With a Trace: Making Procedural Output and ODS Output Objects Work For You

Louise S. Hadden, Abt Associates Inc.

ABSTRACT

SAS® procedures can convey an enormous amount of information – sometimes more information than is needed. Most SAS procedures generate ODS objects behind the scenes. SAS uses these objects with style templates that have custom buckets for certain types of output to produce the output that we see in all destinations (including the SAS listing). By tracing output objects and ODS templates using ODS TRACE (DOM) and by manipulating procedural output and ODS OUTPUT objects, we can pick and choose just the information that we want to see. We can then harness the power of SAS data management and reporting procedures to coalesce the information collected and present the information accurately and attractively.

INTRODUCTION

The Output Delivery System (ODS) delivers what used to be printed output in many convenient forms. What many of us don't realize is that "printed output" from procedures (whether the destination is PDF, RTF, or HTML) is the result of SAS® packaging a collection of items that come out of a procedure that most people want to see in a predefined order (aka template.) With tools such as ODS TRACE, ODS TRACE DOM, ODS OUTPUT, ODS destinations, SAS metadata, SAS Graphics Editor, and reporting procedures, this paper explores the many buried treasures of procedural output and ODS output objects and demonstrates how to use these objects to get exactly the information that is needed, in exactly the format wanted.

Two full sample programs, one designed to be run interactively, and one designed to be run in batch mode, are provided on in the appendix. These programs use the SASHELP.CLASS data set provided with SAS software and demonstrate basic concepts. Code snippets from the programs to illustrate key concepts are provided in the text of the paper.

Additionally, two further examples are provided showing how ODS OUTPUT objects can be used to create a multi-tab Excel codebook and complex analytic tables.

This presentation is suitable for all levels of proficiency and will be useful for programmers working in all industries. Examples shown were run using SAS 9.4 Maintenance Release 5 on a Windows Server platform.

GETTING STARTED

There are four basic concepts involved in re-tooling ODS output objects to produce custom reports:

1. Identify / locate your ODS output object(s) using ODS TRACE or output data set(s) .

2. Analyze your ODS output object(s) or output data set(s) using basic SAS procedures and review.

3. Manipulate your ODS output object(s) or output data set(s) using SAS data steps and/or procedures.

4. Report on your final data set.

These four concepts are explored below.

DISCOVERING ODS OUTPUT OBJECTS

Most SAS procedures generate ODS objects behind the scenes. SAS uses these objects in conjunction with style templates that have custom "buckets" for certain types of output to produce the output we see in all destinations (including the SAS listing).

Use the ODS TRACE command to identify ODS output objects. Choose the procedure that you are using, and "surround" the procedure (including any ODS GRAPHICS calls, etc. you may be using) with ODS TRACE and ODS TRACE OFF commands.

```
ods trace on / label;
ods rtf file=yourfilename.rtf' path=odsout style=styles.journal2;
ods graphics on;
. . . your procedures here . . .
ods rtf close;
ods graphics off;
ods trace off;
```

If you are running interactively, you will see the ODS objects generated in the results window on the left, and can select, view and save these objects. In some cases, you can manipulate the objects. ODS trace information will be located in your SAS log. Although the ODS objects can be viewed in the results window, you will not get enough information about the objects to truly customize your output, so SAS logs and lists is an essential tool for both interactive and batch users.

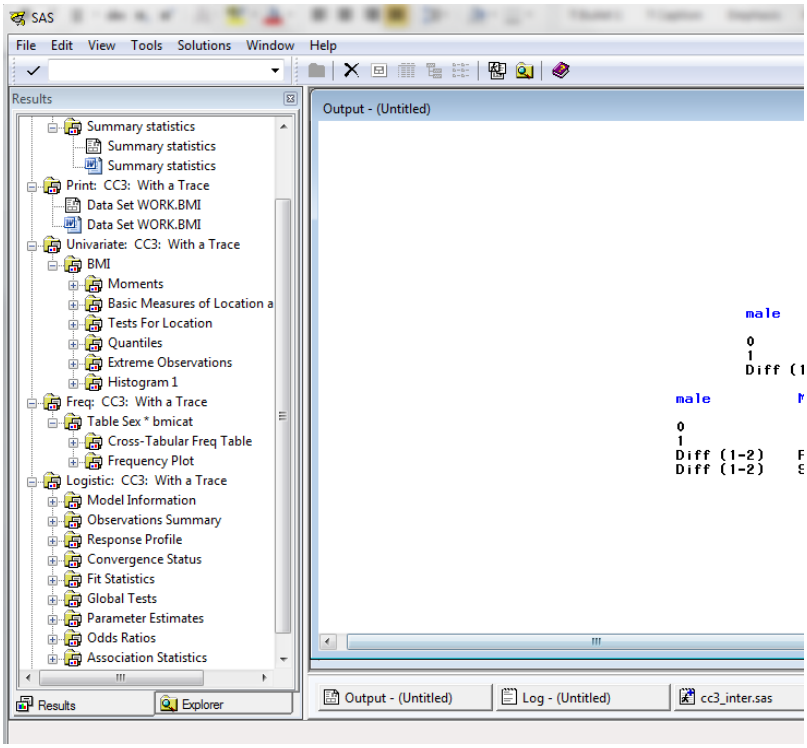


Figure 1. ODS Output Data Sets

If you are running any SG procedures interactively, use the following command before running the procedures.

```
ods listing sge=on;
```

This creates an editable (using the ODS graphics editor) file which allows you to change titles, background colors, etc. in your graphs which you can then save.

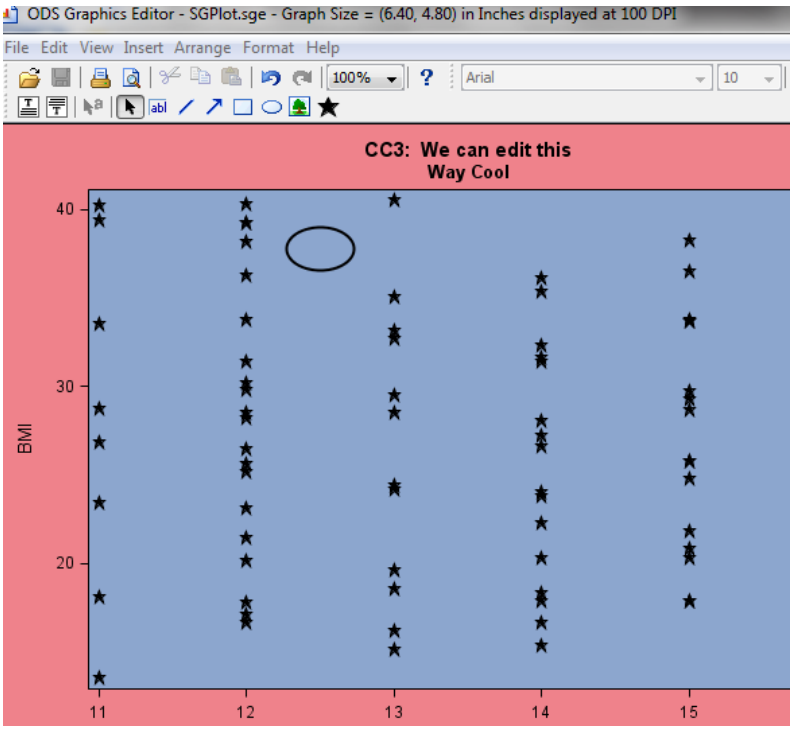


Figure 2. ODS GRAPHICS EDITOR

If you run in batch mode, output from the ODS TRACE command will be included in your log file.

ADDITIONAL NOTES ON THE ODS TRACE STATEMENT

An option on the ODS TRACE statement is ODS TRACE DOM. DOM stands for the ODS Document Object Manager, which provides background information to ODS about document style selectors, etc. As with other ODS output objects, ODS TRACE DOM provides information in the log about ODS style elements which can be very useful for controlling and enhances ODS output.

EXPLORING ODS OUTPUT OBJECTS

After running one of the provided sample programs in the appendix (cc3_batch.sas) we can review our log and identify all the output objects generated by a specific procedure and learn their system names.

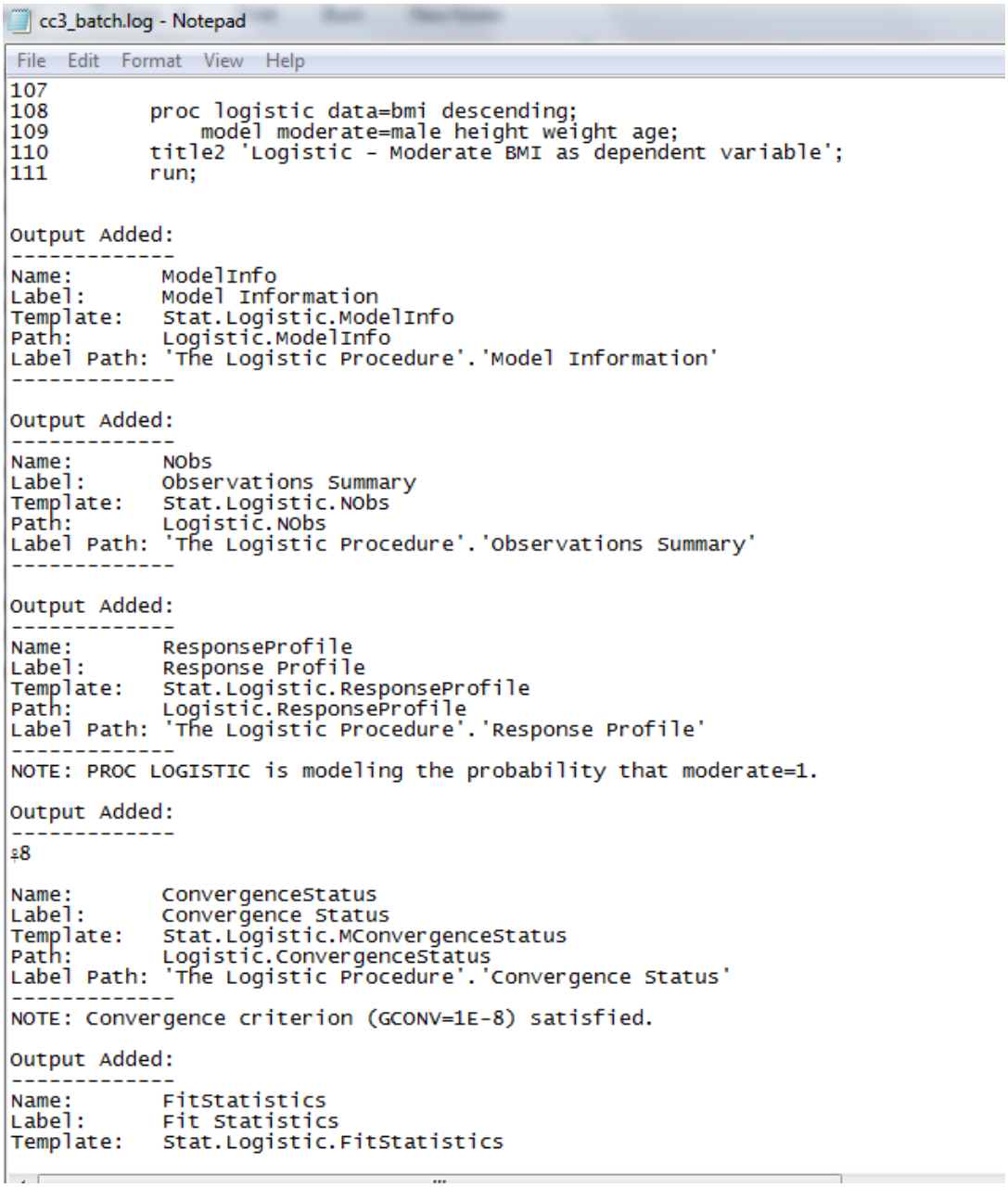


Figure 3. ODS TRACE output

We can see that PROC LOGISTIC generated many ODS output objects (datasets) that go into the "print" output: ModelInfo, Nobs, ResponseProfile, ConvergenceStatus, FitStatistics, GlobalTests, ParameterEstimates, OddsRatio, and Association. These datasets are not on the same level and in order to create a single data set from the various pieces, more exploration is needed. To save these temporary files to either work or permanent data sets, use the ODS OUTPUT statement.

```
ODS OUTPUT
modelinfo=modelinfo
nobs=nobs
responseprofile=responseprofile
convergencestatus=convergencestatus
```

```
fitstatistics=fitstatistics
globaltests=globaltests
parameterestimates=parameterestimates
oddsratios=oddsratios
association=association;
PROC LOGISTIC . . .
ODS OUTPUT CLOSE;
proc contents data=modelinfo varnum;
run;
proc print data=modelinfo (obs=10) noobs;
run; . . .
```

It is recommended that you do a test print of 10 observations even if you are working with a small data set, otherwise your output could be very large. 10 observations is generally enough to get a good picture. The contents is also very important as fields which might appear to be numeric may be character – SAS creates "print" variables for its output – and formatting give you valuable information for further manipulations.

Test print of parameter estimates:

Variable	*DF*	*Estimate*	*StdErr*	*WaldChiSq*	*ProbChiSq*
Intercept	1	-0.5111	3.5321	0.0209	0.8849
male	1	0.8934	0.6131	2.1238	0.1450
Height	1	-0.0168	0.1049	0.0257	0.8728
Weight	1	0.0154	0.00721	4.5349	0.0332
Age	1	-0.1560	0.3291	0.2246	0.6355

PROC CONTENTS of parameter estimates:

		Variables in Creation Order			
#	*Variable*	*Type*	*Len*	*Format*	*Label*
1	Variable	Char	9		
2	DF	Num	8	2.	
3	Estimate	Num	8	D8.	
4	StdErr	Num	8	D8.	Standard Error
5	WaldChiSq	Num	8	10.4	Wald Chi-Square
6	ProbChiSq	Num	8	PVALUE6.4	Pr > Chi-Square

ADDITIONAL NOTES ON THE ODS OUTPUT STATEMENT

Options for the ODS OUTPUT statement include ODS SELECT and ODS EXCLUDE which allow you to pick and choose elements that are sent to ODS destinations. SAS has defaults for ODS Output (all items are EXCLUDED unless overwritten in a ODS OUTPUT statement) and other destinations (all items are SELECTED unless otherwise specified.) If you want to see the defaults, use the ODS SHOW statement, which provides a selection list in the log. It's possible to customize all your ODS output by maintaining your own selection list(s).

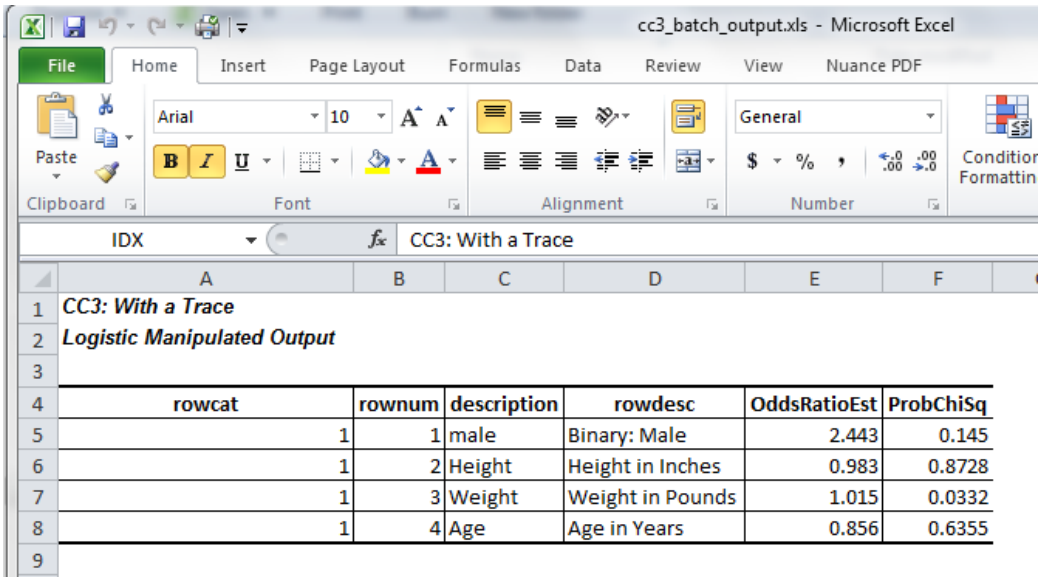
MANIPULATING ODS OUTPUT OBJECTS

Reviewing all the output above allows us to pick and choose statistics from various data sets, and merge it with a descriptive data set for printing. The possibilities are endless. In this case, I used parameter estimates and odds ratio output to construct a single data set with all the statistics the analyst wanted. Note that in order to merge the two data sets, I needed to remove the intercept line from the parameter estimates. The reasoning behind the test prints and PROC CONTENTS outputs becomes clear. The descriptive data set may include longer variable descriptions you may not want to carry in a large data set and variables used for formatting (for example, shading, italicizing, or bolding.) You can also create macro variables for printing from some ODS output objects using data steps and call symput, or PROC SQL INTO:.

```
data lhs;
length rowdesc $ 32;
rowcat=1; rownum=1; rowdesc='Binary: Male'; output;
rowcat=1; rownum=2; rowdesc='Height in Inches'; output;
rowcat=1; rownum=3; rowdesc='Weight in Pounds'; output;
rowcat=1; rownum=4; rowdesc='Age in Years'; output;
run;

data a (drop=variable);
length description $ 32;
set parameterestimates (where=(variable ne 'Intercept'));
counter=_n_+4;
description=variable;
run;

data b (drop=effect);
length description $ 50;
set oddsratios ;
counter=_n_+4;
description=effect;
run;

data d;
length description $ 50;
merge a b;
by counter;
run;

data c (drop=label);
length description $ 50;
set nobs (keep=label n);
description=label;
counter=_n_;
run;

data editoutput;
set d;
if description= 'male' then do; rowcat=1; rownum=1; end;
if description= 'Height' then do; rowcat=1; rownum=2; end;
if description= 'Weight' then do; rowcat=1; rownum=3; end;
if description= 'Age' then do; rowcat=1; rownum=4; end;
run;
```

```
proc sort data=editoutput;
by rowcat rownum;
run;

proc sort data=lhs;
by rowcat rownum;
run;

data printoutput;
merge editoutput lhs;
by rowcat rownum;
run;
```

It then becomes a simple exercise to report on the designer logistic data set.



CC3: With a Trace

Logistic Manipulated Output

rowcat	rownum	description	rowdesc	OddsRatioEst	ProbChiSq
1	1	male	Binary: Male	2.443	0.145
1	2	Height	Height in Inches	0.983	0.8728
1	3	Weight	Weight in Pounds	1.015	0.0332
1	4	Age	Age in Years	0.856	0.6355

Figure 4. Report on Designer PROC LOGISTIC data set

ODS OUTPUT OBJECTS BY EXAMPLE

Below follow two examples of how ODS output objects can be manipulated to create custom reports.

CREATE A MULTI-TABBED METADATA SPREADSHEET

We wished to create an Excel workbook with an index tab with basic information about all data sets being delivered and documented for a given month in a given domain (files to be uploaded to DATA.MEDICARE.GOV), plus tabs for each data set with selected details on variables. In this case, we used the CONTENTS procedure in conjunction with ODS output objects. PROC CONTENTS does create an output data set (OUT=); however, it does not contain one desired piece of information, the variables that a data set are sorted by (if any). It also is rectangular, meaning that many variables on a system level are included on every record. ODS RTF or ODS HTML calls sandwiched around PROC CONTENTS also yields a lot of unnecessary information and would require a lot of post-editing. A macro was designed that produces a PROC CONTENTS for each data set, outputs ODS output objects, manipulates the objects and outputs two temporary data sets (a header line for the index tab, and a

variable listing) for each data set.

After running ODS TRACE on the PROC CONTENTS statement, we reviewed our log and identified all the output objects generated by the contents procedure and their system names. Note that the output objects may vary depending on the procedure AND procedural options – and in some cases, the data! For example, SAS/STAT®'s PROC SURVEYFREQ produces different output objects based on whether you are running a one-way table or crosstabs, and may NOT generate chi-squared statistics if there are any 0 cells in a crosstab. In our case, if a data set is not sorted, PROC CONTENTS will not generate ODS output objects relating to sorting (SORTEDBY). If you want to report on sorting, you may need to have conditional processing that will indicated that a dataset is not sorted if there are no ODS output objects related to sorting. In the case of PROC CONTENTS, you will not get the variable "informat" in the VARIABLES output object if no variables in the data set have informats.

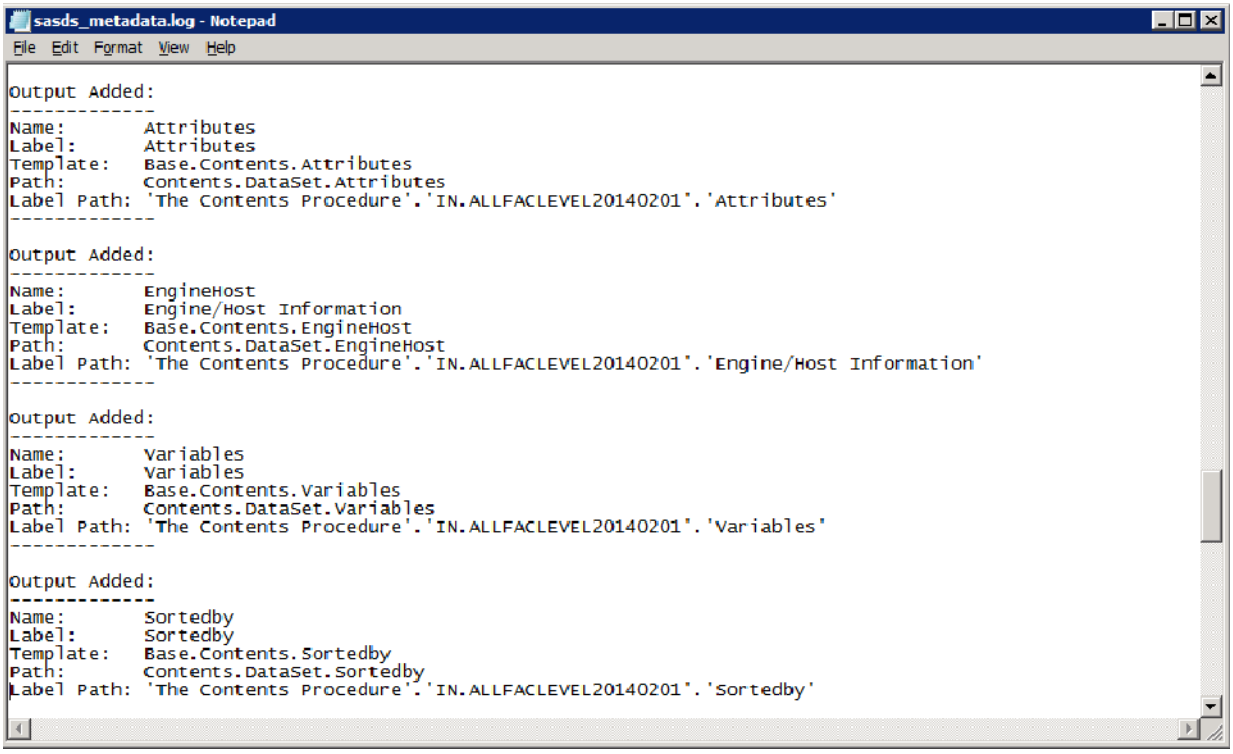


Figure 5. ODS TRACE Results for PROC CONTENTS

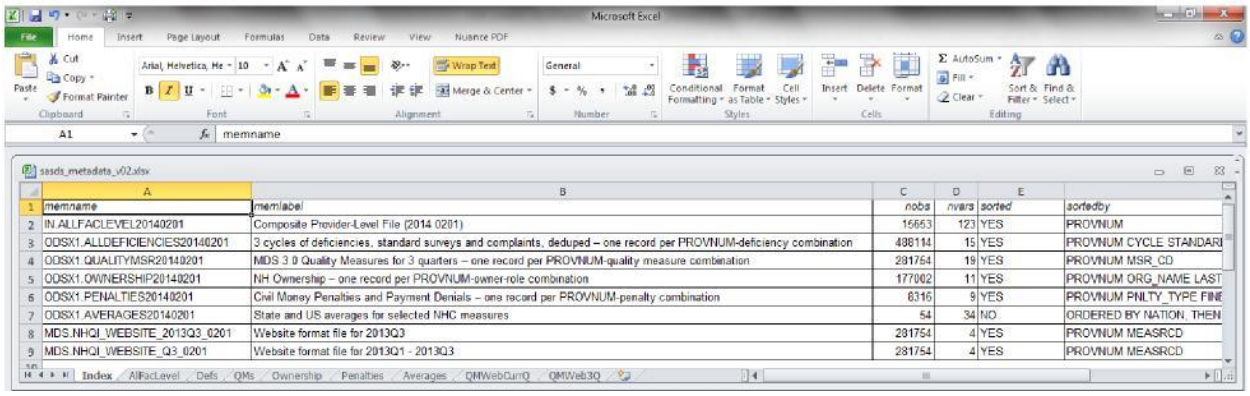
The ATTRIBUTES, VARIABLES, and SORTEDBY output objects were used to construct two designer data sets for each data set to be documented. Note that because one of the input data bases is not sorted, conditional macro processing was used to create SORTEDBY if it does not exist.

```
/* first, set up a macro to (1) collect header level information for each DS
and (2) collect variable level information for each ds */

%MACRO cont2(runnum,inlib,infi,headfi);
ODS OUTPUT ATTRIBUTES=atr VARIABLES=var SORTEDBY=sortedby;
PROC CONTENTS DATA=&inlib..&infi ; RUN;
PROC SORT DATA=var; BY num;
RUN;
DATA &infi;
SET var (drop=member pos);
```

```
RENAME num=varnum
variable=name len=length;
RUN;
%IF %SYSFUNC(EXIST(sortedby)) %then %do;
/* if it exists, go ahead and make our day */
DATA sortvar (KEEP=sortedby);
LENGTH sortedby $ 1000;
SET sortedby (WHERE=(label1='Sortedby'));
sortedby=cvalue1;
RUN;
%END;
%ELSE %DO; /* if it doesn't exist, we make it */
DATA sortvar;
LENGTH sortedby $ 1000;
sortedby='Ordered by Nation, then State';
RUN; %END;
DATA sorted (KEEP=sorted);
SET atr (WHERE=(label1='Data Set Type')); sorted=cvalue2;
RUN;
DATA memlabel (KEEP=memlabel);
SET atr (WHERE=(label1='Label')); memlabel=cvalue1;
RUN;
data nvars (KEEP=nvars);
SET atr (WHERE=(label1='Member Type'));
nvars=INPUT(cvalue2,11.);
FORMAT nvars comma11.;
RUN;
data member (KEEP=memname nobs);
SET atr (WHERE=(label1='Data Set Name'));
memname=cvalue1;
nobs=INPUT(cvalue2,11.);
FORMAT nobs comma11.;
RUN;
/* note lack of merge by variable deliberate */ DATA &headfi;
MERGE member memlabel nvars sorted sortvar; RUN;
%MEND cont2;
%cont2(1,in,allfaclevel&fileyear.&filedate.,header1);
%cont2(3,ODSx1,qualitymsr&fileyear.&filedate.,header3);
%cont2(2,ODSx1,alldeficiencies&fileyear.&filedate.,header2);
%cont2(4,ODSx1,ownership&fileyear.&filedate.,header4);
%cont2(5,ODSx1,penalties&fileyear.&filedate.,header5);
%cont2(6,ODSx1,averages&fileyear.&filedate.,header6);
%cont2(7,mds,nhqi_website_&QMlastQ._&filedate.,header7);
%cont2(8,mds,nhqi_website_q3_&filedate.,header8);
/* now build the header file */
DATA index;
LENGTH memlabel sortedby $ 120 sorted $ 3; SET header1-header8;
RUN;
/* continued */
```

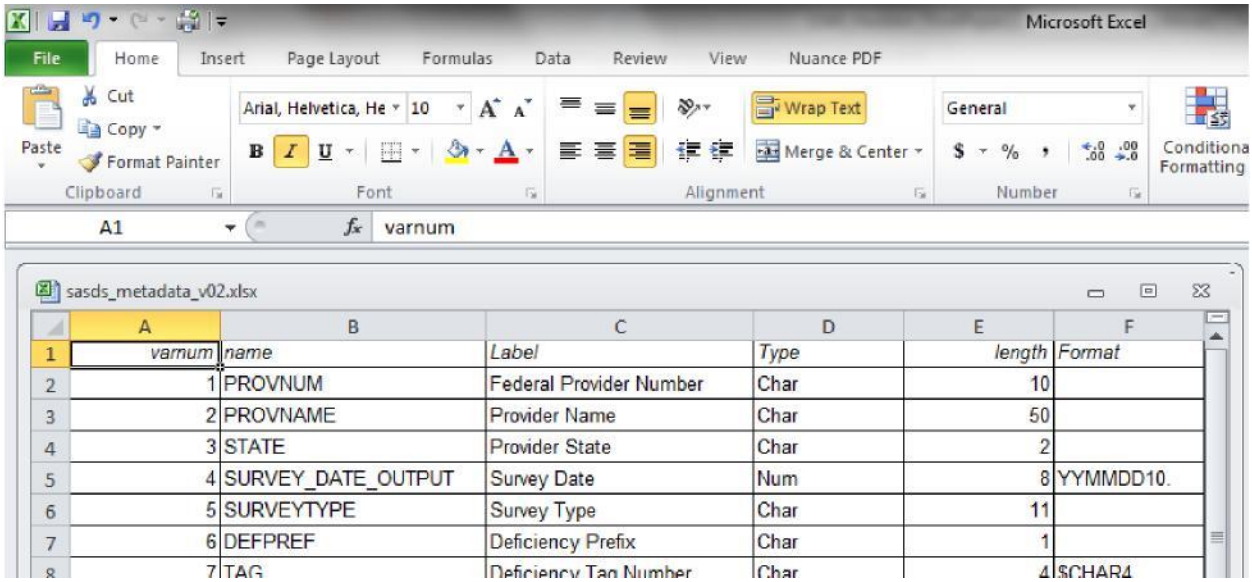
Once the header or index file and individual data set variable listings have been built, a multi-tabbed Excel workbook is created, via ODS.TAGSETS.EXCELXP pre Version 9.4 M4 and the ODS EXCEL destination for M4 and later. ODS gives more control over formatting than a simple PROC EXPORT. We can name our tabs and set column widths within SAS for individual worksheets using options: sheet_name names each tab, and Absolute_Column_Width sets the width for each column. We set and unset the sheet

interval as table to generate a tab for each report.



	A	B	C	D	E	
1	memname	memlabel	nobs	nvars	sorted	sortedby
2	IN.ALLFACLEVEL20140201	Composite Provider-Level File (2014 Q201)	15653	123	YES	PROVNUM
3	ODSX1.ALLDEFICIENCIES20140201	3 cycles of deficiencies, standard surveys and complaints, deduped – one record per PROVNUM-deficiency combination	488114	15	YES	PROVNUM CYCLE STANDARI
4	ODSX1.QUALITYMSR20140201	MDS 3.0 Quality Measures for 3 quarters – one record per PROVNUM-quality measure combination	281754	19	YES	PROVNUM MSR_CD
5	ODSX1.OWNERSHIP20140201	NH Ownership – one record per PROVNUM-owner-role combination	177002	11	YES	PROVNUM ORG_NAME LAST
6	ODSX1.PENALTIES20140201	Civil Money Penalties and Payment Denials – one record per PROVNUM-penalty combination	6316	9	YES	PROVNUM PNLTY_TYPE FINE
7	ODSX1.AVERAGES20140201	State and US averages for selected NHC measures	54	34	NO	ORDERED BY NATION, THEN
8	MDS.NHQI_WEBSITE_2013Q3_0201	Website format file for 2013Q3	281754	4	YES	PROVNUM MEASRCD
9	MDS.NHQI_WEBSITE_Q3_0201	Website format file for 2013Q1 - 2013Q3	281754	4	YES	PROVNUM MEASRCD

Figure 6. Screenshot of the Index Tab of the Metadata Spreadsheet



	A	B	C	D	E	F
1	varnum	name	Label	Type	length	Format
2	1	PROVNUM	Federal Provider Number	Char	10	
3	2	PROVNAME	Provider Name	Char	50	
4	3	STATE	Provider State	Char	2	
5	4	SURVEY_DATE_OUTPUT	Survey Date	Num	8	YYMMDD10.
6	5	SURVEYTYPE	Survey Type	Char	11	
7	6	DEFPREF	Deficiency Prefix	Char	1	
8	7	TAG	Deficiency Tag Number	Char	4	$CHAR4

Figure 7. Screenshot of the Defs Tab of the Metadata Spreadsheet

CREATE A TABLE FOR A JOURNAL ARTICLE

The example shown below manipulates ODS output objects including Chi Squared statistics as well as frequency output from the PROC SURVEYFREQ procedure run on a large complex data set. No post-processing was required.

Table 1. Selected Characteristics by BMI Category

Characteristic	Total 2005	Total 2005	Chi Squared P-Value	Underweight (BMI<18.5)	Normal Weight (BMI 18.5-24.9)	Overweight (BMI 25-29.9)	Obese (BMI 30+)
Total Sample	**15,195 (946,008)**	**100.0±0.00**	**NA**	**1.2±0.14**	**38.3±0.82**	**47.6±0.68**	**12.9±0.55**
Sex							
Men	11,395 (804,888)	85.1±0.71	<.0001	0.9±0.16	35.2±0.94	50.2±0.79	13.7±0.61
Women	3,800 (141,120)	14.9±0.71	<.0001	2.8±0.30	56.1±1.24	32.9±1.01	8.2±0.63
Age							
17-20	1,187 (130,680)	13.8±0.99	<.0001	1.6±0.40	53.8±1.95	37.9±1.78	6.7±1.35
21-30	6,180 (481,590)	50.9±1.21	<.0001	1.5±0.21	42.4±0.98	44.7±0.96	11.4±0.58
31-39	4,610 (230,112)	24.3±1.05	<.0001	0.6±0.17	26.9±1.22	54.8±0.91	17.7±0.85
40+	3,218 (103,627)	11.0±0.75	<.0001	0.7±0.20	25.2±1.15	57.2±1.22	16.9±0.98
Educational Attainment							
High School or Less	4,409 (352,500)	37.3±1.56	<.0001	1.1±0.26	43.9±1.26	41.8±0.95	13.3±0.93
Some College	6,146 (380,144)	40.2±1.26	<.0001	1.6±0.20	34.5±0.80	49.7±0.79	14.2±0.67

Figure 8. Journal article table

MANIPULATING PROCEDURAL OUTPUT

There are a number of procedures that can generate output data sets outside of ODS, and at least one that ONLY generates an output data set (PROC SUMMARY). By exploring the structure and content of these data sets, we can manipulate them to present specific statistics, etc. If the output data sets are made permanent, the possibilities for different presentation styles and output types are endless.

An example of using PROC SUMMARY output follows. PROC SUMMARY was chosen to demonstrate using procedural output because it does not produce any "listing" output. The flexibility offered by ODS output objects is a much better technique for selective presentation for most procedures, as in the example of PROC UNIVARIATE, which can output procedural output as well as (more) ODS output objects.

As in the ODS OUTPUT examples, the rather prosaic tools we'll use to explore our output file are a PROC PRINT (limit to obs=10 if the output file is large) and a PROC CONTENTS. The task is easier in a way in that we already know the name and location of our output file.

PROC SUMMARY example 1

```
proc summary data=bmi nway;
class sex age;
var bmi;
output out=bmisum1 mean=mean_bmi ;
run;
```

PROC CONTENTS of BMI Summary File 1

Variables in Creation Order

#	Variable	Type	Len	Label
1	Sex	Char	1	Gender
2	Age	Num	8	Age
3	_TYPE_	Num	8	
4	_FREQ_	Num	8	
5	mean_bmi	Num	8	BMI

Following a review of the output, you may want to relabel your output and perhaps get rid of and/or rename the _type_ and _freq_ variables.

PROC SUMMARY example 2

```
proc summary data=bmi nway;
class sex age;
var bmi;
output out=bmisum2 (drop=_:)
    idgroup (min(bmi) out(bmi bmicat)=minbmi minbmicat)
    idgroup (max(bmi) out(bmi bmicat)=maxbmi maxbmicat)
    n=n median=median mean=mean std=std;
run;
```

PROC CONTENTS of BMI Summary File 2

Variables in Creation Order

#	Variable	Type	Len	Label
1	Sex	Char	1	Gender
2	Age	Num	8	Age
3	minbmi	Num	8	BMI
4	minbmicat	Num	8	BMI Category (1-3)
5	maxbmi	Num	8	BMI
6	maxbmicat	Num	8	BMI Category (1-3)
7	n	Num	8	BMI
8	median	Num	8	BMI
9	mean	Num	8	BMI
10	std	Num	8	BMI

Notice that multiple variables have the label BMI and that the BMICAT variables also have the originating variable label. You definitely want to relabel the variables in this case. Note also that we dropped the automatic variables _TYPE_ and _FREQ_ in the output statement.

```
/* format, reorder and relabel for printing */

data printsum2;
    retain sex age n minbmi minbmicat mean median maxbmi maxbmicat std;
    set bmisum2;
    label minbmi='Minimum BMI for Sex and Age'
          minbmicat='Minimum BMI Category for Sex and Age'
          maxbmi='Maximum BMI for Sex and Age'
          maxbmicat='Maximum BMI Category for Sex and Age'
          n='# of Obs for Sex and Age'
          mean='Average BMI for Sex and Age'
          median='Median BMI for Sex and Age'
          std='Standard Deviation of BMI for Sex and Age';
run;
```

Final Print of BMI Summary File 2

Gender	Age	# of Obs for Sex and Age	Minimum BMI for Sex and Age	Minimum BMI Category for Sex and Age	Average BMI for Sex and Age	Median BMI for Sex and Age	Maximum BMI for Sex and Age	Maximum BMI Category for Sex and Age	Standard Deviation of BMI for Sex and Age
F	11	4	13.4900	1	28.5160	30.1855	40.2029	3	11.4052
F	12	8	16.6115	1	28.6114	29.7613	39.2565	3	8.3448
F	13	8	16.1568	1	28.1582	29.0156	40.5206	3	8.2236
F	14	8	15.3030	1	26.5825	27.6195	36.0962	2	7.1255
F	15	8	17.8045	1	28.5581	29.4860	38.2432	2	6.9855
M	11	4	18.0733	1	27.3758	26.0469	39.3361	3	9.0783
M	12	12	17.7715	1	28.2744	27.0462	40.2897	3	7.5969
M	13	4	15.1173	1	22.9909	21.8661	33.1141	3	7.6838
M	14	8	16.6115	1	24.2847	23.1117	35.3047	3	6.5163
M	15	8	17.8045	1	26.2197	25.2483	36.4890	3	6.4296
M	16	4	20.3414	1	26.2744	25.4268	33.9024	3	5.7899

CONCLUSION

The examples shown above are fairly simplistic but the basic concepts remain the same whether you are doing a complex weighted multivariate analysis or a simple PROC FREQ. Using ODS output objects and output data sets to create highly customized output can be an enormous time saver. With ODS output objects / output data sets and SAS, the sky is the limit!

REFERENCES

Carey, Helen and Carey, Ginger, 2011. "Tips and Techniques for the SAS Programmer!" Proceedings of SAS Global Forum 2011.

Crawford, Peter, 2013. "A Day in the Life of Data – Part 3." Proceedings of SAS Global Forum 2013.

Fraeman, Kathy Hardis, 2008. "Get into the Groove with %SYSFUNC: Generalizing SAS® Macros with Conditionally Executed Code." Proceedings of NESUG 2008.

Freedman DS, Wang J, Thornton JC, et al. Classification of body fatness by body mass index-for age categories among children. Arch Pediatr Adolesc Med. 2009; 163(9):805–811.

Guan, Calvin and Hadden, Louise. "EXCELing in DDE: Unlock Useful Tools for Processing Excel®". Proceedings of NorthEast SAS Users Group 2013 Conference. September, 2013.

Hadden, Louise, 2014. "Build your Metadata with PROC CONTENTS and ODS OUTPUT", Proceedings of SAS Global Forum 2014.

Hadden, Louise. "With a Trace: Making Procedural Output and ODS Output Objects Work For You". Proceedings of SAS Global Forum 2013 Conference. May 2013.

Huang, Chao, 2014. "Top 10 SQL Tricks in SAS®." Proceedings of SAS Global Forum 2014.

Hughes, Troy Martin and Hadden, Louise. "DOMinate your ODS Output with PROC TEMPLATE, ODS Cascading Style Sheets (CCS), and the ODS Document Object Model (DOM) by Mastering ODS TRACE DOM, Parsing CSS with a PROC TEMPLATE Front End, and Overriding ODS Style Inheritance with Customized Styles". Proceedings of PharmaSUG 2019. June 2019.

Kalt, Mike and Zender, Cynthia. "Introduction to ODS Graphics for the Non-Statistician". Proceedings of SAS Global Forum 2011 Conference. April 2011.

Karafa, Matthew T., 2012. "Macro Coding Tips and Tricks to Avoid "PEBCAK" Errors." Proceedings of SAS Global Forum 2012.

King, John and Zdeb, Mike. "Transposing Data Using PROC SUMMARY'S IDGROUP Option." Proceedings of SAS Global Forum 2010. April 2010.

Kuligowski, Andrew T. and Shankar, Charu, 2013. "Know Thy Data: Techniques for Data Exploration." Proceedings of SAS Global Forum 2013.

Lafler, Kirk Paul, 2014. "Powerful and Hard-to-find PROC SQL Features." Proceedings of SAS Global Forum 2014.

Murphy, William C., 2013. "What's in a SAS® Variable? Get Answers with a V!" Proceedings of SAS Global Forum 2013.

Raithel, Michael A., 2011. "PROC DATASETS: the Swiss Army Knife of SAS® Procedures." Proceedings of SAS Global Forum 2011.

Thornton, Patrick, 2011. "SAS® DICTIONARY: Step by Step." Proceedings of SAS Global Forum 2011.

Williams, Christianna. "Any WAY you Want it: Getting the Right TYPEs of Observations out of PROC SUMMARY or MEANS." Proceedings of SAS Global Forum 2008 Conference. April 2008.

Zhang, Jingxian, 2012. "Techniques for Generating Dynamic Code from SAS® Dictionary Tables." Proceedings of SAS Global Forum 2012.

Zender, Cynthia. "The Greatest Hits: ODS Essentials Every User Should Know". Proceedings of SAS Global Forum 2011 Conference. April 2011.

CONTACT INFORMATION

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

Louise S. Hadden louise_hadden@abtassoc.com

Full code examples are available upon request.

Appendix 1: Sample Code to Explore ODS Output Objects

Cc3_batch.sas

```
options ps=55 ls=175 errorabend errors=1 nofmterr nodate nonumber;

libname dd '.';
filename odsout '.';
libname library '.';
run;

ods noproctitle;

title1 'CC3:  With a Trace';
run;

proc format;
    value bmicatf 1='Normal'
                  2='Moderate'
                  3='Elevated';
run;

/* create a bmi data set */

data bmi;
    set sashelp.class sashelp.class (in=a) sashelp.class (in=b) sashelp.class
(in=c);

    if a and sex='M' then weight=weight+25;
    if b and sex='M' then weight=weight+50;
    if a and sex='F' then weight=weight+50;
    if b and sex='F' then weight=weight+100;
    if c and sex='M' then weight=weight+100;
    if c and sex='F' then weight=weight+75;

    if height ne . and weight ne . then BMI = ( weight / (height*height) ) *
703;

    if bmi ne . then do;
        if sex='M' then do;
            if (age < 9 and bmi < 22)
            OR (9 le age le 11.9 and bmi < 24)
            OR (12 le age le 14.9 and bmi < 23)
            OR (age ge 15 and bmi < 22) then bmicat=1;
            if (age < 9 and 22 le bmi le 26)
            OR (9 le age le 11.9 and 24 le bmi le 34)
            OR (12 le age le 14.9 and 23 le bmi le 32)
            OR (age ge 15 and 22 le bmi le 29) then bmicat=2;
            if (age < 9 and bmi < 26)
            OR (9 le age le 11.9 and bmi > 34)
            OR (12 le age le 14.9 and bmi > 32)
            OR (age ge 15 and bmi > 29) then bmicat=3;
        end;
        if sex='F' then do;
            if (age < 9 and bmi < 27)
            OR (9 le age le 11.9 and bmi < 30)
            OR (12 le age le 14.9 and bmi < 32)
```

```
            OR (age ge 15 and bmi < 36) then bmicat=1;
            if (age < 9 and 27 le bmi le 34)
            OR (9 le age le 11.9 and 30 le bmi le 37)
            OR (12 le age le 14.9 and 32 le bmi le 39)
            OR (age ge 15 and 36 le bmi le 42) then bmicat=2;
            if (age < 9 and bmi < 34)
            OR (9 le age le 11.9 and bmi > 37)
            OR (12 le age le 14.9 and bmi > 39)
            OR (age ge 15 and bmi > 42) then bmicat=3;
        end;
    end;

    if bmicat ne . then elevated=(bmicat=3);
    if bmicat ne . then moderate=(bmicat=2);

    if sex ne '' then male=(upcase(sex)='M');

    label bmi='BMI'
          bmicat='BMI Category (1-3)'
          elevated='Binary for BMI=Elevated'
          moderate='Binary for BMI=Moderate'
          male='Binary for Sex=Male'
          age='Age'
          sex='Gender'
          name='First Name'
          height='Height in Inches'
          weight='Weight in Pounds';
run;

ods trace on / label;

ods rtf file='cc3_batch.rtf' path=odsout style=styles.journal2;

proc contents data=bmi varnum;
title2 'Contents';
run;

proc means data=bmi;
title2 'Means';
run;

proc print data=bmi (obs=10) noobs;
title2 'Test Print 10 Observations';
run;

ods graphics on;

proc univariate data=bmi;
    var bmi;
    histogram bmi / normal;
title2 'Proc Univariate with Histogram';
run;

proc freq data=bmi;
    tables sex*bmicat;
format bmicat bmicatf.;
title2 'Crosstab Sex and BMI Category';
```

```
run;

    ods output
    nobs=nobs
    parameterestimates=parameterestimates
    oddsratios=oddsratios;

proc logistic data=bmi descending;
    model moderate=male height weight age;
title2 'Logistic - Moderate BMI as dependent variable';
run;

ods output close;

proc ttest data=bmi;
    var bmi height weight age bmicat;
    class male;
title2 'Ttest';
run;

proc sgplot data=bmi;
    scatter y=bmi x=age ;
title2 'SG Plot';
run;

ods rtf close;

ods graphics off;

ods trace off;



ods rtf file='cc3_batch_ods_output.rtf' path=odsout style=styles.journal;


proc print data=nobs (obs=10) noobs;
title2 'Test NOBS';
run;

proc contents data=nobs varnum;
run;

proc print data=parameterestimates (obs=10) noobs;
title2 'Test Parameter Estimates';
run;

proc contents data=parameterestimates varnum;
run;

proc print data=oddsratios (obs=10) noobs;
title2 'Test Odds Ratios';
run;

proc contents data=oddsratios varnum;
run;
```

```
ods rtf close;

data lhs;
    length rowdesc $ 32;
    rowcat=1; rownum=1; rowdesc='Binary: Male'; output;
    rowcat=1; rownum=2; rowdesc='Height in Inches'; output;
    rowcat=1; rownum=3; rowdesc='Weight in Pounds'; output;
    rowcat=1; rownum=4; rowdesc='Age in Years'; output;
run;

data a (drop=variable);
    length description $ 32;
    set parameterestimates (where=(variable ne 'Intercept'));
    counter=_n_+4;
    description=variable;
run;
proc print data=a;
run;


data b (drop=effect);
    length description $ 50;
    set oddsratios ;
    counter=_n_+4;
    description=effect;
run;
proc print data=b;
run;

data d;
    length description $ 50;
    merge a b;
    by counter;
run;

proc print data=d;
run;

data c (drop=label);
    length description $ 50;
    set nobs (keep=label n);
    description=label;
    counter=_n_;
run;

proc print data=c;
run;


data editoutput;
    set d;
if description= 'male' then do; rowcat=1; rownum=1; end;
if description= 'Height' then do; rowcat=1; rownum=2; end;
if description= 'Weight' then do; rowcat=1; rownum=3; end;
if description= 'Age' then do; rowcat=1; rownum=4; end;

run;
```

```
proc sort data=editoutput;
    by rowcat rownum;
run;

proc sort data=lhs;
    by rowcat rownum;
run;

data printoutput;
    merge editoutput lhs;
    by rowcat rownum;
run;

ods html file="cc3_batch_output.xls" path=odsout style=styles.journal;

proc print data=printoutput noobs;
    var rowcat rownum description rowdesc oddsratioest probchisq;
title2 'Logistic Manipulated Output';
run;

ods html close;

/* now proc summary */

proc summary data=bmi nway;
    class sex age;
    var bmi;
    output out=bmisum1 mean=mean_bmi ;
run;

ods rtf file='testsummary1.rtf' path=odsout style=styles.journal2;

proc print data=bmisum1 noobs;
title2 'Test PROC SUMMARY 1';
run;

proc contents data=bmisum1 varnum;
run;

ods rtf close;

/* format a little for printing */

data printsum1;
    set bmisum1 (drop=_type_ rename=(_freq_=n_obs));
    label n_obs='# of Obs'
          mean_bmi='Mean BMI';
run;

ods rtf file='bmisummary1.rtf' path=odsout style=styles.journal2;

proc print data=printsum1 label uniform noobs;
run;

ods rtf close;
```

```
proc summary data=bmi nway;
    class sex age;
    var bmi;
    output out=bmisum2 (drop=_:)
        idgroup (min(bmi) out(bmi bmicat)=minbmi minbmicat)
        idgroup (max(bmi) out(bmi bmicat)=maxbmi maxbmicat)
        n=n median=median mean=mean std=std;
run;

ods rtf file='testsummary2.rtf' path=odsout style=styles.journal2;

proc print data=bmisum2 noobs;
title2 'Test PROC SUMMARY 2';
run;

proc contents data=bmisum2 varnum;
run;

ods rtf close;

/* format, reorder and relabel for printing */

data printsum2;
    retain sex age n minbmi minbmicat mean median maxbmi maxbmicat std;
    set bmisum2;
    label minbmi='Minimum BMI for Sex and Age'
          minbmicat='Minimum BMI Category for Sex and Age'
          maxbmi='Maximum BMI for Sex and Age'
          maxbmicat='Maximum BMI Category for Sex and Age'
          n='# of Obs for Sex and Age'
          mean='Average BMI for Sex and Age'
          median='Median BMI for Sex and Age'
          std='Standard Deviation of BMI for Sex and Age';
run;

ods rtf file='bmisummary2.rtf' path=odsout style=styles.journal2;

proc print data=printsum2 label uniform noobs;
run;

ods rtf close;
```

cc3_inter.sas

```
options ps=55 ls=175 errors=1 nofmterr nodate nonumber;

/* note, depending on your SAS set up you may need to replace the dot below
with full pathnames */

libname dd '.';
filename odsout '.';
libname library '.';
run;

ods noproctitle;

title1 'CC3:  With a Trace';
run;

proc format;
    value bmicatf 1='Normal'
                  2='Moderate'
                  3='Elevated';
run;

/* create a bmi data set */

data bmi;
    set sashelp.class sashelp.class (in=a) sashelp.class (in=b) sashelp.class
(in=c);

    if a and sex='M' then weight=weight+25;
    if b and sex='M' then weight=weight+50;
    if a and sex='F' then weight=weight+50;
    if b and sex='F' then weight=weight+100;
    if c and sex='M' then weight=weight+100;
    if c and sex='F' then weight=weight+75;

    if height ne . and weight ne . then BMI = ( weight / (height*height) ) *
703;

    if bmi ne . then do;
        if sex='M' then do;
            if (age < 9 and bmi < 22)
            OR (9 le age le 11.9 and bmi < 24)
            OR (12 le age le 14.9 and bmi < 23)
            OR (age ge 15 and bmi < 22) then bmicat=1;
            if (age < 9 and 22 le bmi le 26)
            OR (9 le age le 11.9 and 24 le bmi le 34)
            OR (12 le age le 14.9 and 23 le bmi le 32)
            OR (age ge 15 and 22 le bmi le 29) then bmicat=2;
            if (age < 9 and bmi < 26)
            OR (9 le age le 11.9 and bmi > 34)
            OR (12 le age le 14.9 and bmi > 32)
            OR (age ge 15 and bmi > 29) then bmicat=3;
        end;
        if sex='F' then do;
            if (age < 9 and bmi < 27)
            OR (9 le age le 11.9 and bmi < 30)
```

```
            OR (12 le age le 14.9 and bmi < 32)
            OR (age ge 15 and bmi < 36) then bmicat=1;
            if (age < 9 and 27 le bmi le 34)
            OR (9 le age le 11.9 and 30 le bmi le 37)
            OR (12 le age le 14.9 and 32 le bmi le 39)
            OR (age ge 15 and 36 le bmi le 42) then bmicat=2;
            if (age < 9 and bmi < 34)
            OR (9 le age le 11.9 and bmi > 37)
            OR (12 le age le 14.9 and bmi > 39)
            OR (age ge 15 and bmi > 42) then bmicat=3;
        end;
    end;

    if bmicat ne . then elevated=(bmicat=3);
    if bmicat ne . then moderate=(bmicat=2);

    if sex ne '' then male=(upcase(sex)='M');

    label bmi='BMI'
          bmicat='BMI Category (1-3)'
          elevated='Binary for BMI=Elevated'
          moderate='Binary for BMI=Moderate'
          male='Binary for Sex=Male'
          age='Age'
          sex='Gender'
          name='First Name'
          height='Height in Inches'
          weight='Weight in Pounds';
run;

ods trace on / label;

ods rtf file='cc3_inter.rtf' path=odsout style=styles.journal2;

proc contents data=bmi varnum;
title2 'Contents';
run;

proc means data=bmi;
title2 'Means';
run;

proc print data=bmi (obs=10) noobs;
title2 'Test Print 10 Observations';
run;

ods graphics on;

proc univariate data=bmi;
    var bmi;
    histogram bmi / normal;
title2 'Proc Univariate with Histogram';
run;

proc freq data=bmi;
    tables sex*bmicat;
format bmicat bmicatf.;
```

```
title2 'Crosstab Sex and BMI Category';
run;

    ods output
    nobs=nobs
    parameterestimates=parameterestimates
    oddsratios=oddsratios;

proc logistic data=bmi descending;
    model moderate=male height weight age;
title2 'Logistic - Moderate BMI as dependent variable';
run;

ods output close;

proc ttest data=bmi;
    var bmi height weight age bmicat;
    class male;
title2 'Ttest';
run;


ods listing sge=on;

proc sgplot data=bmi;
    scatter y=bmi x=age ;
title2 'SG Plot';
run;

ods rtf close;

ods graphics off;

ods trace off;

ods rtf file='cc3_inter_ods_output.rtf' path=odsout style=styles.journal;


proc print data=nobs (obs=10) noobs;
title2 'Test NOBS';
run;

proc contents data=nobs varnum;
run;

proc print data=parameterestimates (obs=10) noobs;
title2 'Test Parameter Estimates';
run;

proc contents data=parameterestimates varnum;
run;

proc print data=oddsratios (obs=10) noobs;
title2 'Test Odds Ratios';
run;

proc contents data=oddsratios varnum;
```

```
run;

ods rtf close;

data lhs;
    length rowdesc $ 32;
    rowcat=1; rownum=1; rowdesc='Binary: Male'; output;
    rowcat=1; rownum=2; rowdesc='Height in Inches'; output;
    rowcat=1; rownum=3; rowdesc='Weight in Pounds'; output;
    rowcat=1; rownum=4; rowdesc='Age in Years'; output;
run;

data a (drop=variable);
    length description $ 32;
    set parameterestimates (where=(variable ne 'Intercept'));
    counter=_n_+4;
    description=variable;
run;
proc print data=a;
run;


data b (drop=effect);
    length description $ 50;
    set oddsratios ;
    counter=_n_+4;
    description=effect;
run;
proc print data=b;
run;

data d;
    length description $ 50;
    merge a b;
    by counter;
run;

proc print data=d;
run;

data c (drop=label);
    length description $ 50;
    set nobs (keep=label n);
    description=label;
    counter=_n_;
run;

proc print data=c;
run;


data editoutput;
    set d;
if description= 'male' then do; rowcat=1; rownum=1; end;
if description= 'Height' then do; rowcat=1; rownum=2; end;
if description= 'Weight' then do; rowcat=1; rownum=3; end;
if description= 'Age' then do; rowcat=1; rownum=4; end;
```

```
run;

proc sort data=editoutput;
    by rowcat rownum;
run;

proc sort data=lhs;
    by rowcat rownum;
run;

data printoutput;
    merge editoutput lhs;
    by rowcat rownum;
run;

ods html file="cc3_inter_output.xls" path=odsout style=styles.journal;

proc print data=printoutput noobs;
    var rowcat rownum description rowdesc oddsratioest probchisq;
title2 'Logistic Manipulated Output';
run;

ods html close;

/* now proc summary */

proc summary data=bmi nway;
    class sex age;
    var bmi;
    output out=bmisum1 mean=mean_bmi ;
run;

ods rtf file='testsummary1_inter.rtf' path=odsout style=styles.journal2;

proc print data=bmisum1 noobs;
title2 'Test PROC SUMMARY 1';
run;

proc contents data=bmisum1 varnum;
run;

ods rtf close;

/* format a little for printing */

data printsum1;
    set bmisum1 (drop=_type_ rename=(_freq_=n_obs));
    label n_obs='# of Obs'
          mean_bmi='Mean BMI';
run;

ods rtf file='bmisummary1_inter.rtf' path=odsout style=styles.journal2;

proc print data=printsum1 label uniform noobs;
run;
```

```
ods rtf close;


proc summary data=bmi nway;
    class sex age;
    var bmi;
    output out=bmisum2 (drop=_:)
        idgroup (min(bmi) out(bmi bmicat)=minbmi minbmicat)
        idgroup (max(bmi) out(bmi bmicat)=maxbmi maxbmicat)
        n=n median=median mean=mean std=std;
run;

ods rtf file='testsummary2_inter.rtf' path=odsout style=styles.journal2;

proc print data=bmisum2 noobs;
title2 'Test PROC SUMMARY 2';
run;

proc contents data=bmisum2 varnum;
run;

ods rtf close;

/* format, reorder and relabel for printing */

data printsum2;
    retain sex age n minbmi minbmicat mean median maxbmi maxbmicat std;
    set bmisum2;
    label minbmi='Minimum BMI for Sex and Age'
          minbmicat='Minimum BMI Category for Sex and Age'
          maxbmi='Maximum BMI for Sex and Age'
          maxbmicat='Maximum BMI Category for Sex and Age'
          n='# of Obs for Sex and Age'
          mean='Average BMI for Sex and Age'
          median='Median BMI for Sex and Age'
          std='Standard Deviation of BMI for Sex and Age';
run;

ods rtf file='bmisummary2_inter.rtf' path=odsout style=styles.journal2;

proc print data=printsum2 label uniform noobs;
run;

ods rtf close;
```