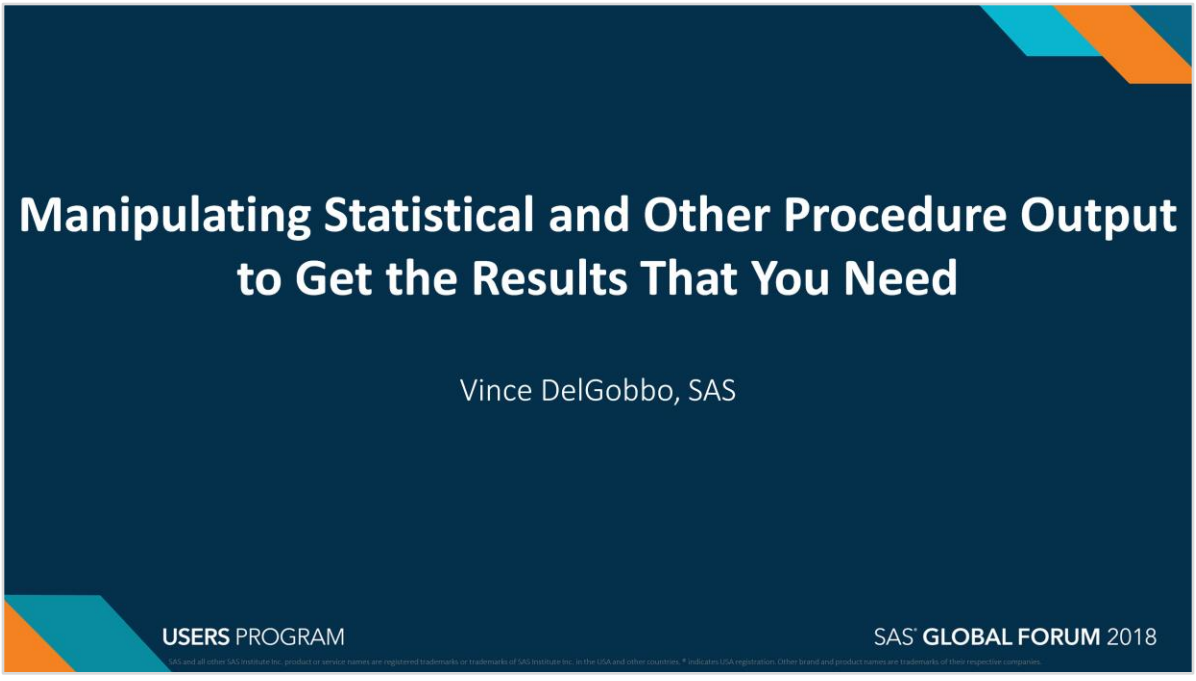


SAS Global Forum 2018
April 8-11, 2018
Colorado Convention Center
Denver, CO

A special "thank you" to Nancy Brucken and Fei Wang for inviting me to present this topic, and to John King of Ouachita Clinical Data Services, and Chris Barrett and Randy Tobias of SAS Institute Inc. for their valuable contributions to the accompanying paper.

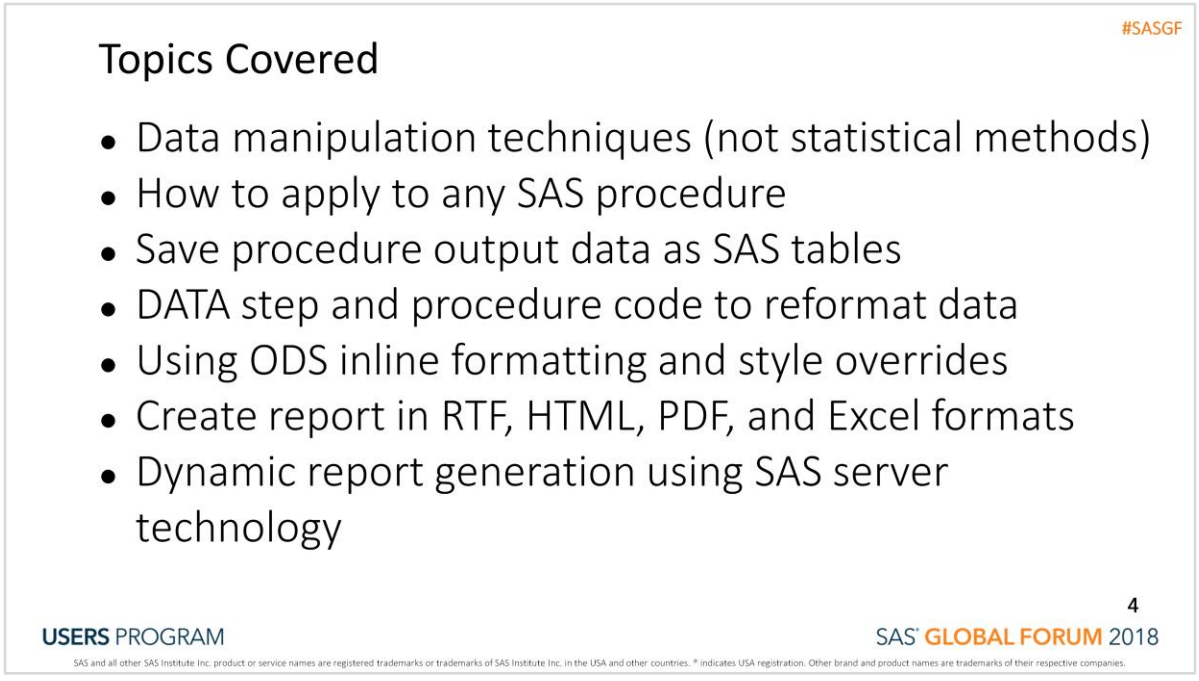
Manipulating Statistical and Other Procedure Output to Get the Results That You Need

Vince DelGobbo, SAS

Vince DelGobbo is a Senior Software Developer in the Metadata and Execution Services group at SAS. This group's responsibilities include the SAS/IntrNet Application Dispatcher and SAS Stored Processes. He is involved in the development of new Web- and server-based technologies, bringing 3rd-party metadata into SAS, and integrating SAS output with Microsoft Office. He was also involved in the early development of the ExcelXP ODS tagset. Vince has been a SAS Software user since 1982, and joined SAS in 1992.

Goals

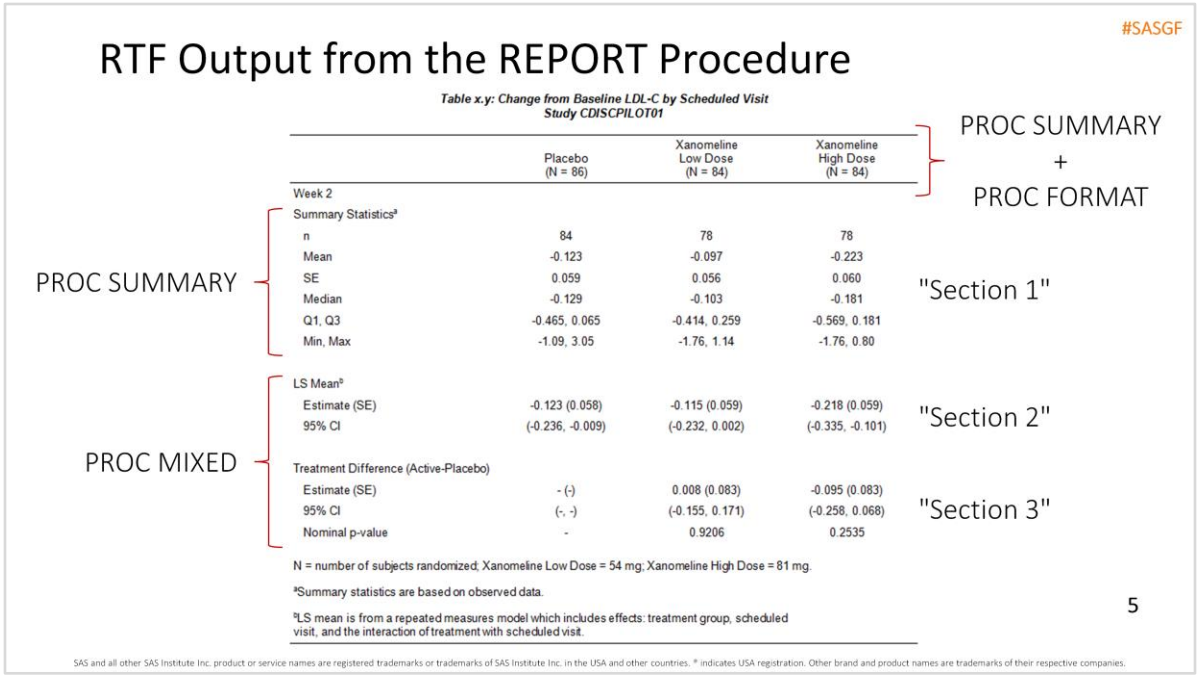
- Save output from any SAS procedure
- Manipulate the output to meet your needs
- Create a report in several formats
- Get you thinking about how to adapt to ***your*** data



This workshop focuses on the data manipulation techniques used to transform SAS® procedure output to conform to any journal or corporate style, rather than on the statistical methods and procedures that produce the data for the report.

You'll learn how to save data from any SAS procedure, change the data format using the DATA step, and dynamically create a format based on your data. You'll also use the SAS Output Delivery System (ODS) inline formatting functions and style overrides, and produce several output formats, including RTF (for Microsoft Word), HTML, PDF, and Microsoft Excel files.

Finally, you'll learn how to create and deliver the output on-demand using SAS server technology.



	Placebo (N = 86)	Xanomeline Low Dose (N = 84)	Xanomeline High Dose (N = 84)
Week 2			
Summary Statistics[a]			
n	84	78	78
Mean	-0.123	-0.097	-0.223
SE	0.059	0.056	0.060
Median	-0.129	-0.103	-0.181
Q1, Q3	-0.465, 0.065	-0.414, 0.259	-0.569, 0.181
Min, Max	-1.09, 3.05	-1.76, 1.14	-1.76, 0.80
LS Mean[b]			
Estimate (SE)	-0.123 (0.058)	-0.115 (0.059)	-0.218 (0.059)
95% CI	(-0.236, -0.009)	(-0.232, 0.002)	(-0.335, -0.101)
Treatment Difference (Active-Placebo)			
Estimate (SE)	- (-)	0.008 (0.083)	-0.095 (0.083)
95% CI	(-, -)	(-0.155, 0.171)	(-0.258, 0.068)
Nominal p-value	-	0.9206	0.2535

N = number of subjects randomized; Xanomeline Low Dose = 54 mg; Xanomeline High Dose = 81 mg.

[a]Summary statistics are based on observed data.

[b]LS mean is from a repeated measures model which includes effects: treatment group, scheduled visit, and the interaction of treatment with scheduled visit.

The report was created using the results from the SAS SUMMARY and MIXED procedures, and is displayed using the REPORT procedure.

The report shows statistics for the change from baseline cholesterol values for two dosages of the drug Xanomeline (54 mg and 81 mg) and for a placebo over time. While this type of output is common for those performing clinical drug trials in the United States, the concepts and techniques that you learn here can be applied to other data and output.

We programmatically create the report without the need for editing, resulting in a process that is accurate and repeatable with different data. The report is categorized into three main sections:

Section 1 - Summary statistics (N, MEAN, MIN, MAX, and so on) created by the SUMMARY procedure.

Section 2 - LS mean standard estimates and confidence intervals, created by the MIXED procedure.

Section 3 - Differences between the two active dosages of the drug and the placebo, created by the MIXED procedure.

#SASGF

Data Used for the Column Headings

Column Name	Type	Label	Values
STUDYID	Character	Study Identifier	CDISCPILOT01
TRT01PN	Numeric	Planned Treatment for Period 01 (N)	0, 54, and 81
TRT01P	Character	Planned Treatment for Period 01	Placebo, Xanomeline Low Dose, and Xanomeline High Dose

6

USERS PROGRAM SAS® GLOBAL FORUM 2018

TRT01PN indicates the treatment that the patient received, either Placebo (0), Xanomeline Low Dose (54), or Xanomeline High Dose (81). There is one record for each subject in the study.

TRT01P is the text description corresponding to TRT01PN.

CDISC Data:

https://www.cdisc.org/pilot-project-submission-package

#SASGF

Data Used for the Report Body

Column Name	Type	Label	Values
TRT01PN	Numeric	Planned Treatment for Period 01 (N)	0, 54, and 81
TRT01P	Character	Planned Treatment for Period 01	Placebo, Xanomeline Low Dose, and Xanomeline High Dose
PARAMCD	Character	Parameter Code	CHOL
USUBJID	Character	Unique Subject Identifier	01-701-1015, 01-701-1023, ...
AVISITN	Numeric	Analysis Visit (N)	2, 4, 6, 8, 12, 16, 20, 24, and 26
CHG	Numeric	Change from Baseline	-1.9395, -0.46548, 1.49988, ...

7

USERS PROGRAM SAS® GLOBAL FORUM 2018

TRT01PN and TRT01P have the same meaning as in the previous table.

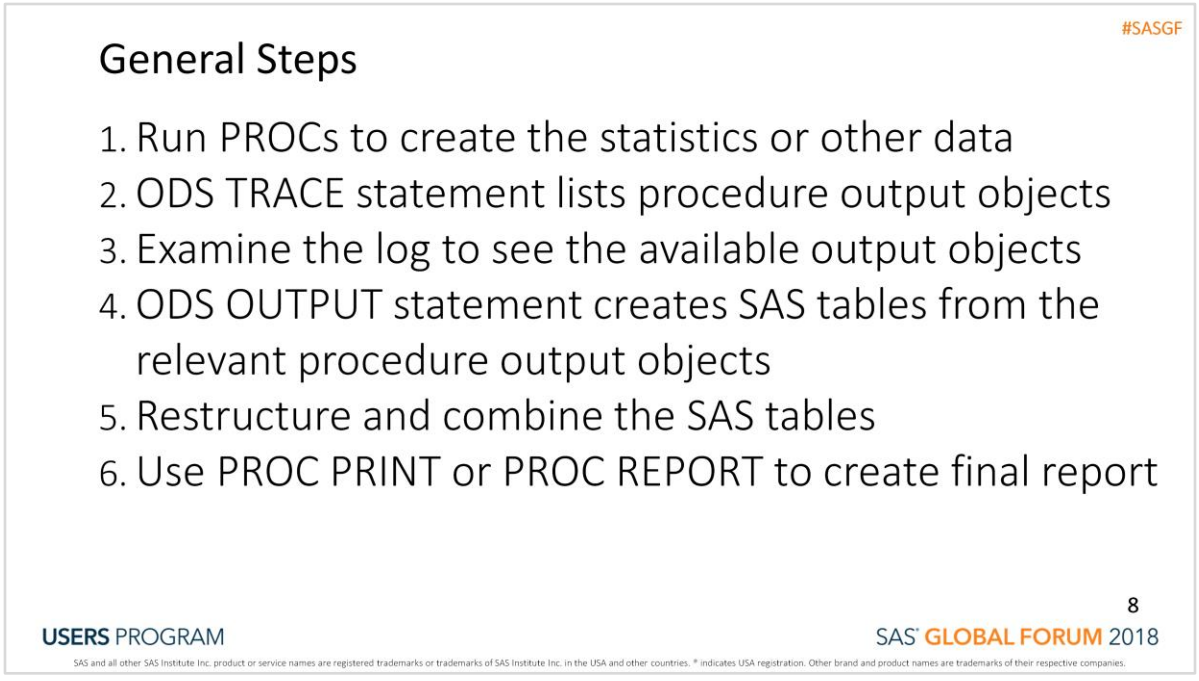
PARAMCD indicates the lab test. Only records for cholesterol results are included.

AVISITN represents the week number when the lab test was performed.

CHG is the change in cholesterol at a given visit, relative to the baseline value.

CDISC Data:

https://www.cdisc.org/pilot-project-submission-package

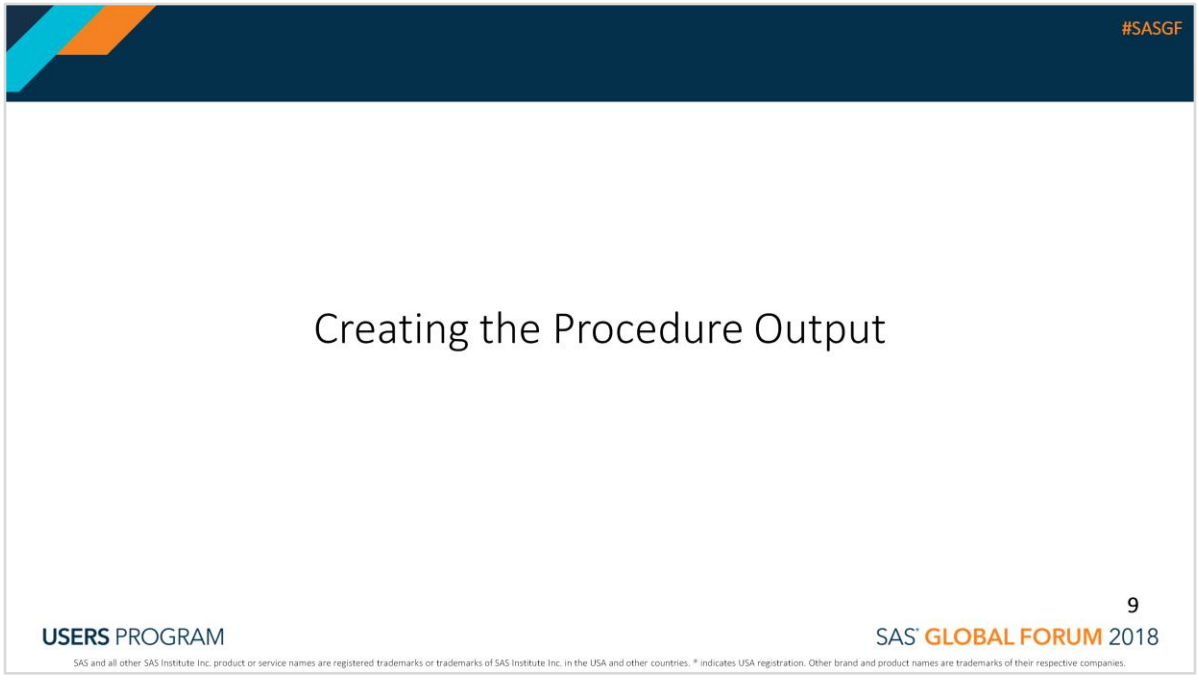


Every procedure creates output objects that contain information for the visual output. Use the ODS TRACE statement to list the output objects.

The ODS OUTPUT statement, explained later, saves the output objects in SAS tables.

Then use DATA step and procedure code to modify the data to get the structure that you need.

Create the final report using PROC PRINT or PROC REPORT with the restructured data, applying ODS inline formatting functions and style overrides to the appearance that you want.


#SASGF
Creating the Procedure Output
9
USERS PROGRAM
SAS GLOBAL FORUM 2018
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.

#SASGF

SAS Code to Create the Statistical Output – 1 of 2

```
* Column Headings;

proc summary print data=sample.adsl;
  class trt01pn trt01p;
run; quit;

* Summary Statistics;

proc summary print data=sample.analysis missing
    stackods n mean stderr median q1 q3 min max;
  class avisitn trt01pn;
  var chg;
run; quit;
```

10

USERS PROGRAM

SAS® GLOBAL FORUM 2018

The first instance of PROC SUMMARY computes the number of subjects for each of the three treatment groups. These values are used in the **N =** text in the column headings.

Running PROC SUMMARY on the ANALYSIS table computes the change from baseline statistics displayed in the **Summary Statistics** section of the report. The statistics are computed for each distinct combination of visit and treatment.

SUMMARY Procedure:

https://goo.gl/z3uRkU

#SASGF

SAS Code to Create the Statistical Output – 2 of 2

```
* LS Means and Active-Placebo Differences;

proc mixed data=sample.analysis;
  class usubjid avisitn trt01pn;
  model chg = avisitn trt01pn avisitn*trt01pn /
              ddfm=kenwardroger;
  repeated avisitn / subject=usubjid type=un;

  lsmeans avisitn*trt01pn / cl;

  slice   avisitn*trt01pn / cl
          diff=control('2' '0') sliceby=avisitn;
run; quit;
```

11

USERS PROGRAM SAS® GLOBAL FORUM 2018

The MIXED procedure computes the statistics used in the **LS Mean** and **Treatment Difference (Active - Placebo)** sections of the report.

The REPEATED statement specifies a repeated measures analysis, with AVISITN as the repeated measures effect and USUBJID as the subject identifier.

The LS means and associated statistics for the **LS Mean** section of the report are computed for each visit-treatment combination.

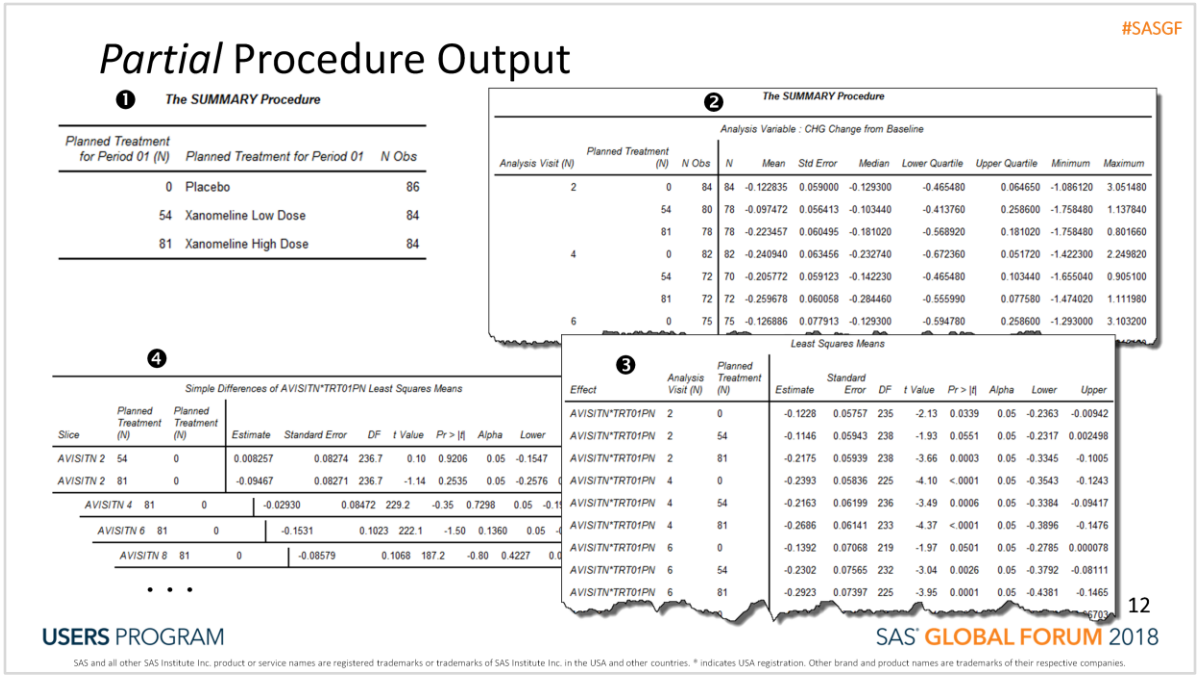
The SLICE statement produces LS mean differences and associated statistics used in the **Treatment Difference (Active - Placebo)** section of the report.

A value for the AVISITN and TRT01PN variables are required in the CONTROL specification, even though the SLICEBY option causes only the value of TRT01PN to be considered. Any value of AVISITN present in the data can be specified.

We specify **'0'** for TRT01PN to indicate that Placebo is taken to be the control for each value of AVISITN.

MIXED Procedure:

https://goo.gl/ioDAcv



Clockwise from upper, left corner:

1. PROC SUMMARY output used for the column headings.
2. PROC SUMMARY output used for the **Summary Statistics** section.
3. PROC MIXED output used for the **LS Means** section.
4. PROC MIXED output used for the **Treatment Difference (Active - Placebo)** section (4 of 9 tables shown).

Additional output, such as model information, covariance parameter estimates, and fit statistics, are not shown.

#SASGF

Set Up the Program Environment

1. Start SAS
2. Click **File > Open Program**
3. Select **Setup.sas** and then click **Open**
4. Review code and then submit

#SASGF

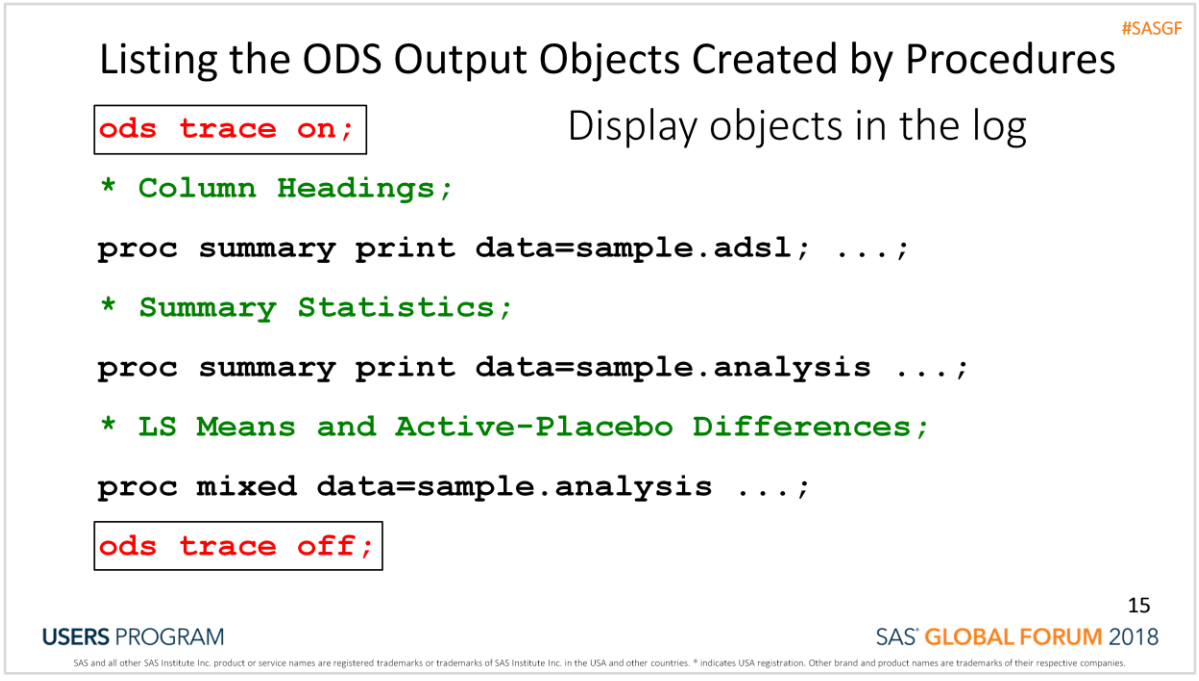
Ex. 1 – Create the Statistical Output

1. Go to SAS
2. Click **File > Open Program > Exercise1.sas**
3. Review code and then submit
4. Review SUMMARY and MIXED procedure output

14

We turn off ODS graphics because we do not need to see the charts produced by the MIXED procedure.

The full output from the SUMMARY and MIXED procedures is stored in the "Statistics.htm" file in our workshop directory. You can refer to this file as needed throughout the workshop.



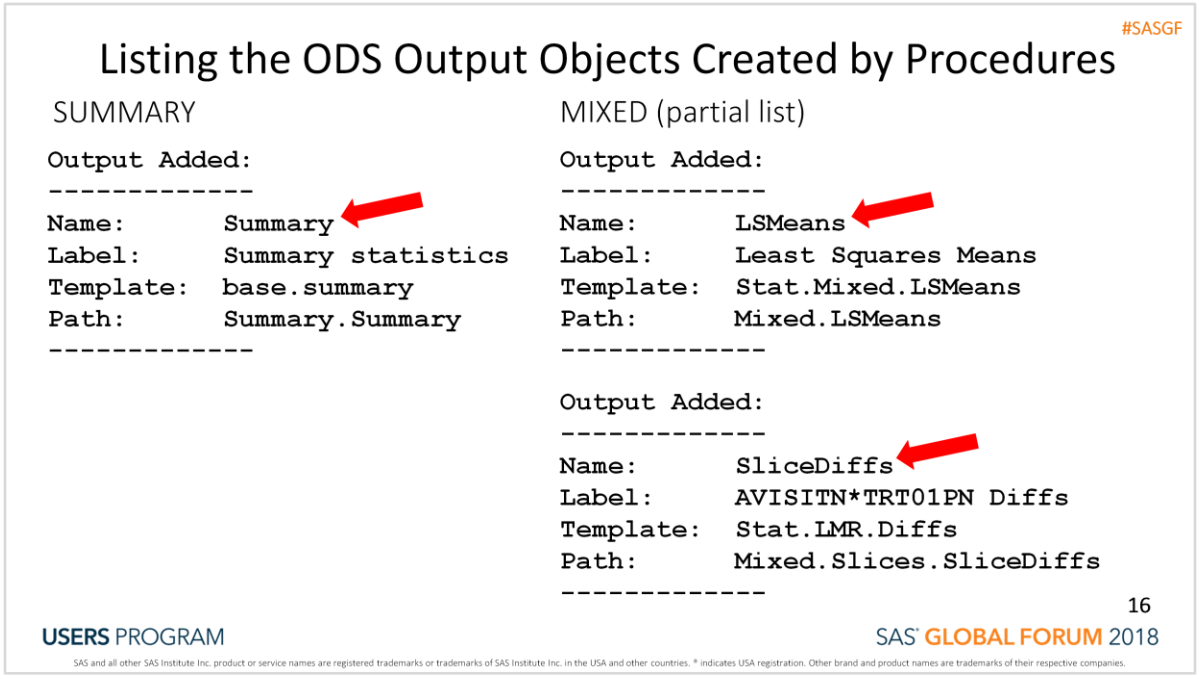
ODS output objects contain output data resulting from the execution of SAS procedures.

You can save the output objects as SAS tables, and then use the tables to create customized reports.

Use the ODS TRACE statement to list the output objects created by the SUMMARY and MIXED procedures.

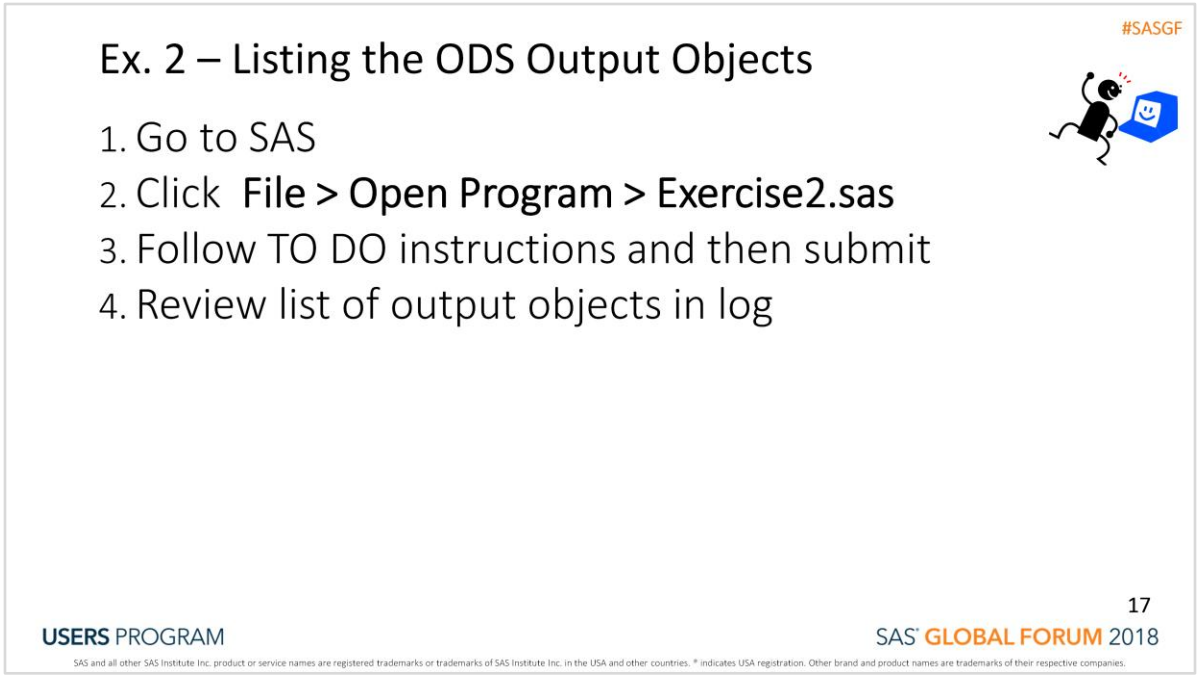
ODS TRACE Statement:

https://goo.gl/EKP8AM



The output objects for PROC SUMMARY are displayed twice in the log, once for each instance of the SUMMARY procedure. The SUMMARY objects contain the data used to create the column headings, and also the statistics in the **Summary Statistics** section of the report.

Only two of the many ODS output objects available from PROC MIXED are shown. The LSMEANS object contains the data used in the **LS Means** section of the report, and the data for the **Treatment Difference (Active - Placebo**) section is stored in the SLICEDIFFS object.



TO DO:

Specify the ODS statements to list the available output tables (Lines 14 and 49).

ODS output objects are displayed in the log.

Hint:

Refer to Slide 15.

#SASGF

Creating SAS Tables from the Output Objects

- ODS OUTPUT statement creates tables from objects
- General Syntax:

```
ods output output-object-name1 = SAS-table-name1
           output-object-name2 = SAS-table-name2 ...;
```

Report Item	Procedure	Object Name	SAS Table Name
Column Headings	SUMMARY	Summary	work.colhead
Summary Statistics	SUMMARY	Summary	work.summary
LS Means	MIXED	LSMeans	work.lsmeans
Treatment Difference	MIXED	SliceDiffs	work.diffs

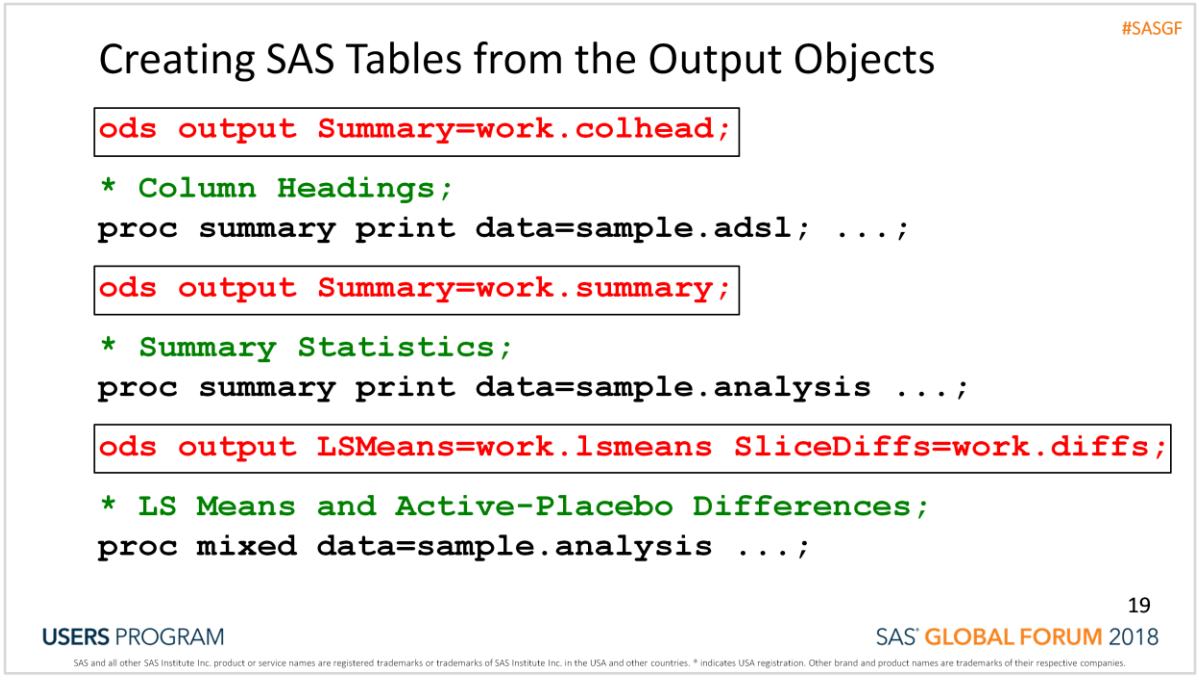
18

USERS PROGRAM SAS® GLOBAL FORUM 2018

This table shows the output objects needed for the report items, and the name of the SAS tables created from them. Refer to Slide 15.

ODS OUTPUT Statement:

https://goo.gl/pDwq7u



Because we need to manipulate data from only some of the output objects, we add these ODS OUTPUT statements to our code.

The SLICEDIFFS output object does not contain data for the AVISITN variable or a record for the placebo. Later we use DATA step code to post-process DIFFS table to create the DIFFS2 table.

Ex. 3 – Creating SAS Tables from the Output Objects

#SASGF

1. Go to SAS
2. Click **File > Open Program > Exercise3.sas**
3. Follow TO DO instructions and then submit
4. Review PROC PRINT output of the tables

20

USERS PROGRAM

SAS® GLOBAL FORUM 2018

TO DO:

Add the ODS statements to save only the output tables of interest (Lines 18, 28, and 43).

NOTE addition of DATA step code (starting at Line 55) to post-process the DIFFS table, and PRINT of preliminary report data (starting at Line 75).

The DIFFS table does not contain data for the AVISITN variable or a record for the placebo. We post-process this data to correct these issues and create the DIFFS2 table.

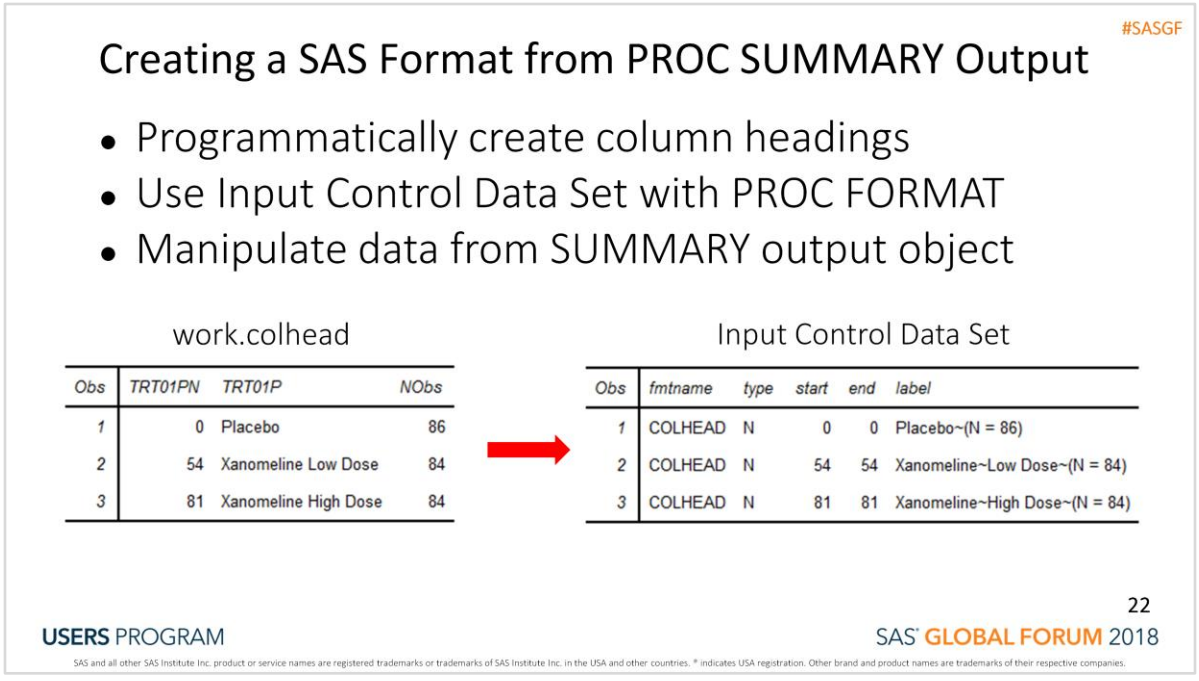
_TRT01PN is our control variable in PROC MIXED and always has a value of zero (Placebo). All statistics for the placebo record have missing values and are displayed as a dash (-) in our report.

The "Statistics.htm" file now contains the SUMMARY and MIXED procedure output, and a listing of the COLHEAD, SUMMARY, LSMEANS, and DIFFS2 tables.

Hint:

Refer to Slide 19.

Manipulating the Procedure Output

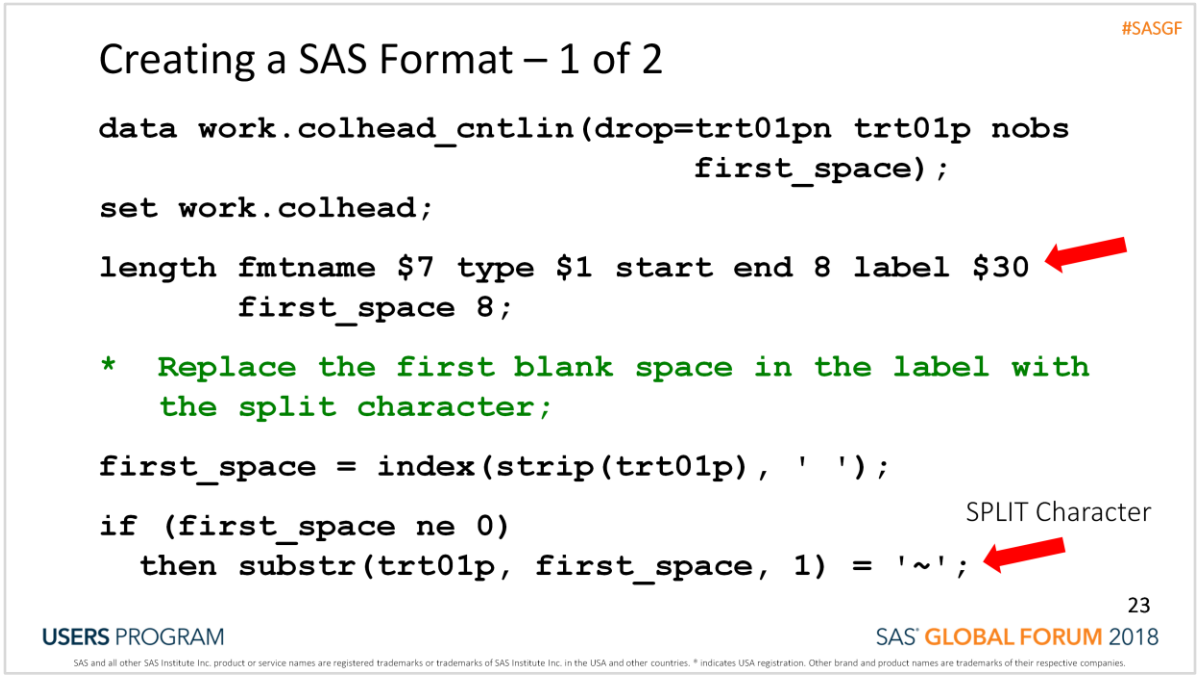


We need to create column headings based on the formatted value of the TRT01PN variable. We could create a SAS format by hard-coding FORMAT procedure statements using data from printed output of the SUMMARY procedure.

But dynamically creating the format from an input control data set provides a more flexible, data-driven approach.

Input Control Data Set:

https://goo.gl/biFEuP



The COLHEAD table, created by saving an output object from the first instance of the SUMMARY procedure, has the data needed to create the format.

The LENGTH statement defines the variables required by PROC FORMAT.

We need to insert our SPLIT character to ensure that the column headings have the desired appearance.

The INDEX function searches TRT01P for the first blank space, and then returns either the position of the blank space, or zero if no blank space is found.

The SUBSTR function, when used to the left of the equals sign, modifies the value of TRT01P. The first blank space is replaced with the split character.

SUBSTR (left of =) Function:

https://goo.gl/HUxYyF

#SASGF

Creating a SAS Format – 2 of 2

```
*  Specify the required information for the format;

fmtname = 'COLHEAD';
type    = 'N';
start   = trt01pn;
end     = trt01pn;
label   = strip(trt01p) || '~(N = ' ||
          strip(put(nobs, &STATFMT0D)) || ')';
run;

proc format cntlin=work.colhead_cntlin; run; quit;
```

STATFMT0D = 12.

24

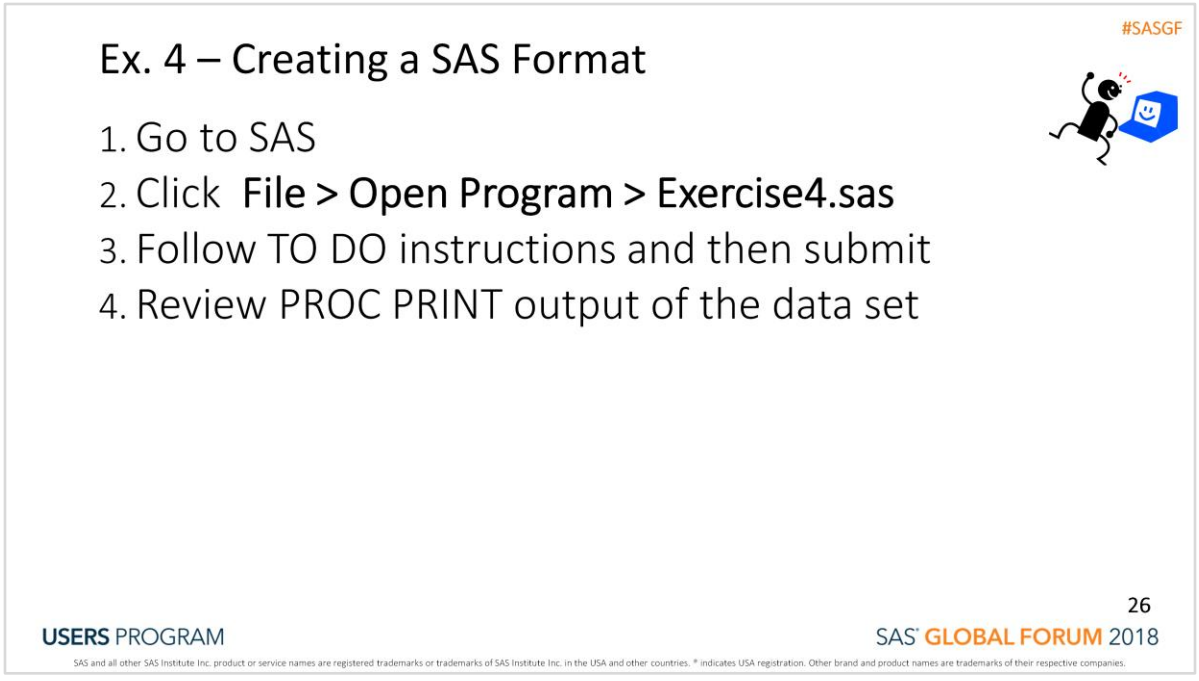
USERS PROGRAM

SAS® GLOBAL FORUM 2018

Specify values for the required variables of the COLHEAD_CNTLIN input control data set. We add the appropriate N= value to the end of TRT01P to obtain the value of the format label.

The Input Control Data Set

Obs	fmtname	type	start	end	label
1	COLHEAD	N	0	0	Placebo~(N = 86)
2	COLHEAD	N	54	54	Xanomeline~Low Dose~(N = 84)
3	COLHEAD	N	81	81	Xanomeline~High Dose~(N = 84)



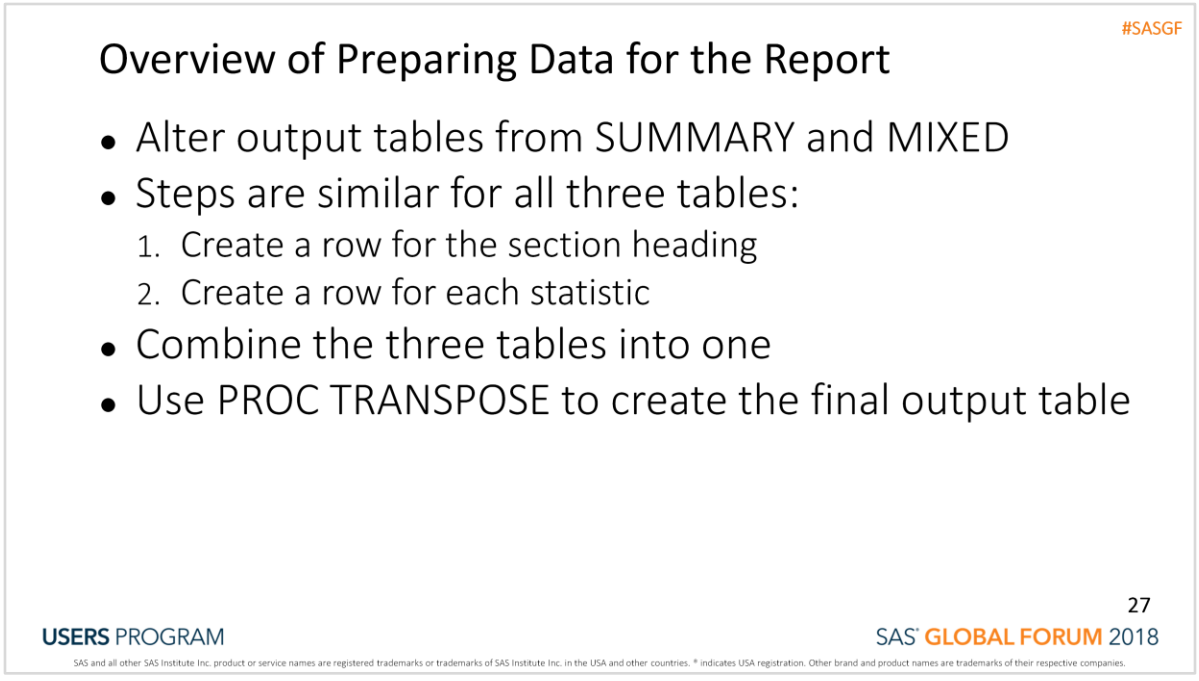
TO DO:

Specify the values needed to create the input control data set for a numeric format named COLHEAD (Lines 25, 26, 27, and 28).

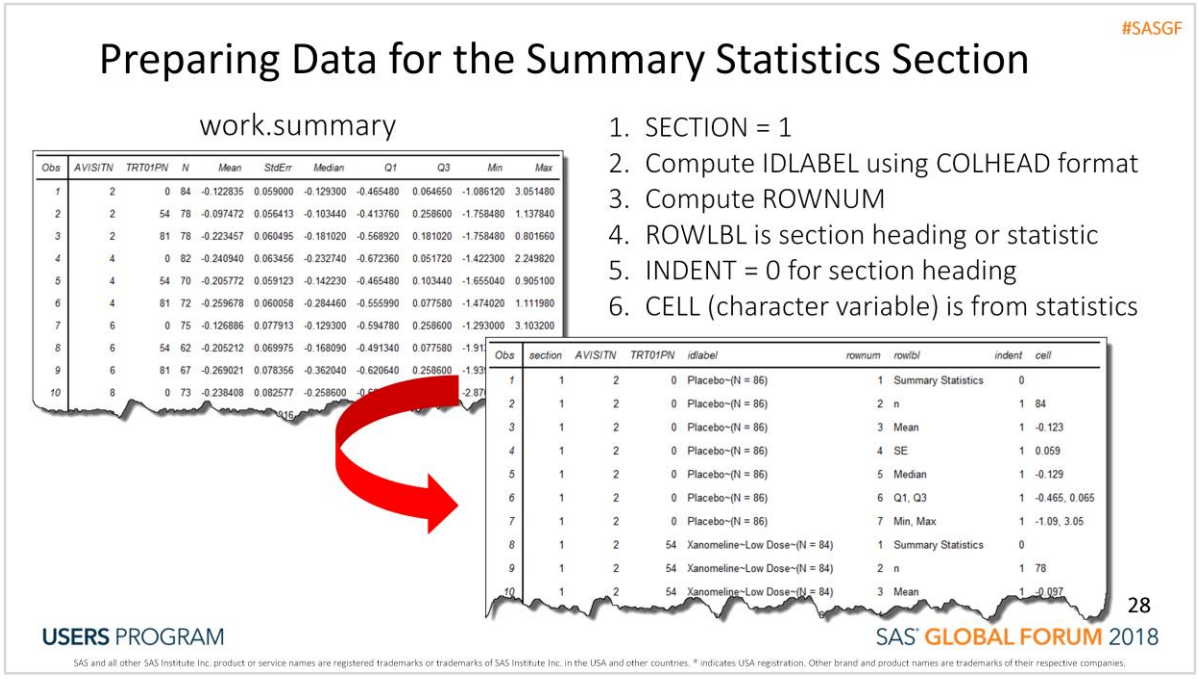
Information about the format is displayed in the "COLHEADFormat.htm" file in our workshop directory. Use this information to verify that the format was created correctly.

Hint:

Refer to Slide 24.



The three tables from the SUMMARY and MIXED procedures need to be altered and combined before they can be used to produce the report.



We need to use DATA step code to create the SAS table shown on the right from the SUMMARY table saved from the second instance of PROC SUMMARY.

The SECTION variable indicates that this data is used for the **Summary Statistics** section.

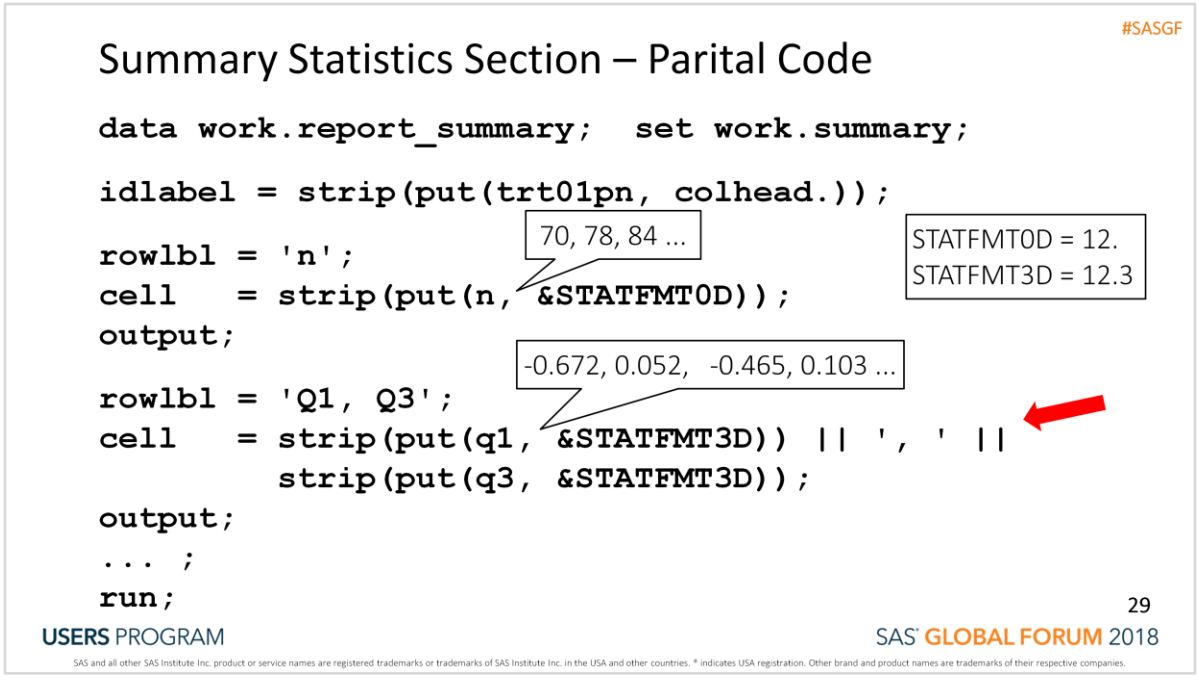
The IDLABEL variable, used later to format the column headings in the report, is created using the COLHEAD format.

A row number is created for each report item based on the distinct visit-treatment combination.

Values of ROWLBL are either **Summary Statistics** (the section heading), or the statistic label such as **N**, **Mean**, **SE**, and so on.

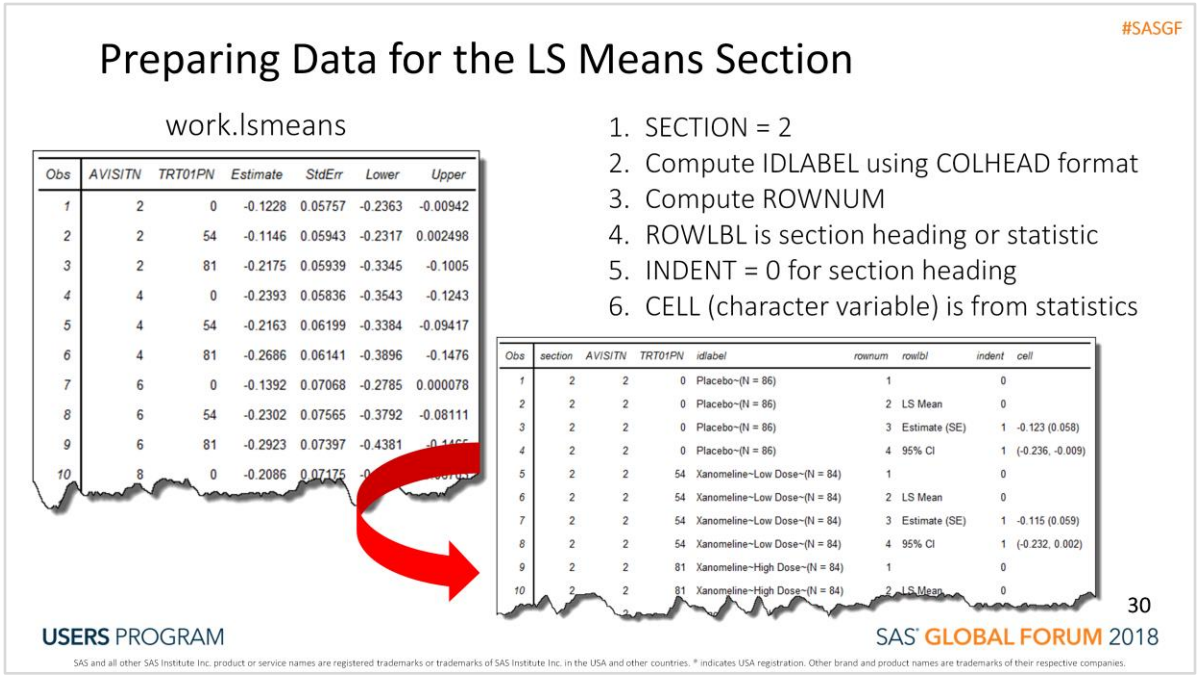
INDENT is set to 1 for the statistic labels to indicate that these labels should be indented underneath the section heading.

All data values are stored in the CELL *character* variable

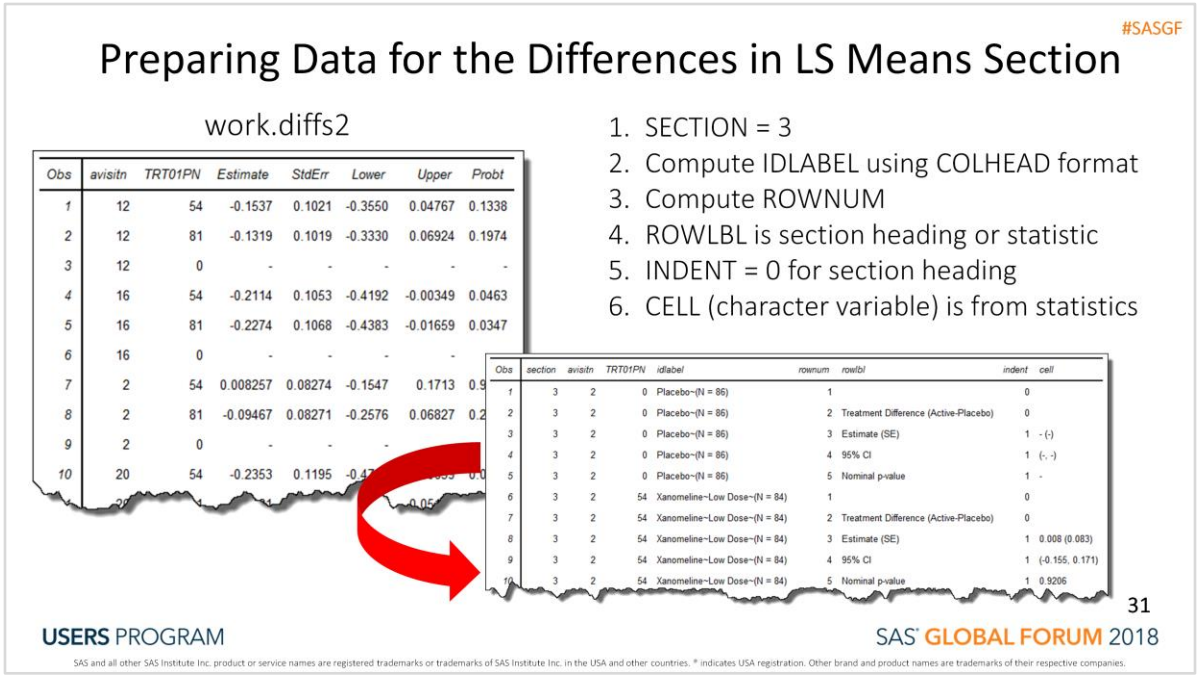


The STATFMT0D macro variable ensures that the N statistic is displayed without decimal places, and the STATFMT3D macro variable displays the Q1 and Q3 statistics with three decimal places.

The sample values of the **Q1, Q3** row illustrate why the CELL variable must be of character type.



The DATA step code to create the SAS table shown on the right from the LSMEANS table is similar to the code used in the **Summary Statistics** section.



The DATA step code to create the SAS table shown on the right from the DIFFS2 table is similar to the code used in the **Summary Statistics** section.

The PVALUE format ensures that p-values are appropriately displayed.

PVALUE*w.d* Format:

https://goo.gl/bU5XTH

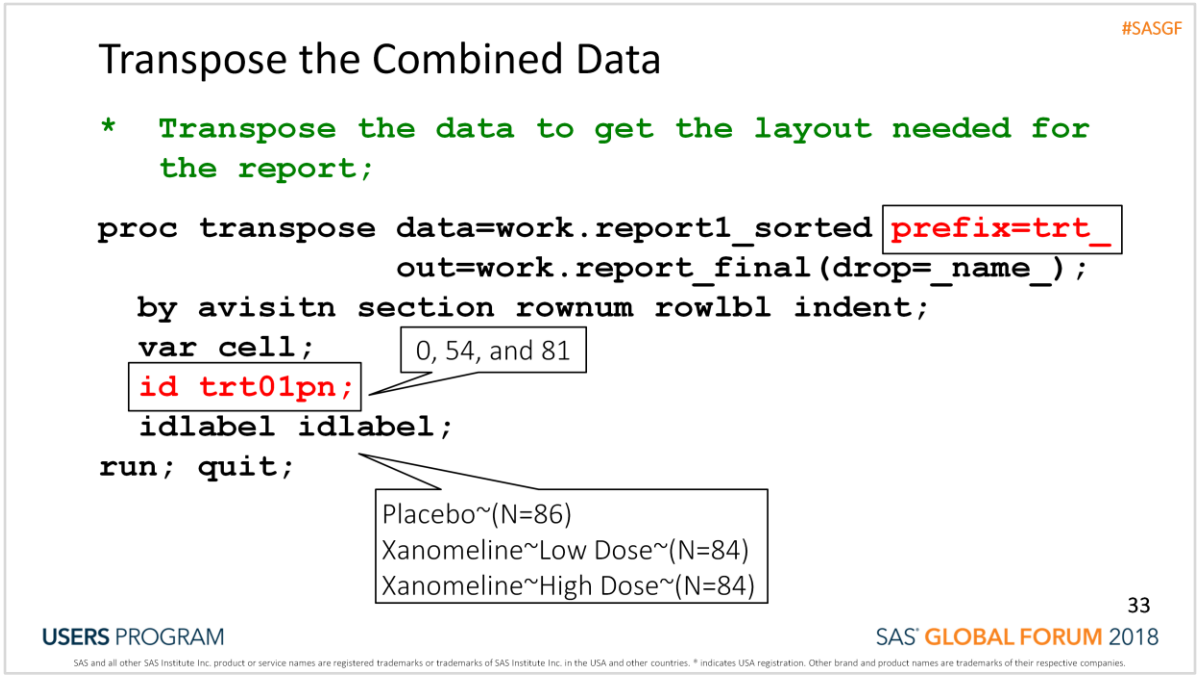
#SASGF

Combine Data from the Three Sections and Sort

```
*  Combine the preliminary data into a single table;

data work.report1;
set work.report_summary
    work.report_lsmeans
    work.report_diffs;
run;

*  Sort in preparation for transposition;

proc sort data=work.report1 out=work.report1_sorted;
  by avisitn section rownum rowlbl indent trt01pn;
run; quit;
```

32

Create the final table by combining the REPORT_SUMMARY, REPORT_LSMEANS, and REPORT_DIFFS tables, and then sort the table.



The REPORT1_SORTED table must be transposed to get the final structure needed for the report.

The BY statement specifies grouping of the data in the REPORT_FINAL table. These variables appear in the output data set, but are not transposed. The VAR statement specifies the variable to be transposed.

Because ID variable TRT01PN has three distinct values (0, 54, and 81), the output data set contains three new columns corresponding to the placebo, the low dose active drug, and the high dose active drug.

The values of the TRT01PN variable are used to name the three new columns in the output data set. Because 0, 54, and 81 are not valid SAS names, the PREFIX option is used to specify valid column names: TRT_0, TRT_54, and TRT_81.

The values of the IDLABEL variable are used to apply the labels "Placebo~(N=86)", "Xanomeline~Low Dose~(N=84)", and "Xanomeline~High Dose~(N=84)" to the TRT_0, TRT_54, and TRT_81 columns, respectively.

#SASGF

The Final Report Data *Without* Column Labels

Obs	AVISITN	section	rownum	indent	rowlbl	trt_0	trt_54	trt_81
1	2	1	1	0	Summary Statistics			
2	2	1	2	1	n	84	78	78
3	2	1	3	1	Mean	-0.123	-0.097	-0.223
4	2	1	4	1	SE	0.059	0.056	0.060
5	2	1	5	1	Median	-0.129	-0.103	-0.181
6	2	1	6	1	Q1, Q3	-0.465, 0.065	-0.414, 0.259	-0.569, 0.181
7	2	1	7	1	Min, Max	-1.09, 3.05	-1.76, 1.14	-1.76, 0.80
8	2	2	1	0				
9	2	2	2	0	LS Mean			
10	2	2	3	1	Estimate (SE)	-0.123 (0.058)	-0.115 (0.059)	-0.218 (0.059)
11	2	2	4	1	95% CI	(-0.236, -0.009)	(-0.232, 0.002)	(-0.335, -0.101)
12	2	3	1	0				
13	2	3	2	0	Treatment Difference (Active-Placebo)			
14	2	3	3	1	Estimate (SE)	- (-)	0.008 (0.083)	-0.095 (0.083)
15	2	3	4	1	95% CI	(-, -)	(-0.155, 0.171)	(-0.258, 0.068)
16	2	3	5	1	Nominal p-value	-	0.9206	0.2535
17	4	1	1	0	Summary Statistics			

34

The REPORT_FINAL table begins to resemble the format of the report.

The newly created TRT_0, TRT_54, and TRT_81 have the labels "Placebo~(N=86)", "Xanomeline~Low Dose~(N=84)", and "Xanomeline~High Dose~(N=84)", respectively.

#SASGF

Ex. 5 – Preparing Data for the Report

1. Go to SAS
2. Click **File > Open Program > Exercise5.sas**
3. Review code and then submit
4. Review PROC PRINT output of the data sets

35

USERS PROGRAM

SAS® GLOBAL FORUM 2018

The first 50 records of the three modified tables (REPORT_SUMMARY, REPORT_LSMEANS, and REPORT_DIFFS) and the REPORT_FINAL table are displayed in the "ReportData.htm" file in our workshop directory.

The STATFMT0D, STATFMT2D, and STATFMT3D macro variables, and the PVALUE format, ensure that the values are appropriately displayed.

PVALUE*w.d* Format:

https://goo.gl/bU5XTH


#SASGF
Creating the Report in RTF Format
36
USERS PROGRAM
SAS GLOBAL FORUM 2018
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.

#SASGF

Partial PROC REPORT Code – 1 of 3

```
proc report data=work.report_final nowd missing
  split='~';
columns avisitn section rownum indent rowlbl trt_: ;
define trt_: / display center;
compute before avisitn / style=[just=l];
  text = catx(' ', 'Week', avisitn);
  line text $;
endcomp;
```

37

USERS PROGRAM

SAS® GLOBAL FORUM 2018

We specify the SPLIT option to ensure that our column headings have the desired appearance.

The TRT_: name prefix in the COLUMNS and DEFINE statements is used to specify the TRT_0, TRT_54, and TRT_81 variables. Using this type of syntax adds flexibility to our program when used with different data.

The COMPUTE block displays the visit number before the data. The CATX function removes leading and trailing blanks from its arguments, concatenates the arguments separated by a delimiter, and then returns the concatenated string. We use a blank space as the delimiter to create values like "Week 2", "Week 16", and so on.

Name Prefix Lists:

https://goo.gl/KzH67R

CATX Function:

https://goo.gl/D6MZZb

#SASGF

Partial PROC REPORT Code – 2 of 3

```
compute rowlbl;
     *  Placeholder for superscript and indenting;
     if (rowlbl eq 'Summary Statistics')
       then;
     else if (rowlbl eq 'LS Mean')
       then;
     if (indent.sum ne 0)
       then;
   endcomp;
```

38

USERS PROGRAM

SAS® GLOBAL FORUM 2018

The null THEN statements serve as placeholders for code to insert superscript text, and to indent row labels under their section headings.

#SASGF

Partial PROC REPORT Code – 3 of 3

```
compute after avisitn / style=[just=l];
  line ' ';
  line 'N = number of subjects randomized ... ';
  line ' ';
  line 'Summary statistics are based on observed
        data.';
  line ' ';
  line 'LS mean is from a repeated measures model
        which includes ... ';
endcomp;

run; quit;
```

39

Superscript text will be added to the text in the two LINE statements indicated.

#SASGF

Create the Report in RTF Format

Table x.y: Change from Baseline LDL-C by Scheduled Visit
Study CDISCPILOT01

	Placebo (N = 86)	*Xanomeline Low Dose (N = 84)*	*Xanomeline High Dose (N = 84)*
Week 2			
Summary Statistics			
n	84	78	78
Mean	-0.123	-0.097	-0.223
SE	0.059	0.056	0.060
Median	-0.129	-0.103	-0.181
Q1, Q3	-0.465, 0.065	-0.414, 0.259	-0.569, 0.181
Min, Max	-1.09, 3.05	-1.76, 1.14	-1.76, 0.80
LS Mean			
Estimate (SE)	-0.123 (0.058)	-0.115 (0.059)	-0.218 (0.059)
95% CI	(-0.236, -0.009)	(-0.232, 0.002)	(-0.335, -0.101)
Treatment Difference (Active-Placebo)			
Estimate (SE)	- (-)	0.008 (0.083)	-0.095 (0.083)
95% CI	(-, -)	(-0.155, 0.171)	(-0.258, 0.068)
Nominal p-value	-	0.9206	0.2535

N = number of subjects randomized; Xanomeline Low Dose = 54 mg; Xanomeline High Dose = 81 mg.

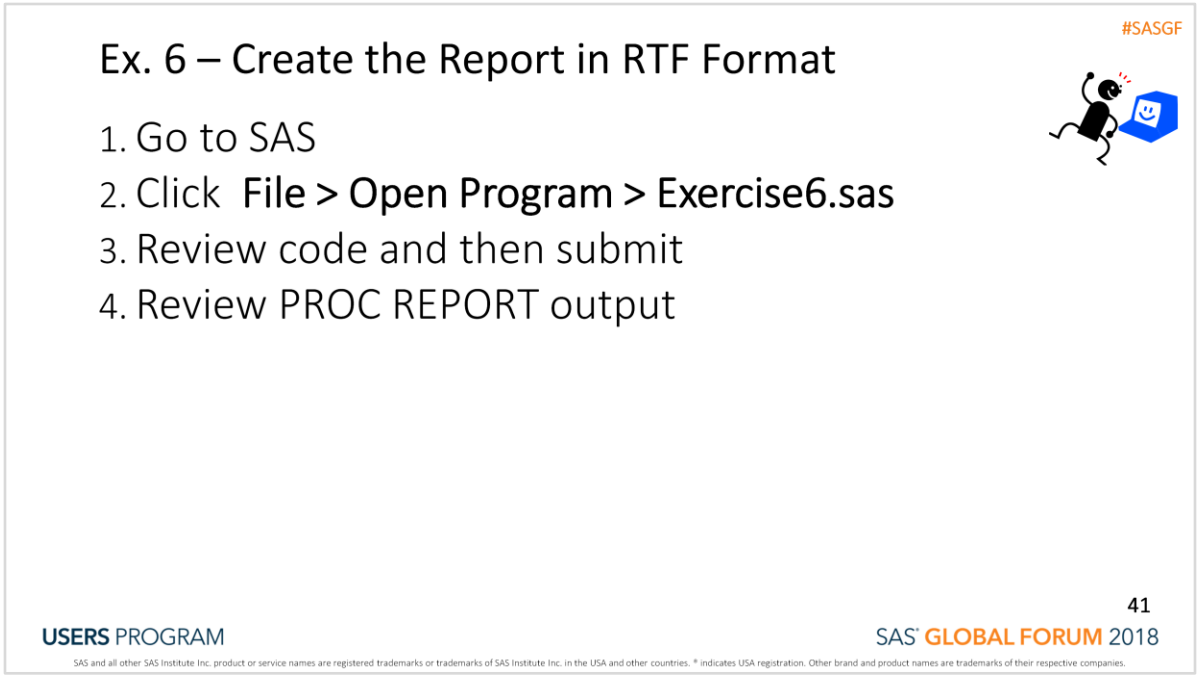
Summary statistics are based on observed data.

LS mean is from a repeated measures model which includes effects: treatment group, scheduled visit, and the interaction of treatment with scheduled visit.

40

Our output closely resembles the final report, except for the following:

1. Superscript text is not present.
2. Statistic row labels are not indented underneath their section headings.
3. The column heading text is displayed in italic text.



The BODYTITLE option in the ODS statement specifies that title text is placed into the body of the RTF document instead of the header section.

The STUDYID macro variable has the value "CDISCPILOT01" and was defined in the "Setup.sas" file.

The TRT_: name prefix in the COLUMNS and DEFINE statements is used to specify the TRT_0, TRT_54, and TRT_81 variables.

The BREAK statement ensures that data for each visit begins on a new page.

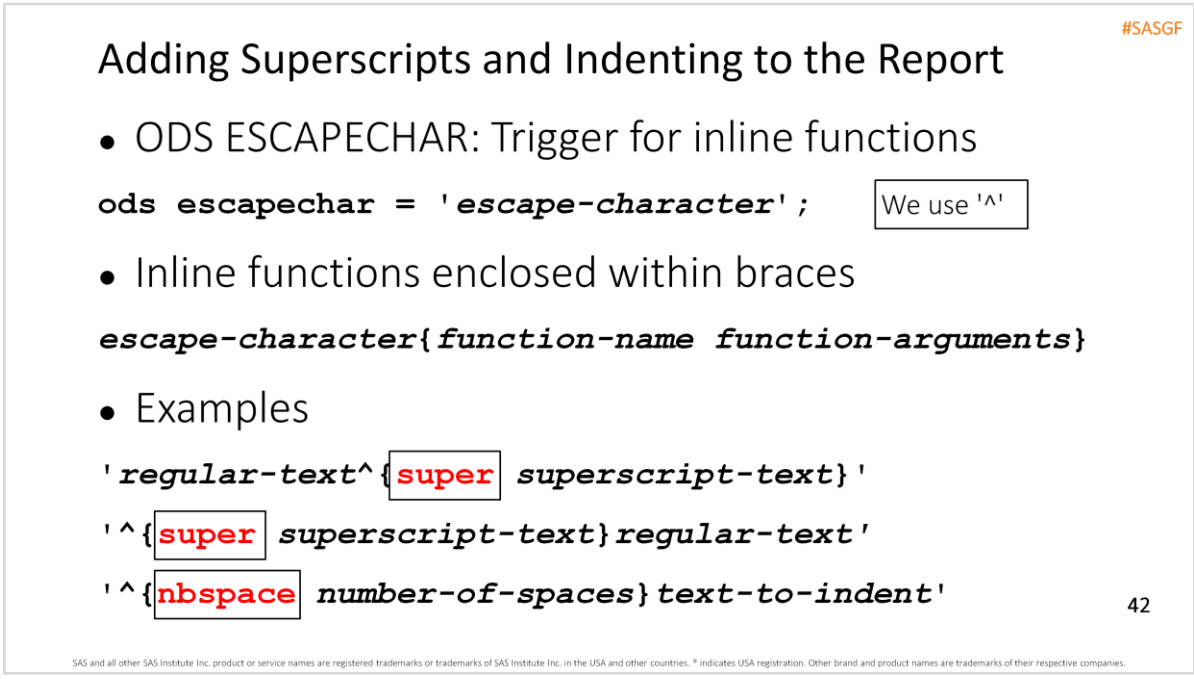
The first COMPUTE block displays the visit number before the data.

The second COMPUTE block is a placeholder for where we will add superscript and indented text to the row labels.

The third COMPUTE block prints information after the body of the report.

ODS RTF Statement:

https://goo.gl/XC1pxJ



Our report is missing the superscript text, and the statistic row labels are not indented under their section headings. We can use ODS inline formatting functions in the COMPUTE blocks to add these features to the text without modifying the REPORT_FINAL SAS table.

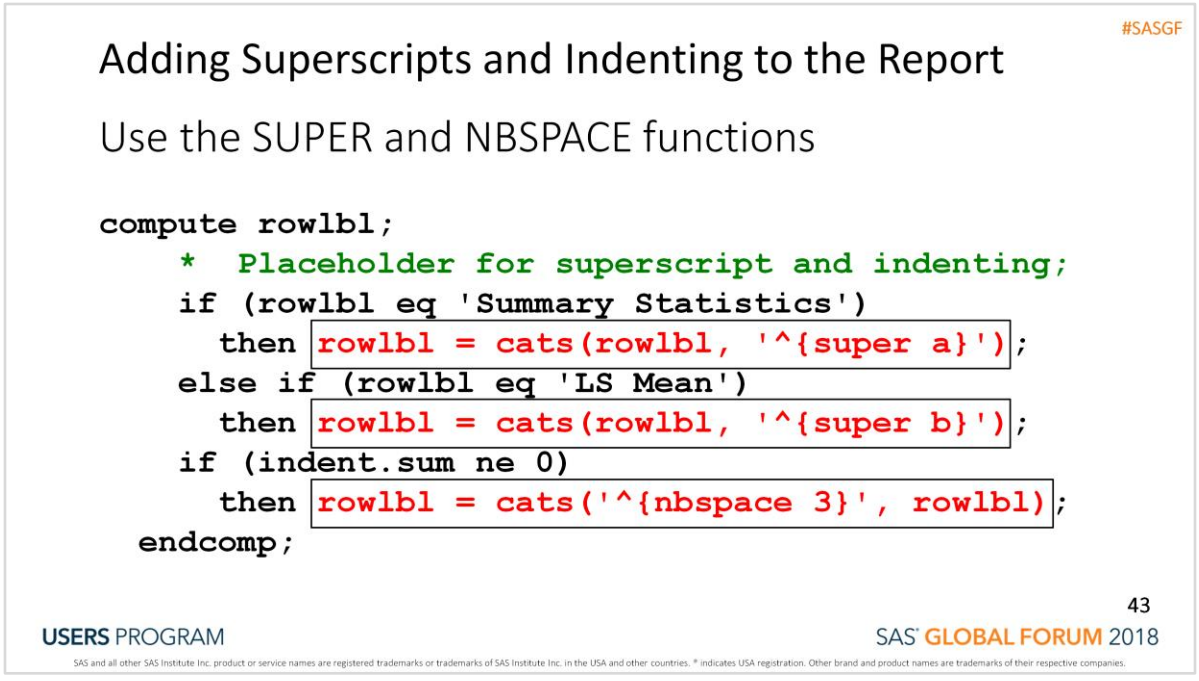
The ODS ESCAPECHAR statement defines a special character to indicate that inline formatting operations follow. We specified ^ for an escape character in the "Setup.sas" file because it is not used in any of our text.

Inline formatting instructions are included in text strings, start with the escape character, and are enclosed within braces.

We use the SUPER function to add superscript text, and the NBSPACE function to indent text using non-breaking spaces.

ODS ESCAPECHAR Statement:

https://goo.gl/pm26iS



These changes are made in the second COMPUTE block.

The CATS function removes leading and trailing blanks from its arguments, concatenates the arguments, and then returns the concatenated string.

CATS Function:

https://goo.gl/6UKnEy

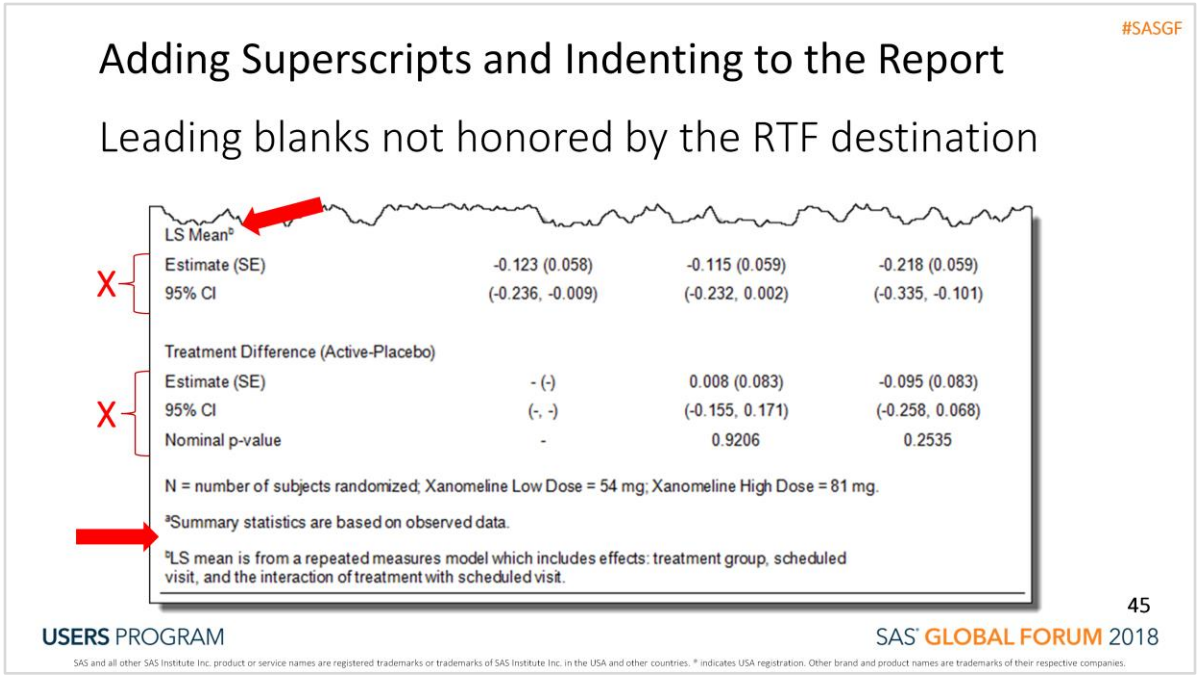
#SASGF

Adding Superscripts and Indenting to the Report

```
compute after avisitn / style=[just=l];
  line ' ';
  line 'N = number of subjects randomized ... ';
  line ' ';
  line '^{super a}Summary statistics are based on
        observed data.';
  line ' ';
  line '^{super b}LS mean is from a repeated
        measures model which includes ... ';
endcomp;

run; quit;
```

44

These changes are made in the third COMPUTE block.



This partial view of the report shows that the row label text is not indented. The ODS RTF destination does not honor leading blanks by default.

Ex. 7 – Adding Superscripts and Indenting

#SASGF

1. Go to SAS
2. Click **File > Open Program > Exercise7.sas**
3. Follow TO DO instructions and then submit
4. Review PROC REPORT output

46

USERS PROGRAM

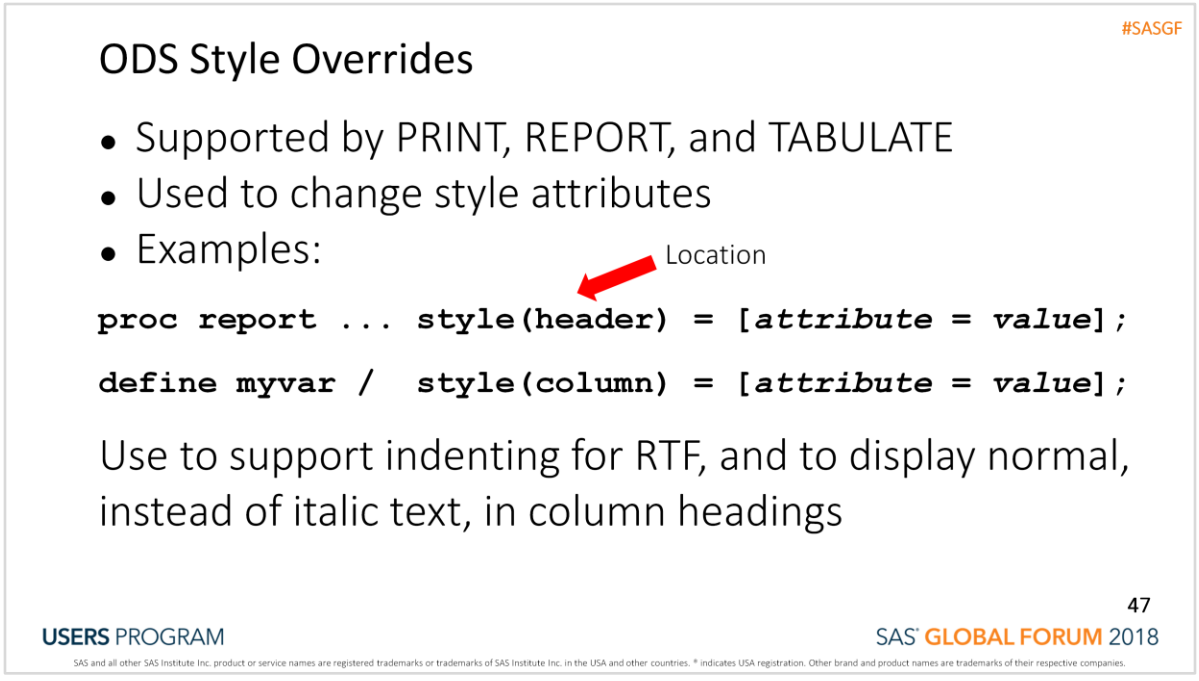
SAS® GLOBAL FORUM 2018

TO DO:

Add the ODS inline formatting functions for superscripts (Lines 34, 36, 45, and 47) and text indenting (Line 38).

Hint:

Refer to Slides 43 and 44.

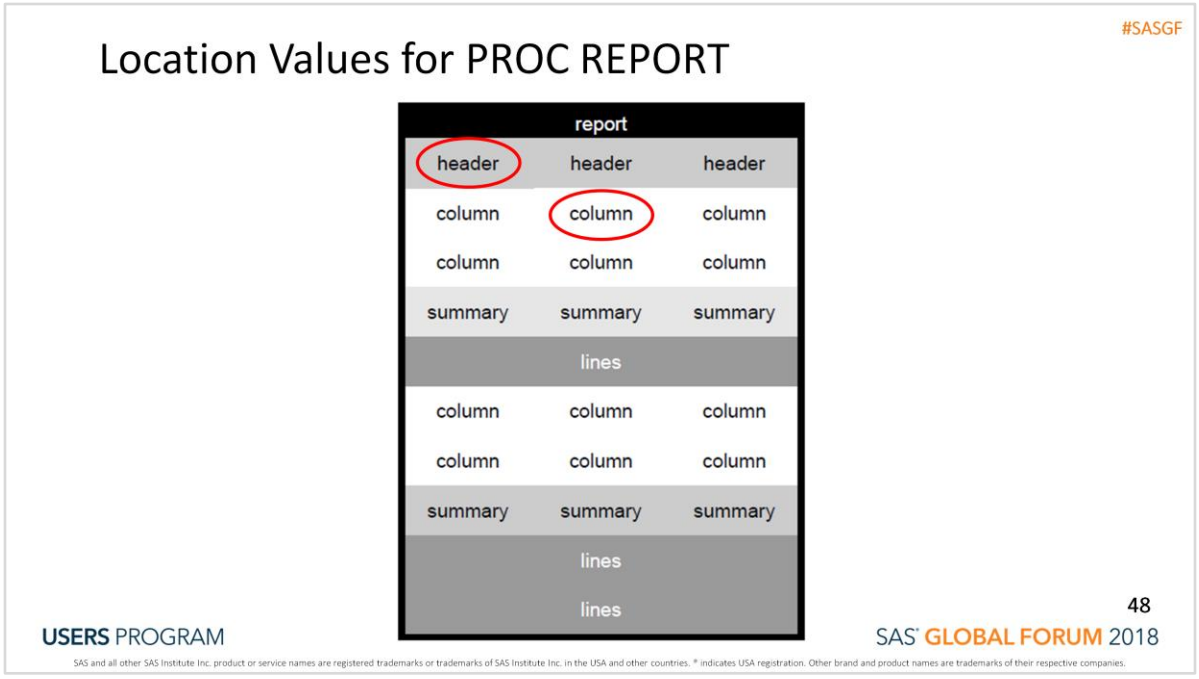


The row label text is not indented, and the column headings in our final report are not italicized. We could use PROC TEMPLATE to change the Journal style to correct these issues, but it is easier instead to use ODS style overrides.

You can alter the appearance of specific parts of your PRINT, REPORT, and TABULATE procedure output by using style overrides. These specific parts of your SAS output are called "locations".

The HEADER location controls the appearance of column headings, and the COLUMN location applies to the data cells.

The first example changes an attribute of *all* column heading cells. The second changes an attribute of only the MYVAR data cells.



These are the locations that are pertinent to the REPORT procedure output.

The HEADER location controls the appearance of column headings, and the COLUMN location applies to the data cells.

SAS 9 Reporting Procedure Styles Tip Sheet:

https://goo.gl/Bmox

#SASGF

Applying Style Overrides to Heading and Data Cells

```
proc report data=work.report_final nowd missing
  split='~' style(header)=[font_style=roman];
columns avisitn section rownum indent rowlbl trt_: ;
define rowlbl / display ' ' style(column)=[asis=on];
... ;
run;
```

The output now has desired appearance

49

You could change the appearance of the column headings by applying a style override in the DEFINE statement for each column displayed in the report, but this is cumbersome when there are several columns. A more efficient way to change the appearance of all column headings is to specify the style override in the PROC statement.

The leading blanks in our statistic labels are honored when we specify `asis=on` in the DEFINE statement for the ROWLBL column.

#SASGF

Ex. 8 – Applying Style Overrides

1. Go to SAS
2. Click **File > Open Program > Exercise8.sas**
3. Follow TO DO instructions and then submit
4. Review PROC REPORT output

50

USERS PROGRAM

SAS® GLOBAL FORUM 2018

TO DO:

Add a style override to the PROC statement to change the heading text from italic to "roman" (Line 15).

Note the addition of style override on Line 22.

Hint:

Refer to Slide 49.

Creating the Report in HTML and PDF Format

Creating the Report in HTML and PDF Format

Add these statements to the final PROC REPORT code

```
ods rtf   file='CHOL_Report.rtf' style=Journal ... ;
ods pdf   file='CHOL_Report.pdf' style=Journal;
ods html5 file='CHOL_Report.htm' style=Journal;

proc report ... ; run; quit;

ods rtf   close;
ods pdf   close;
ods html5 close;
```

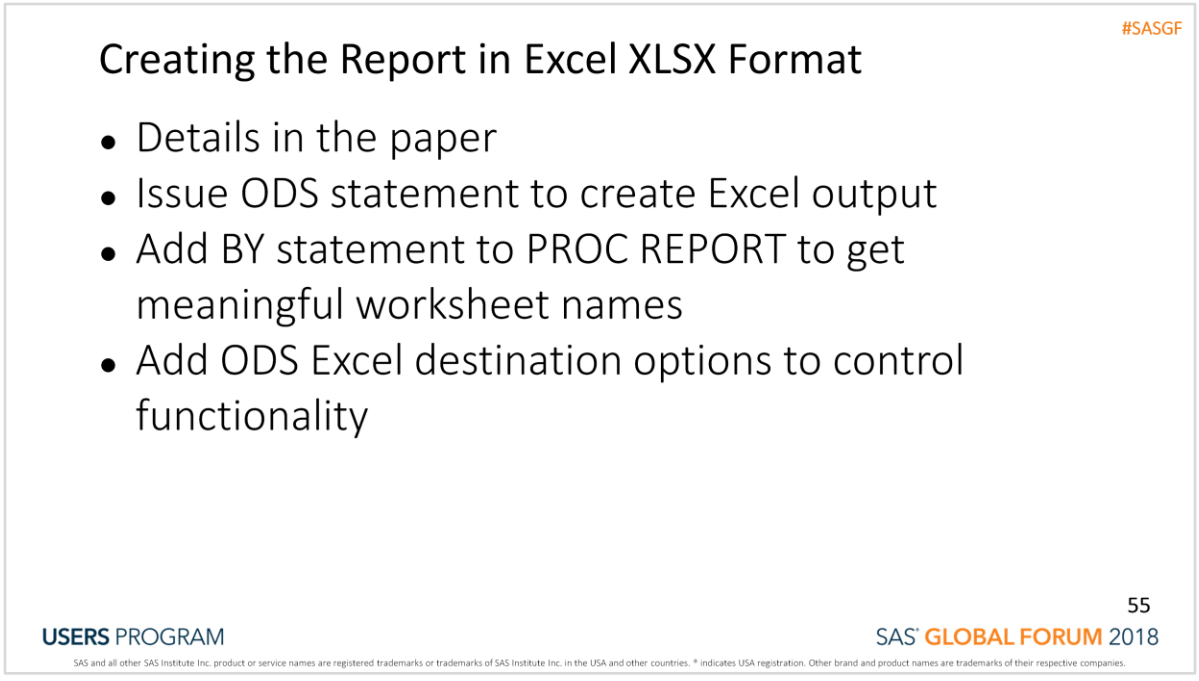
Ex. 9 – Create the Report in HTML and PDF Format

#SASGF

1. Go to SAS
2. Click **File > Open Program > Exercise9.sas**
3. Review code and then submit
4. Review PROC REPORT output

Note the ODS statements at Lines 9, 10, 56, and 57.

Creating the Report in Excel XLSX Format



The ODS Excel destination chooses default worksheet names that are usually not attractive. Using a BY statement combined with Excel-specific options results in meaningful worksheet names. Other options improve the functionality and appearance of the workbook.

#SASGF

Creating the Report in Excel XLSX Format

```
ods excel file='CHOL_Report.xlsx' style=Journal
  options(orientation='landscape'
          embedded_titles='yes'
          suppress_bylines='yes'
          sheet_name='Week #byval(avisitn)');

proc report ... ;
  by avisitn;
   ... ;
run; quit;

ods excel close;
```

56

USERS PROGRAM

SAS® GLOBAL FORUM 2018

Add a BY statement to the PROC REPORT code to create Microsoft Excel output with worksheets named according to the value of AVISITN.

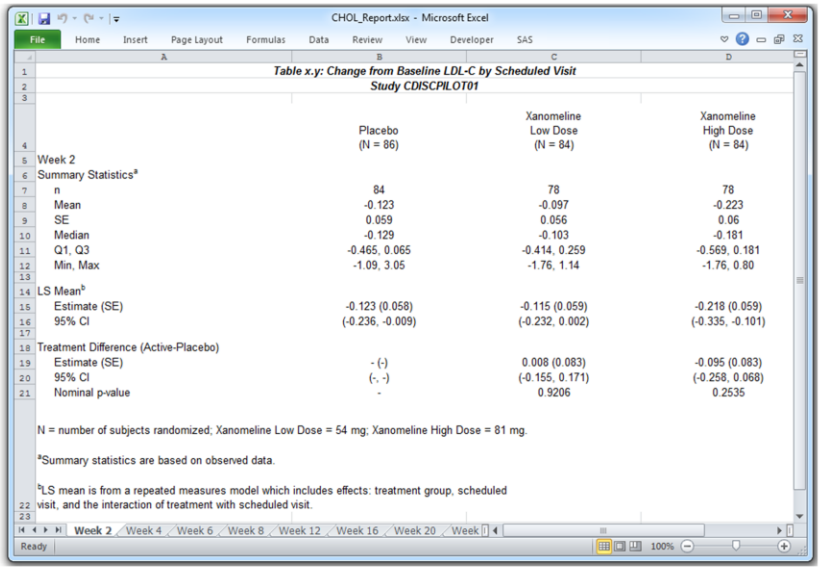
Excel destination options:

orientation	Printed output fits to a single page in landscape orientation.
embedded_titles	Title text appears in the workbook, instead of the print header.
suppress_bylilnes	BY line text is not included in the output.
sheet_name	Worksheet names begin with Week, followed by the week number.

Creating attractive and functional Excel output using SAS is discussed in earlier papers:

http://support.sas.com/rnd/papers/intro-multisheet-excel-with-sas/index.html

Creating the Report in Excel XLSX Format



CHOL_Report.xlsx - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Developer SAS

Table x.y: Change from Baseline LDL-C by Scheduled Visit
Study CDISCPILOT01

	Placebo (N = 86)	Xanomeline Low Dose (N = 84)	Xanomeline High Dose (N = 84)
Week 2			
Summary Statistics[a]			
n	84	78	78
Mean	-0.123	-0.097	-0.223
SE	0.059	0.056	0.06
Median	-0.129	-0.103	-0.181
Q1, Q3	-0.465, 0.065	-0.414, 0.259	-0.569, 0.181
Min, Max	-1.09, 3.05	-1.76, 1.14	-1.76, 0.80
LS Mean[b]			
Estimate (SE)	-0.123 (0.058)	-0.115 (0.059)	-0.218 (0.059)
95% CI	(-0.236, -0.009)	(-0.232, 0.002)	(-0.335, -0.101)
Treatment Difference (Active-Placebo)			
Estimate (SE)	- (-)	0.008 (0.083)	-0.095 (0.083)
95% CI	(-, -)	(-0.155, 0.171)	(-0.258, 0.068)
Nominal p-value	-	0.9206	0.2535

N = number of subjects randomized; Xanomeline Low Dose = 54 mg; Xanomeline High Dose = 81 mg.

[a]Summary statistics are based on observed data.

[b]LS mean is from a repeated measures model which includes effects: treatment group, scheduled visit, and the interaction of treatment with scheduled visit.

Week 2 Week 4 Week 6 Week 8 Week 12 Week 16 Week 20 Week

Ready 100%

57

The output viewed using Microsoft Excel.

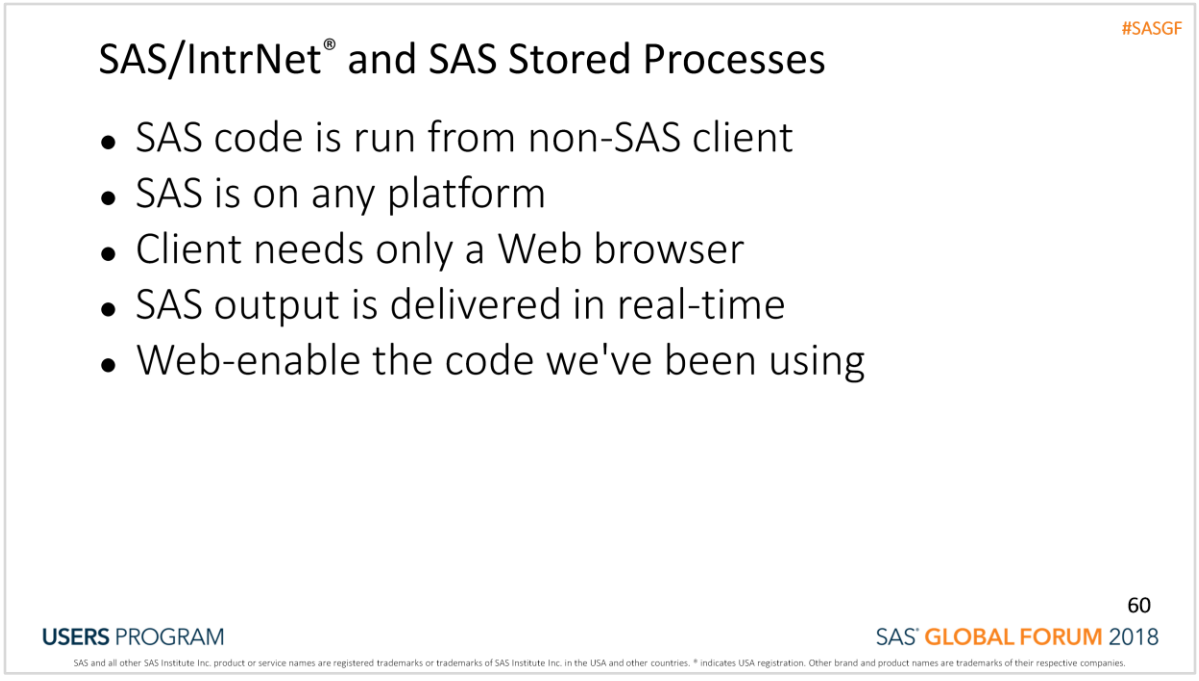
Ex. 10 – Create the Report in Excel XLSX Format

#SASGF

1. Go to SAS
2. Click **File > Open Program > Exercise10.sas**
3. Review code and then submit
4. Review PROC REPORT output

Note the ODS statements at Lines 16 and 64.

Using SAS/IntrNet® and SAS Stored Processes



The purpose of the SAS/IntrNet® Application Dispatcher or SAS Stored Processes are to allow you to execute SAS programs from a client machine that does not have SAS installed. The client machine *may* have SAS installed, but that is not required.

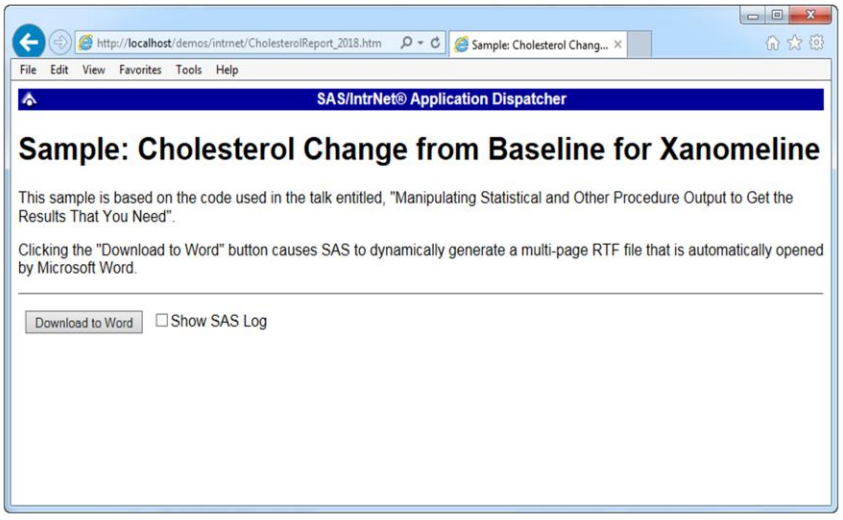
A typical client-server model is followed. The SAS server can reside on any hardware platform (Windows, UNIX, z/OS, and so on) and is standing by, waiting to execute a SAS program. The most common client is a Web browser, again, running on any platform.

When the OK button of the Web page is clicked, input parameters, if any, are sent to the SAS server. Your SAS code executes, and the output is delivered in real-time to the Web browser.

The following slides illustrate this process, using a Web-enabled version of the SAS code we have been working with.

#SASGF

Dynamically Generated RTF



61

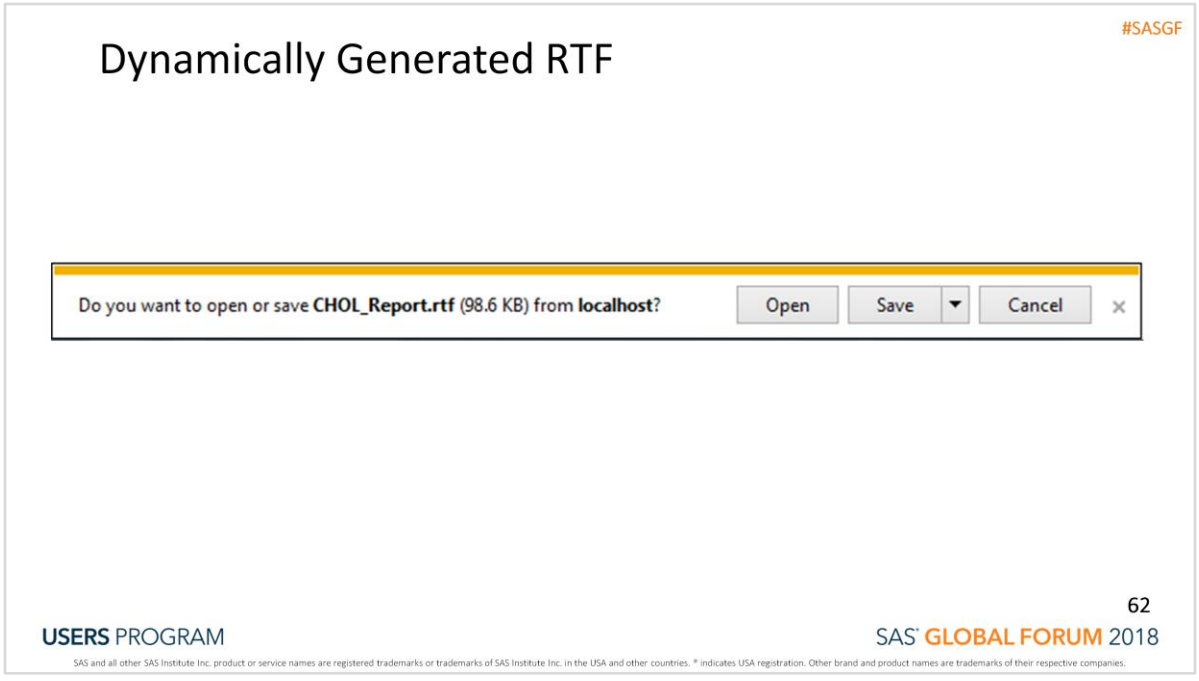
USERS PROGRAM SAS® GLOBAL FORUM 2018

Here is a simple Web page that is used to execute SAS code stored on a server using the SAS/IntrNet Application Dispatcher.

The code that is executed is substantially similar to the final version of the code that we used to generate RTF file, with a few changes to "Web enable" it.

Refer to the accompanying paper for instructions on how to Web enable your code.

The SAS program executes on the server when the **Download to Word** button is clicked.



Once the SAS program executes, the results are sent back to the Web browser.

Instead of the results being displayed in the Web browser, you are presented with a File Download dialog box.

You can click **Open** to immediately open your SAS output using Excel, or click **Save** to save a copy for later use.

#SASGF

Dynamically Generated RTF

Table x.y: Change from Baseline LDL-C by Scheduled Visit
Study CDISCPILOT01

	Placebo (N = 86)	Xanomeline Low Dose (N = 84)	Xanomeline High Dose (N = 84)
Week 2			
Summary Statistics[a]			
n	84	78	78
Mean	-0.123	-0.097	-0.223
SE	0.059	0.056	0.060
Median	-0.129	-0.103	-0.181
Q1, Q3	-0.465, 0.065	-0.414, 0.259	-0.569, 0.181
Min, Max	-1.09, 3.05	-1.76, 1.14	-1.76, 0.80
LS Mean[b]			
Estimate (SE)	-0.123 (0.058)	-0.115 (0.059)	-0.218 (0.059)
95% CI	(-0.236, -0.009)	(-0.232, 0.002)	(-0.335, -0.101)
Treatment Difference (Active-Placebo)			
Estimate (SE)	- (-)	0.008 (0.083)	-0.095 (0.083)
95% CI	(-, -)	(-0.155, 0.171)	(-0.258, 0.068)
Nominal p-value	-	0.9206	0.2535

N = number of subjects randomized; Xanomeline Low Dose = 54 mg; Xanomeline High Dose = 81 mg.

[a]Summary statistics are based on observed data.

[b]LS mean is from a repeated measures model which includes effects: treatment group, scheduled visit, and the interaction of treatment with scheduled visit.

63

Here is the SAS output, created in real-time, and delivered to the client. Refer to the accompanying paper for more information about this topic.

SAS/IntrNet Application Dispatcher
https://goo.gl/ktequS

SAS Stored Processes
https://goo.gl/vVZ5DU

#SASGF

Summary and Conclusion

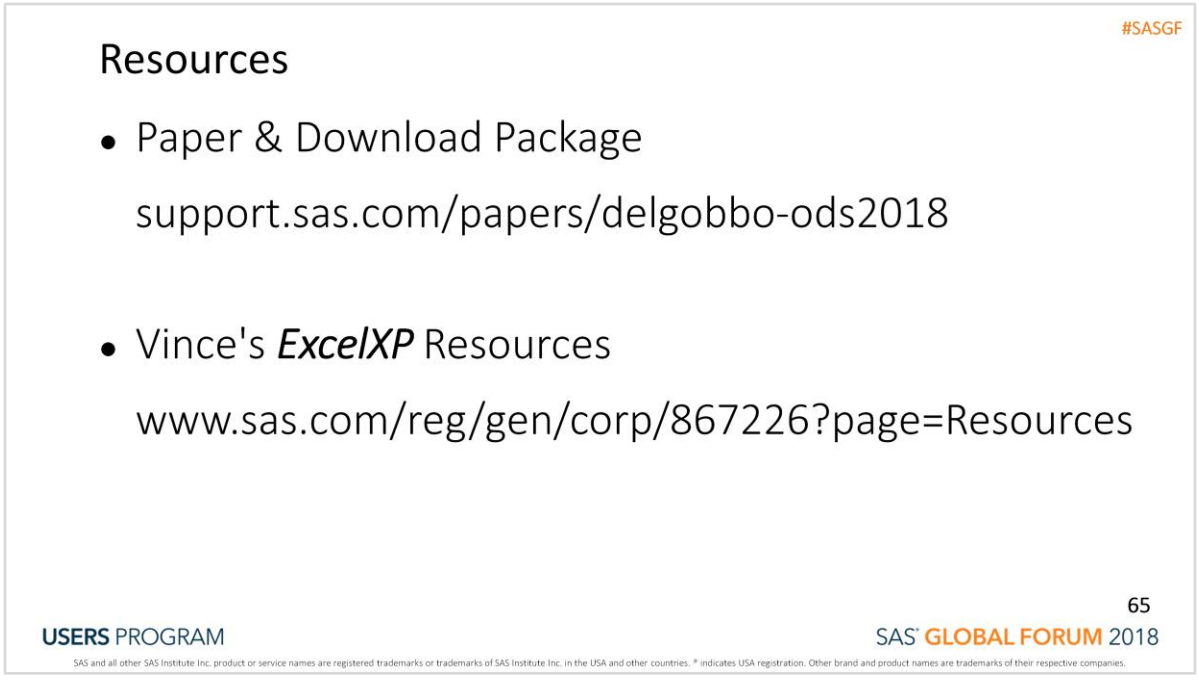
- ODS TRACE to list procedure output objects
- ODS OUTPUT to save the objects as SAS tables
- Can manipulate the data using DATA step and PROCs
- Dynamically generate a SAS format
- Use ODS inline formatting functions and style overrides
- Easy to create RTF, HTML, and PDF output
- A few extra steps to create Excel XLSX output
- Can deliver results in real-time w/ server technology

64

USERS PROGRAM

SAS® GLOBAL FORUM 2018

Here is a summary of what we have learned.

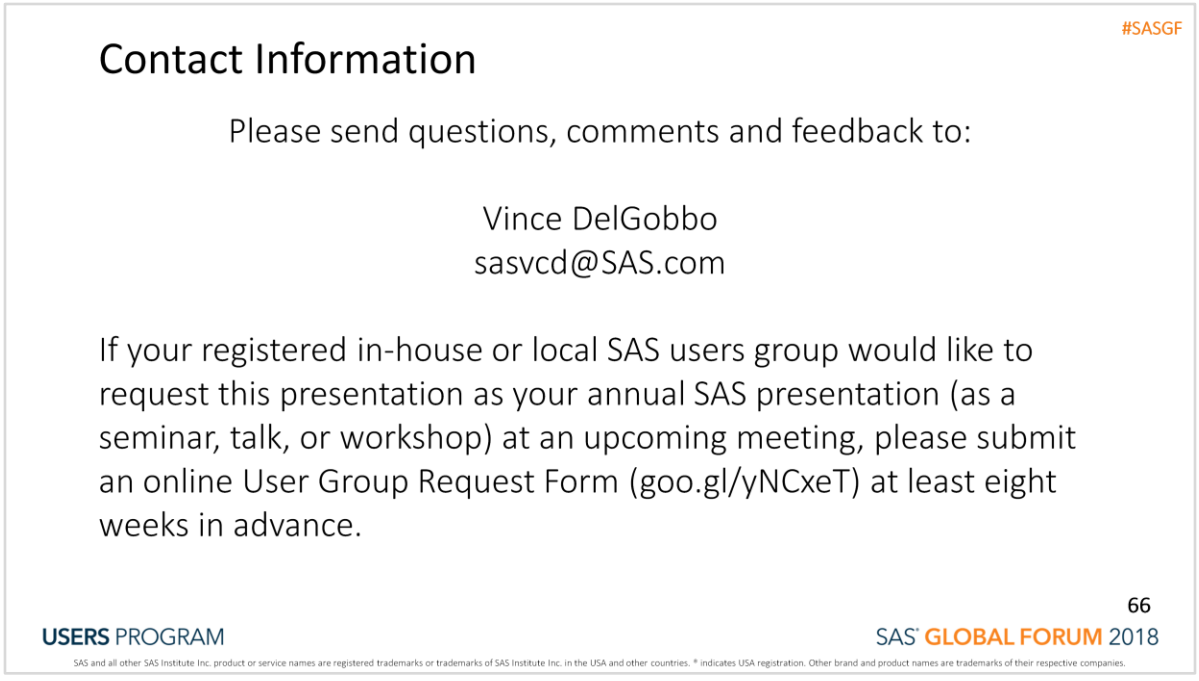


Paper & Download Package

http://support.sas.com/papers/delgobbo-ods2018

Vince's *ExcelXP* Resources

http://www.sas.com/reg/gen/corp/867226?page=Resources



About the author:

Vince DelGobbo is a Senior Software Developer in the Metadata and Execution Services group at SAS. This group's responsibilities include the SAS/IntrNet Application Dispatcher and SAS Stored Processes. He is involved in the development of new Web- and server-based technologies, bringing 3rd-party metadata into SAS, and integrating SAS output with Microsoft Office. He was also involved in the early development of the ExcelXP ODS tagset. Vince has been a SAS Software user since 1982, and joined SAS in 1992.

LinkedIn:

https://www.linkedin.com/in/vince-delgobbo-a717b88/

User Group Request Form

http://goo.gl/yNCxeT