Experimenting with the ODS DATA Step Object (Part I)

Richard Koopmann Jr. – Capella University – Minneapolis, MN

ABSTRACT

Recently, I was tasked with developing a SAS program to phase out tedious, error-prone process of transcribing a few key numbers from hundreds of item summary tables to a dozen one-page summary tables. To reduce end-user confusion, the revised reports needed to resemble the current layout and format as much as possible. Since the key statistics change across subgroups and span several item summary tables, a single tabulate or report procedure wouldn't be sufficient. It appeared that traditional ODS reporting was out.

Other reporting methods were explored, specifically `DATA _NULL_` reporting which allows the dynamic use of variables—via macro variables—and provides the layout control that would be needed. Due to the monospace appearance of these reports, however, `DATA _NULL_` reporting seemed to be out as well.

Fortunately, ODS DATA Step Object's ability to blend the layout control of `DATA _NULL_` reporting with the formatting control of traditional ODS would help save the day. with ODS DATA Step Object, SAS programmers can turn raw data into production quality reports ready for distribution. The ODS DATA Step Object is a powerful Base SAS feature that is available (in preproduction) as of version 9.1.3 and is able to generate production quality PDF reports directly since it provides SAS programmers simultaneous control over report layout and formatting.

This is the first in a series of papers that will document the evolution from a reporting process that is currently only 10% SAS to one that approaches 100% SAS.

The code discussed in this paper was developed with SAS 9.1.3-SP4 running under Windows XP Professional-SP2. The code was developed specifically for the PDF destination, you may experience some nuances when applying these techniques to other destinations.

Keywords: ODS DATA Step Object, DATA Step Component Interface (DSCI), macro, DATA NULL

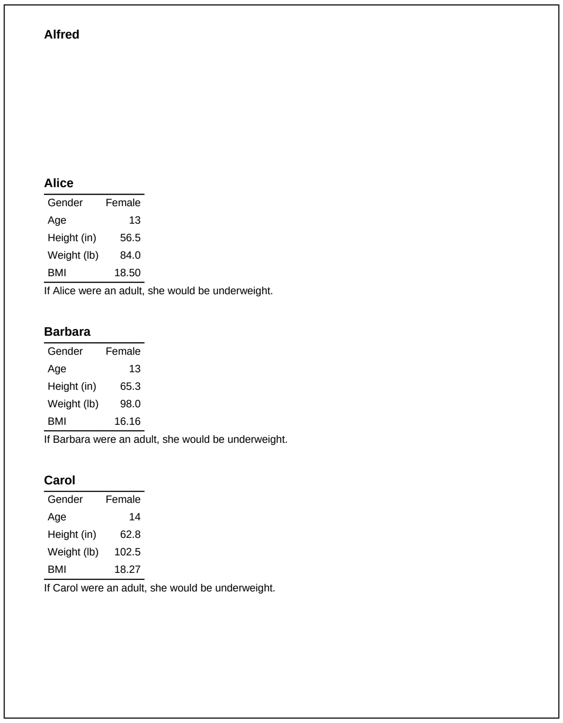
DATA NULL + ODS $<$ ODS DATA Step Object

The ODS DATA Step Object is an object-oriented approach to building reports–Daniel O'Connor's SUGI 28 paper provides a great background on the ODS DATA Step Object.

All this controll doesn't come without cost to the programmer. The syntax is quite particular with ordering of statements and micro-coding of the structure and format of the report.

If you've worked with HTML documents then you're already familiar with the idea of opening and closing tags. But HTML is a forgiving language; if you forget to close a paragraph, it will do it for you. Same with list items. XML, on the other hand, is very strict with its structure. Open tags must be closed and closed in proper order. If not, you're be stuck with a document that is not 'well-formed'.

In this sence, ODS DATA Step Object is much closer to XML than to HTML. Methods are called to start and end layouts, tables, table heads, table bodies, table feet, rows, and cells. They must also end in proper order (i.e., in the reverse order in which they were started). For each unit in the report (the methods), a set of arguments can be called. Some arguments are required for certain methods while most are optional. Most arguments are relevant to a handful of methods and are irrelevant for other methods. If you try using an argument that isn't appropriate for a given method, SAS will throw an error (hopefully the helpful variety) and halt the DATA step.



Alfred

Alice

Gender	Female
Age	13
Height (in)	56.5
Weight (lb)	84.0
BMI	18.50

If Alice were an adult, she would be underweight.

Barbara

Gender	Female
Age	13
Height (in)	65.3
Weight (lb)	98.0
BMI	16.16

If Barbara were an adult, she would be underweight.

Carol

Gender	Female
Age	14
Height (in)	62.8
Weight (lb)	102.5
BMI	18.27

If Carol were an adult, she would be underweight.

Figure 1: Simple Report for SASHELP.Class

To make the simple report in Figure 1, required a relatively convoluted program given—around 80 lines to create nineteen 5x2 cell tables with some leading and trailing text.

Listing 1: Simple Report for SASHELP.Class

```
proc format;
  value bmi
    /* http://www.cdc.gov/nccdphp/dnpa/bmi/adult_BMI/about_adult_BMI.htm */
    low  -< 18.5 ='underweight'
    18.5 -< 25.0 ='normal'
    25.0 -< 30.0 ='overweight'
    30.0 -  high ='obese';
  value $pronoun
    'F' = 'she' 'M' = 'he';
run;

ods listing close;
ods pdf file='class.pdf';
ods proclabel='ODS DATA Step Object Example Report';
data _null_;
  set sashelp.class end=last;
  if _n_ = 1 then declare odsout obj();
  /* http://www.cdc.gov/nccdphp/dnpa/bmi/adult_BMI/about_adult_BMI.htm */
  bmi = weight / (height*height) * 703;

  obj.layout_gridded ( width: '4in', height: '2.0in' );
  *obj.image( text: 'filename of pre-table image');
  obj.format_text( text: name, overrides: 'font=(helvetica,14 pt,bold)');
  obj.table_start( label: name );

    obj.row_start();
      obj.cell_start( inhibit: 'BLR' );
        obj.format_text( text: 'Gender', overrides: 'font_face=arial' );
      obj.cell_end();
      obj.cell_start( inhibit: 'BLR', overrides: 'just=right' );
        if sex eq 'M'
           then obj.format_text( text: 'Male', overrides: 'font_face=arial' );
           else obj.format_text( text: 'Female', overrides: 'font_face=arial' );
      obj.cell_end();
    obj.row_end();

    obj.row_start();
      obj.cell_start( inhibit: 'BLR' );
        obj.format_text( text: 'Age', overrides: 'font_face=arial' );
      obj.cell_end();
      obj.cell_start( inhibit: 'BLR', overrides: 'just=right' );
        obj.format_text( text: put(age, 3. ), overrides: 'font_face=arial');
      obj.cell_end();
    obj.row_end();

    obj.row_start();
      obj.cell_start( inhibit: 'BLR' );
        obj.format_text( text: 'Height (in)', overrides: 'font_face=arial' );
      obj.cell_end();
      obj.cell_start( inhibit: 'BLR', overrides: 'just=right' );
        obj.format_text( text: put(height,5.1), overrides: 'font_face=arial' );
      obj.cell_end();
    obj.row_end();

    obj.row_start();
      obj.cell_start( inhibit: 'BLR' );
        obj.format_text( text: 'Weight (lb)', overrides: 'font_face=arial' );
      obj.cell_end();
      obj.cell_start( inhibit: 'BLR', overrides: 'just=right' );
        obj.format_text( text: put(weight,5.1), overrides: 'font_face=arial' );
      obj.cell_end();
    obj.row_end();

    obj.row_start();
      obj.cell_start( inhibit: 'LR' );
        obj.format_text( text: 'BMI', overrides: 'font_face=arial' );
      obj.cell_end();
      obj.cell_start( inhibit: 'LR', overrides: 'just=right' );
        obj.format_text( text: put(bmi,5.2), overrides: 'font_face=arial' );
      obj.cell_end();
    obj.row_end();

  obj.table_end();
  obj.format_text( text: catx(' ', 'If', name, 'were an adult,'
                                 , put(sex,$pronoun.), 'would be', put(bmi,bmi.) || '.')
                 , overrides: 'font_face=arial');
  obj.layout_end();
  *if not last then obj.page(); *force a page break;
run;
ods pdf close;
ods listing;
```

USING MACROS TO REDUCE REPETITIOUS CODE

After reviewing Listing 1, you will probably notice that there is a lot of repetitious between `table_start()` and `table_end()`: start a row, start a cell, format the cell, end the cell, start a cell, format the cell, end the cell, end the row, start a row. . . and the cells and text share some common options. If you micro-code a lengthy ODS DATA Step Object report, you'll quicky get pulled into the minutia; Macro-coding (with macros and macro variables, of course) lets you keep your eye on the big picture while removing redundancies and isolating formatting.

Listing 2 shows the reporting macros and macro variables that were used. For each element in a list fed to the `%cutRows()` macro, the `%varRow()` macro is called and generates the row and cell starts and ends along with standardized formatting (`inhibit`s and `override`s).

Listing 2: Reporting Macros

```
%let h1     = %str(font=(arial, 14 pt, bold roman));
%let h2     = %str(font=(arial, 12 pt, italic narrow));
%let normal = %str(font=(times, 10 pt, roman));
%let head   = %str(font=(arial, 10 pt, bold));
%let stub   = %str(font=(arial, 9 pt));
%let cell   = %str(font=(arial, 9 pt, narrow));
%let foot   = %str(font=(arial, 8 pt, narrow) foreground=silver);
%let warn   = %str(font=(arial, 10 pt, bold roman) foreground=red);

%macro textRow(text);
  obj.row_start();
    obj.cell_start( inhibit: 'TLR', overrides: 'cellwidth=5.25in' );
      obj.format_text( text: "&text", overrides: "&head");
    obj.cell_end();
    obj.cell_start( inhibit: 'TL', overrides: 'cellwidth=0.60 in just=right' );
      obj.format_text( text: 'CU %', overrides: "&head foreground=silver" );
    obj.cell_end();
    obj.cell_start( inhibit: 'TR', overrides: 'cellwidth=0.60 in just=right' );
      obj.format_text( text: '%~{super 1}', overrides: "&head" );
    obj.cell_end();
    obj.cell_start( inhibit: 'TLR', overrides: 'cellwidth=0.80 in just=right' );
      obj.format_text( text: 'n / N~{super 1}', overrides: "&head" );
    obj.cell_end();
  obj.row_end();
%mend;

%macro varRow(var);
  *****TABLE COLUMNS*****;
  obj.row_start();
    obj.cell_start( inhibit: 'BLR' );
      obj.format_text( text: catx(' ', "&var..", "%vlabel(aos,&var)" ), overrides: "&stub" );
    obj.cell_end();
    obj.cell_start( inhibit: 'BL', overrides: 'just=right' );
      obj.format_text( text: put(&var._cu, percent8.1), overrides: "&cell foreground=silver" );
    obj.cell_end();
    obj.cell_start( inhibit: 'BR', overrides: 'just=right' );
      obj.format_text( text: put(&var, percent8.1), overrides: "&cell" );
    obj.cell_end();
    obj.cell_start( inhibit: 'BLR', overrides: 'just=right' );
      obj.format_text( text: catx(' / ', put(&var._1, comma8.), put(&var._n, comma8.))
                     , overrides: "&cell" );
    obj.cell_end();
  obj.row_end();
%mend;

%macro vlabel(dsn,var);
  %local dsid vlabel;
  %let dsid=%sysfunc(open(&dsn));
  %let vlabel=%qsysfunc(varlabel(&dsid,%sysfunc(varnum(&dsid,&var))));
  %let dsid=%sysfunc(close(&dsid));
  &vlabel
%mend vlabel;

%macro cutRows/parmbuff;
   %let num=1;
   %let var=%scan(&syspbuff,&num);
   %do %while(&var ne);
      %varRow(&var)
      %let num=%eval(&num+1);
      %let var=%scan(&syspbuff,&num);
   %end;
%mend;
```

Each time ODS DATA Step Object code is submitted, SAS reminds us that the ODS "data step interface is experimental in this release". When working with preproduction features, it's a good idea to save often (every few minutes isn't too often) and submit more often (after each marginally significant change to confirm everything runs as expected). Windows users can use FileHamster which automatically copies versions of the file at each save point. When you're beginning to work with ODS DATA Step Object, errors tend to creep into code pretty fast. If you forget to save and submit for too long, weeding out errors quickly becomes a Sisyphean task.

As luck would have it, the class report (Listing 1) is one of those cases where too much was changed between runs and an error snuck in—did you notice Alfred's information was missing in Figure 1? Since the listing was intended to show the repetitious code, I let the error stand to illustrate my point of saving often and submiting frequently.

TWO TYPES OF ERRORS

Occasionally, the error messages SAS generates are quite helpful. For example, this error message told me that `column_span` was not a valid option for the `override` argument and proceeded to give me a listing of the options that were valid:

```
ERROR 22-322: Syntax error, expecting one of the following: ;, ABSTRACT, ACTIVELINKCOLOR, ASIS, BACKGROUND,
BACKGROUNDIMAGE, BODYSCROLLBAR, BODYSIZE, BORDERCOLOR, BORDERCOLORDARK, BORDERCOLORLIGHT, BORDERWIDTH, BOTTOMMARGIN,
BULLET, CELLPADDING, CELLSPACING, CONTENTPOSITION, CONTENTSCROLLBAR, CONTENTSIZE, CONTRASTCOLOR, CSSSTYLE, CSSTEXT,
DESCRIPTION, DROPSHADOW, ENDCOLOR, FILLRULEWIDTH, FLYOVER, FONT, FONT_FACE, FONT_SIZE, FONT_STYLE, FONT_WEIGHT,
FONT_WIDTH, FOREGROUND, FRAME, FRAMEBORDER, FRAMEBORDERWIDTH, FRAMESPACING, GRADIENT_DIRECTION, HREFTARGET, HTMLCLASS,
HTMLCONTENTTYPE, HTMLDOCTYPE, HTMLID, HTMLSTYLE, IMAGE, IMAGESTYLE, INDENT, JUST, LEFTMARGIN, LINESTYLE, LINETHICKNESS,
LINKCOLOR, LISTENTRYANCHOR, LISTENTRYDBLSPACE, MARKERSIZE, MARKERSYMBOL, MINOR, NEUTRALCOLOR, NOBREAKSPACE, OFFSET,
OUTPUTHEIGHT, OUTPUTWIDTH, OVERHANGFACTOR, PAGEBREAKHTML, POSTHTML, POSTIMAGE, POSTTEXT, PREHTML, PREIMAGE, PRETEXT,
PROTECTSPECIALCHARS, RIGHTMARGIN, RULES, STARTCOLOR, TAGATTR, THRESHOLD, TICKDISPLAY, TOPMARGIN, TRANSPARENCY, URL,
VISITEDLINKCOLOR, VJUST, WATERMARK.
```

More often, however, the error messages are somewhat less than useful:

```
ERROR:  Read Access Violation In Task [ DATASTEP )
Exception occurred at (60DA1222)
Task Traceback
Address   Frame     (DBGHELP API Version 4.0 rev 5)
60DA1222  03B3F880  sasmrk:mcn_main+0x50222
60DA1046  03B3F8B4  sasmrk:mcn_main+0x50046
60D65210  03B3F950  sasmrk:mcn_main+0x14210
60C1E04B  03B3F998  saspso:mcn_main+0x1D04B
60D62C20  03B3FA90  sasmrk:mcn_main+0x11C20
60C1D2A3  03B3FAE0  saspso:mcn_main+0x1C2A3
60D6127F  03B3FC20  sasmrk:mcn_main+0x1027F
60D556F8  03B3FCBC  sasmrk:mcn_main+0x46F8
60D551A4  03B3FCE4  sasmrk:mcn_main+0x41A4
67A06B0A  03B3FD48  saswob:mcn_main+0x5B0A
60E73E43  03B3FE5C  sasdsi:mcn_main+0x2E43
6725B296  03B3FEC0  sasdsx:mcn_main+0x2A296
05DA9661  03B3FEE4  0003:00009661 <unknown>
01307DB5  03B3FF04  sashost:Main+0xB1125
6723158A  03B3FF44  sasdsx:mcn_main+0x58A
673115BC  03B3FF88  sasds:mcn_main+0x5BC
01262B02  03B3FFA0  sashost:Main+0xBE72
01266C20  03B3FFB4  sashost:Main+0xFF90
7C80B683  03B3FFEC  kernel32:GetModuleFileNameA+0x1B4

ERROR: Generic error in the XOB subsystem.
```

Saving often and submitting frequently can help reverse programming errors.

THE SOURCE DATA

Submissions from a graduate survey will serve as the actual source data for the actual report, for these examples, however, fictitious data was used. The granularity of the source data is one observation per graduate with each survey item represented by one variable.

The immediate goal in programming this process is to automate the current reporting needs: upper management needs to see results aggregated by school and degree (by 'program'). As described above, this was a tedious process of transcriptionresulting in one-page summary sheets for each program. In its current state, this process is definately not scalable, but by programming the full process from raw data to aggregated data to final reports, the process can easily be scaled out when finer aggregations are requested.

PREPARING THE DATA

The source data set needs to be summarized and aggregated by program. This is handled by a custom macro that aggregates one variable at a time to a separate data set. These data sets are then combined into a single data set;

this is the data set used for the report-generating data step. There are a variety of item types in the survey, and each item type is aggregated according to different rules. For example, Likert-type response items are aggregated as top responders / valid responders (see Listing 3). The resulting data set contains the by variables and two sets (one for the subgroup and one for the population as a whole) of three variables (the count of top responders, the total count of responders, and the percent of top responders).

Listing 3: The `%aggregate()` Macro

```
proc format;
  value two_five 1-2=0 3-5=1 other=.;
  value two_four 1-2=0 3-4=1 other=.;
run;

%macro aggregate(data=aos, var=, fmt=two_five, print=N)
      / des='Aggregate single-select items';
proc freq data=&data noprint;
  by newschool degree;
  where put(&var,&fmt..) ne '.'; *ignore invalid formatted values;
  format &var &fmt..;
  table &var
      / out=&var(drop=percent where=(&var ne .));
run;
proc transpose data   =&var
               out    =&var(drop= _name_ _label_)
               prefix =&var._;
  by newschool degree;
  id &var;
run;
proc sql;
  create table &var as
  select
    d.newschool
  , d.degree
  , &var._1 / (&var._0 + &var._1) as &var format=percent8.2
  , &var._1
  , &var._0 + &var._1 as &var._n
  , a.*
  from &var as d,
  ( select
      sum(&var._1) / (sum(&var._0) + sum(&var._1)) as &var._cu format=percent8.2
    , sum(&var._1) as &var._cu_1
    , sum(&var._0) + sum(&var._1) as &var._cu_n
    from &var ) as a
  order by newschool, degree;
quit;
%if &print ne N %then %do;
proc print data=&var noobs; run;
%end;
%mend;
```

THE REPORT

The structure of the reporting data step will depend heavily upon the the structure of the aggregated data and upon the desired final report structure. Since the desired report structure should closely resemble the structure of the existing report, it seems appropriate to describe the current report now.

Pages One page per subgroup; subgroups will be determined via by variables to increase scalability.

Subgroup Description Provide an explanation of the subgroup (what school and degree) along with response count, rate, and graduation time span.

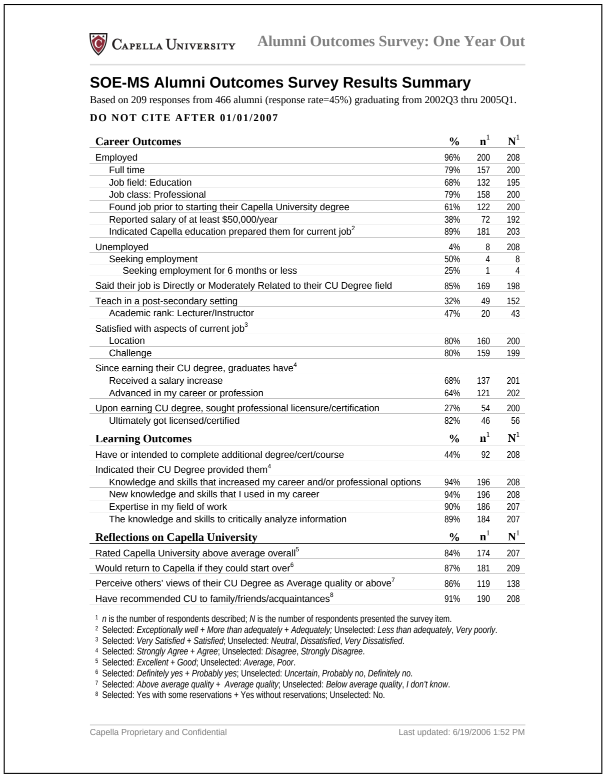
Table Rows Heading rows and data rows with one key survey item consuming one row in the report.

Table Columns Specify the key survey item along with quantification of the current subgroup and population comparison.

Notes Some of the key variables are aggregated differently. These differences are specified at the end of each page along with a date stamp.

Listing 4: The New Report Code

```
option nonumber;
title;
footnote;
ods escapechar="~";
ods listing close;
ods pdf file='report.pdf';
ods proclabel='Results from Alumni Outcomes Survey: One Year Out';
data _null_;
  set aos_aggregated end=last;
  where degree ne 'PSD';
  if _n_ = 1 then declare odsout obj();

  obj.image( text: 'CU_Color_Horiz_Small.gif' );
  *****SUBGROUP DESCRIPTION*****;
  obj.format_text( text: catx(' ', catx('-', NewSchool, Degree)
                                 , 'Results from Alumni Outcomes Survey: One Year Out')
                 , overrides: "&H1" );
  obj.format_text( text: catx(' ', 'Based on',put(didSubmit_1,comma8.-l)
                                 , 'submissions from', put(didSubmit_n,comma8.-l), 'alumni'
                                 , '(' || put(didsubmit,percent7.1-l), 'response rate)'
                                 , 'graduating from', put(graddate_min,yyq6.)
                                 , 'thru', put(graddate_max,yyq6.) || '.')
                 , overrides: "&normal");
  obj.format_text( text: catx('    '
     , catx(' ','~S={foreground=red font_weight=bold}DO NOT CITE AFTER'
               , put(intnx('year.7',"&sysdate9"d,1) - 1, mmddyy10.))
     , '~S={foreground=black font_weight=light}Questions? Contact AssessmentOffice@capella.edu'
     , "~S={font_style=italic}Generated &sysdate9..")
                 , overrides: "&normal");

  obj.layout_gridded ( width: '8.5in', height: '11in', columns: 1, row_gutter: '0in' );
  obj.region();

  obj.table_start( label: catx('-', NewSchool, Degree) );

  *****TABLE ROWS*****;
  %textRow(Satisfied with aspects of current job~{super 3})
  %cutRows(Q17_1 Q17_2 Q17_3 Q17_4 Q17_5 Q17_6)

  %textRow(Since earning their Capella University degree%str(,) graduates have~{super 4})
  %cutRows(Q18_1 Q18_2 Q18_3 Q18_4 Q18_5 Q18_6)

  %textRow(Indicated their Capella University Degree provided them~{super 4})
  %cutRows(Q25_1 Q25_2 Q25_3 Q25_4 Q25_5 Q25_6 Q25_7 Q25_8 Q25_9 Q25_10
           Q25_11 Q25_12 Q25_13 Q25_14 Q25_15)

  *****NOTES*****;
  obj.row_start();
    obj.cell_start( inhibit: 'BLR', column_span: 4);
      obj.format_text( text: catx('~n'
                      , 'Note: CU% represents the overall % for all Capella alumni in the specified timeframe.'
                      , catx('; ','~{super 1} n = number of respendents described'
                                 , 'N = number of people providing a valid response to the item; % = n / N')
                      , catx('; ','~{super 2} Selected: Exceptionally well + More than adequately + Adequately'
                                 , 'Unselected: Less than adequately, Very poorly.')
                      , catx('; ','~{super 3} Selected: Very Satisfied + Satisfied'
                                 , 'Unselected: Neutral, Dissatisfied, Very Dissatisfied.')
                      , catx('; ','~{super 4} Selected: Strongly Agree + Agree'
                                 , 'Unselected: Disagree, Strongly Disagree.')
                      , catx('; ','~{super 5} Selected: Excellent + Good'
                                 , 'Unselected: Average, Poor.')
                      , catx('; ','~{super 6} Selected: Definitely yes + Probably yes'
                                 , 'Unselected: Uncertain, Probably no, Definitely no.')
                      , catx('; ','~{super 7} Selected: Above average quality + Average quality'
                                 , 'Unselected: Below average quality, I don''t know.')
                      , catx('; ','~{super 8} Selected: Yes with some reservations + Yes without reservations'
                                 , 'Unselected: No.')
                       )
                      , overrides: "&foot");
    obj.cell_end();
  obj.row_end();
  obj.table_end();
  obj.layout_end();
  if not last then obj.page(); *force a page break;
run;
ods pdf close;
ods listing;
```



Capella University Alumni Outcomes Survey: One Year Out

SOE-MS Alumni Outcomes Survey Results Summary

Based on 209 responses from 466 alumni (response rate=45%) graduating from 2002Q3 thru 2005Q1.

DO NOT CITE AFTER 01/01/2007

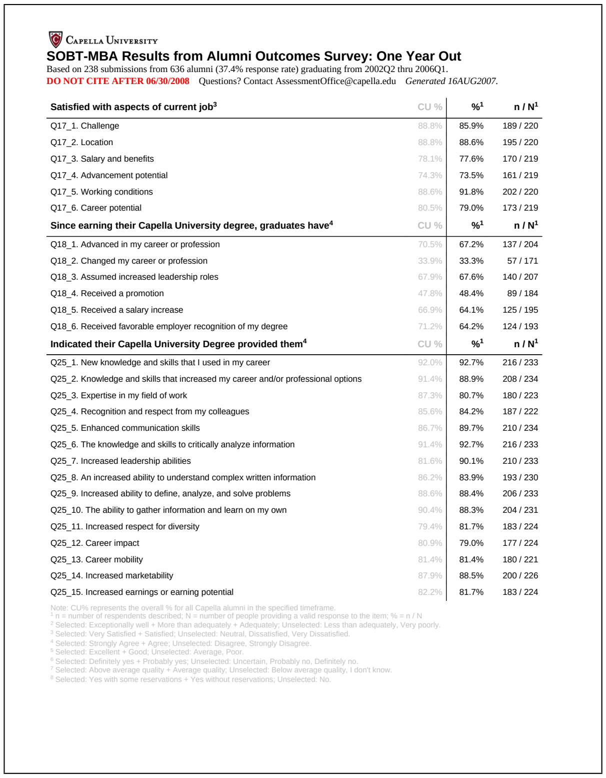
Career Outcomes	%	n[1]	N[1]
Employed	96%	200	208
Full time	79%	157	200
Job field: Education	68%	132	195
Job class: Professional	79%	158	200
Found job prior to starting their Capella University degree	61%	122	200
Reported salary of at least $50,000/year	38%	72	192
Indicated Capella education prepared them for current job[2]	89%	181	203
Unemployed	4%	8	208
Seeking employment	50%	4	8
Seeking employment for 6 months or less	25%	1	4
Said their job is Directly or Moderately Related to their CU Degree field	85%	169	198
Teach in a post-secondary setting	32%	49	152
Academic rank: Lecturer/Instructor	47%	20	43
Satisfied with aspects of current job[3]			
Location	80%	160	200
Challenge	80%	159	199
Since earning their CU degree, graduates have[4]			
Received a salary increase	68%	137	201
Advanced in my career or profession	64%	121	202
Upon earning CU degree, sought professional licensure/certification	27%	54	200
Ultimately got licensed/certified	82%	46	56

Learning Outcomes	%	n[1]	N[1]
Have or intended to complete additional degree/cert/course	44%	92	208
Indicated their CU Degree provided them[4]			
Knowledge and skills that increased my career and/or professional options	94%	196	208
New knowledge and skills that I used in my career	94%	196	208
Expertise in my field of work	90%	186	207
The knowledge and skills to critically analyze information	89%	184	207

Reflections on Capella University	%	n[1]	N[1]
Rated Capella University above average overall[5]	84%	174	207
Would return to Capella if they could start over[6]	87%	181	209
Perceive others' views of their CU Degree as Average quality or above[7]	86%	119	138
Have recommended CU to family/friends/acquaintances[8]	91%	190	208

[1] *n* is the number of respondents described; *N* is the number of respondents presented the survey item.
[2] Selected: *Exceptionally well + More than adequately + Adequately;* Unselected: *Less than adequately*, *Very poorly*.
[3] Selected: *Very Satisfied + Satisfied*; Unselected: *Neutral*, *Dissatisfied*, *Very Dissatisfied*.
[4] Selected: *Strongly Agree + Agree*; Unselected: *Disagree*, *Strongly Disagree*.
[5] Selected: *Excellent + Good*; Unselected: *Average*, *Poor*.
[6] Selected: *Definitely yes + Probably yes*; Unselected: *Uncertain*, *Probably no*, *Definitely no*.
[7] Selected: *Above average quality + Average quality*; Unselected: *Below average quality*, *I don't know*.
[8] Selected: Yes with some reservations + Yes without reservations; Unselected: No.

Capella Proprietary and Confidential   Last updated: 6/19/2006 1:52 PM



Capella University

SOBT-MBA Results from Alumni Outcomes Survey: One Year Out

Based on 238 submissions from 636 alumni (37.4% response rate) graduating from 2002Q2 thru 2006Q1.

DO NOT CITE AFTER 06/30/2008   Questions? Contact AssessmentOffice@capella.edu   *Generated 16AUG2007.*

Satisfied with aspects of current job[3]	CU %	%[1]	n / N[1]
Q17_1. Challenge	88.8%	85.9%	189 / 220
Q17_2. Location	88.8%	88.6%	195 / 220
Q17_3. Salary and benefits	78.1%	77.6%	170 / 219
Q17_4. Advancement potential	74.3%	73.5%	161 / 219
Q17_5. Working conditions	88.6%	91.8%	202 / 220
Q17_6. Career potential	80.5%	79.0%	173 / 219
Since earning their Capella University degree, graduates have[4]	**CU %**	**%[1]**	**n / N[1]**
Q18_1. Advanced in my career or profession	70.5%	67.2%	137 / 204
Q18_2. Changed my career or profession	33.9%	33.3%	57 / 171
Q18_3. Assumed increased leadership roles	67.9%	67.6%	140 / 207
Q18_4. Received a promotion	47.8%	48.4%	89 / 184
Q18_5. Received a salary increase	66.9%	64.1%	125 / 195
Q18_6. Received favorable employer recognition of my degree	71.2%	64.2%	124 / 193
Indicated their Capella University Degree provided them[4]	**CU %**	**%[1]**	**n / N[1]**
Q25_1. New knowledge and skills that I used in my career	92.0%	92.7%	216 / 233
Q25_2. Knowledge and skills that increased my career and/or professional options	91.4%	88.9%	208 / 234
Q25_3. Expertise in my field of work	87.3%	80.7%	180 / 223
Q25_4. Recognition and respect from my colleagues	85.6%	84.2%	187 / 222
Q25_5. Enhanced communication skills	86.7%	89.7%	210 / 234
Q25_6. The knowledge and skills to critically analyze information	91.4%	92.7%	216 / 233
Q25_7. Increased leadership abilities	81.6%	90.1%	210 / 233
Q25_8. An increased ability to understand complex written information	86.2%	83.9%	193 / 230
Q25_9. Increased ability to define, analyze, and solve problems	88.6%	88.4%	206 / 233
Q25_10. The ability to gather information and learn on my own	90.4%	88.3%	204 / 231
Q25_11. Increased respect for diversity	79.4%	81.7%	183 / 224
Q25_12. Career impact	80.9%	79.0%	177 / 224
Q25_13. Career mobility	81.4%	81.4%	180 / 221
Q25_14. Increased marketability	87.9%	88.5%	200 / 226
Q25_15. Increased earnings or earning potential	82.2%	81.7%	183 / 224

Note: CU% represents the overall % for all Capella alumni in the specified timeframe.
[1] n = number of respendents described; N = number of people providing a valid response to the item; % = n / N
[2] Selected: Exceptionally well + More than adequately + Adequately; Unselected: Less than adequately, Very poorly.
[3] Selected: Very Satisfied + Satisfied; Unselected: Neutral, Dissatisfied, Very Dissatisfied.
[4] Selected: Strongly Agree + Agree; Unselected: Disagree, Strongly Disagree.
[5] Selected: Excellent + Good; Unselected: Average, Poor.
[6] Selected: Definitely yes + Probably yes; Unselected: Uncertain, Probably no, Definitely no.
[7] Selected: Above average quality + Average quality; Unselected: Below average quality, I don't know.
[8] Selected: Yes with some reservations + Yes without reservations; Unselected: No.

Here you can see the original report (left) and the report up to this point (right). The appearance of the new report is not intended to be a perfect matchto the appearance of the current report, but rather a close approximation. In the next part of this series, I will walk through the dynamic selection of variables for reporting which should hopefully end with the desired report created 100% in SAS. Stay tuned.

It should again be noted that fictitious data was used in the creation of the pictured reports.

END MATTER

SOME RESOURCES

Hamilton, J. (2004). *Using New Features in ODS to Create Master/Detail Reports*. SUGI 29, Montréal, Québec, Canada. `http://www.excursive.com/sas/p229-29.pdf`

Koopmann, R. Jr. (2007). *ODS DATA Step Object 101: Tip Sheet*
`http://www.sascommunity.org/mwiki/images/b/be/ODS_DATA_Step_Object_Method_101-Tip_Sheet.pdf`

Ladan, A. I. (2006). *The Absolute Nitty-griddy of ODS Layout: Part I*. SESUG 2006, Atlanta, GA.
`http://www8.sas.com/scholars/Proceedings/2006/Posters/PO02_06.PDF`

O'Connor, D. (2003). *Methods for the ODS DATA Step Object*.
`http://support.sas.com/rnd/base/topics/datastep/dsobject`

O'Connor, D. (2003). *Next Generation `DATA _NULL_` Report Writing Using ODS OO Features*. SUGI 28, Seattle, WA. `http://www2.sas.com/proceedings/sugi28/022-28.pdf`

CONTACT INFORMATION

Your comments and questions on this paper and on my ODS DATA Step Object tip sheet are valued and encouraged. Feel free to contact the author:

Name	Richard Koopmann Jr.
Company	Capella University
E-mail	richard [dot] koopmann [at] capella [dot] edu
sasCommunity	User:RichardK

COLOPHON

This document was typeset in LaTeX using the following document class, packages, and editor:

sugconf A custom document class for SAS User Group Conference proceedings.

listings Include source code from the source files directly into LaTeX document. . . can Word do that?

textcomp LaTeX automatically converts straight quotes (which SAS likes) to curly quotes (which SAS does not like). This package reverses this effect to make included code more portable through copy-paste-run.

graphicx Allows graphics to be included from a variety of file formats—including PDF!

floatflt Wraps text around graphics to conserve space.

hyperref Fine-tunes PDF (document properties, link properties, etc.).

PSPad An extremely powerful freeware text editor that supports over 100 different programming and scripting languages along with syntax highlighting (for Windows only).