topic);
%do ix=1 %to &nstudy;
  %do jx=1 %to &&nind&ix;
    %byind(study=&&study&ix, ind=&&ind&ix._&jx);
  %end;
%end;

%mend web_rpt;

* Call the macro with your own parameter values: ;
%web_rpt(ds_in=your_master_data_file, metadata_locn=path_to_Excel_file);

```

CONCLUSION

The macro presented here demonstrates an efficient way to generate a large number of similar web ready tables with varying titles, footnotes and superscripts using PROC REPORT and Excel metadata files.

REFERENCES

Tungala, Rex and Montgomery, Rebecca Yost. "Dynamic Superscripts and Footnotes" *Proceedings of the Thirty-first Annual SAS® Users Group International Conference*. March 2006.

< <http://www2.sas.com/proceedings/sugi31/177-31.pdf> >

ACKNOWLEDGMENTS

Thanks to Recep Gezer for designing the output table code in PROC REPORT and for being involved in much of the development work.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Paul Cascagnette <pcascagnette@hqc.sk.ca>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.