Paper 49-2019

Interactive Graphs

Kriss Harris, SAS Specialists Limited; Richann Watson, DataRich Consulting

ABSTRACT

This paper demonstrates how you can use interactive graphics in SAS® 9.4 to assess and report your safety data. The interactive visualizations that you will be shown include the adverse event and laboratory results. In addition, you will be shown how to display "details-on-demand" when you hover over a point.

Adding interactivity to your graphs will bring your data to life and help improve lives!

INTRODUCTION

Being able to drill-down on your graphs enables you to interrogate your graphs and understand your data better. That is the beauty of having interactive visualizations.

To create graphs, which can be drilled down on requires HTML. Based on the assumption that you already know how to create graphs with the SG Procedures or using the Graph Template Language (GTL), then the only additional things you need to produce drill-down graphs in SAS 9.4 are the following:

- Using ODS OUTPUT HTML statements around your SG procedures or your GTL statements;
- Using the URL= <*CHARACTER-VARIABLE*> option in your plot statements;
- Using the IMAGEMAP = ON option within ODS GRAPHICS.

Being able to filter graphs by selecting items from a drop-down box is very useful too. This can be achieved in SAS by using the same methods, which the drill-down uses, and in addition using a few lines of HTML code, to create your drop-down menu. JavaScript and Cascading Style Sheets (CSS) are also needed, however the JavaScript and CSS that you need are provided with these examples, and therefore, you do not need to learn how to write JavaScript or CSS, to be able to use filtering.

For more details on creating graphs using GTL then please see (Harris & Watson, Great Time to Learn GTL, 2018), (Harris, Hands-On Graph Template Language (GTL): Part A, 2017) or (Watson, 2019)

The adverse event and laboratory data sets used in this paper are from the CDISC SDTM / ADaM Pilot Project and they were obtained from the CDISC website (CDISC, 2013).

ADVERSE EVENT DRILL-DOWN EXAMPLE

The following link shows an Adverse Event (AE) plot, which you are able to drill-down on - https://www.krissharris.co.uk/sgf/2019/3261/output/aes.html.

When incorporating a drill-down feature, it is important to know what you want to display after you have drilled-down. When you hover over the graph, you can display the active link that can take you to another graph or a table or it can display basic information pertaining to that specific item in the graph. In the AE example in the above link, when you click on (or drill-down on) the bar which shows the percentage of subjects with at least one **Application Site Dermatitis** event in the **Placebo** treatment group, you are then shown the number of subjects which had that AE by each severity group. In this example, all the Application Site Dermatitis events were mild, and so if you click on that bar, you are then shown the results table of the five subjects, which had that event.

Using the AE example, drill-down capability was achieved by:

1. Producing subject level listings of each AE, for each treatment, and severity group. (1)
2. Producing graph of counts of AE's, for each treatment, and severity group. (2)
    - Linking these bars to the subject level listings in **"1".**
3. Producing a graph of percentage of subjects with the AE's by each treatment group. (3)

- Linking these bars to the counts of AE's in **"2"**.

In order to achieve the drill-down affect, an ouptut for each combination specified in bullet points 1 and 2 above needs to be created. Because an output for all combinations needs to be created, it is recommended to use a macro.

PRODUCING SUBJECT LEVEL LISTING OF EACH AE, TREATMENT AND SEVERITY GROUP

Output 1 below shows an example of the listings, which are created for each AE, treatment and severity group. This is the final level of the drill-down.

Click back arrow on navigation bar to return to severity bar chart

Body System or Organ Class=GENERAL DISORDERS AND ADMINISTRATION SITE CONDITIONS Dictionary-Derived Term=APPLICATION SITE DERMATITIS Severity/Intensity=MILD Actual Treatment=Placebo

Unique Subject Identifier	Reported Term for the Adverse Event	Analysis Start Date	Analysis End Date	AE Duration (N)	Serious Event	Causality	AREL	Outcome of Adverse Event
01-709-1088	APPLICATION SITE DERMATITIS	08MAY2014	.	.	N	PROBABLE	RELATED	NOT RECOVERED/NOT RESOLVED
01-709-1312	APPLICATION SITE DERMATITIS	17MAY2014	.	.	N	PROBABLE	RELATED	NOT RECOVERED/NOT RESOLVED
01-710-1077	APPLICATION SITE DERMATITIS	12DEC2013	12DEC2013	1	N	POSSIBLE	RELATED	RECOVERED/RESOLVED
01-713-1256	APPLICATION SITE DERMATITIS	30OCT2012	31OCT2012	2	N	PROBABLE	RELATED	NOT RECOVERED/NOT RESOLVED
	APPLICATION SITE DERMATITIS	30OCT2012	31OCT2012	2	N	PROBABLE	RELATED	RECOVERED/RESOLVED
01-718-1355	APPLICATION SITE DERMATITIS	27APR2013	27APR2013	1	N	PROBABLE	RELATED	RECOVERED/RESOLVED
	APPLICATION SITE DERMATITIS	12MAY2013	12MAY2013	1	N	PROBABLE	RELATED	RECOVERED/RESOLVED
	APPLICATION SITE DERMATITIS	16MAY2013	16MAY2013	1	N	PROBABLE	RELATED	RECOVERED/RESOLVED
	APPLICATION SITE DERMATITIS	20MAY2013	20MAY2013	1	N	PROBABLE	RELATED	RECOVERED/RESOLVED

Output 1: Subject Level Listing

Program 1 below shows how to produce the subject level listings illustrated in Output 1.

```
/* Generate the supporting listing for each preferred term by treatment */
%macro genlist;
   %let x = 1;
   %let upttrt = %scan(&allpttrt, &x, '#');  (1)

   %do %while (&upttrt ne   );
      proc sort data = adae
                 out = adae___&x;            (2)
         where pttrtsev = "&upttrt";
         by AEBODSYS AEDECOD AESEV TRTA USUBJID ASTDT AENDT;
      run;

      ods html path = outp file = "&upttrt..html";   (3)
      title color=red "Click back arrow on navigation bar to return to severity bar chart";
      proc report data = adae___&x;
         by AEBODSYS AEDECOD AESEV TRTA;
         columns USUBJID AETERM ASTDT AENDT ADURN AESER AEREL AREL AEOUT;
         define USUBJID  / group;
         break after USUBJID / skip;
      run;
      ods html close;

      %let x = %eval(&x + 1);   (4)
      %let upttrt = %scan(&allpttrt, &x, '#');
   %end;
%mend genlist;

%genlist
```

Program 1: Code to Produce Subject Level Listings

(1) For this example, the **allpttrt** macro variable contains the 53 combinations of the AE name, treatment group, and AE severity. Here the **upttrt** macro variable value consists of the first **allpttrt** combination,

which is the value **application_site_dermatitis_MILD_0**. Where application_site_dermatitis is the name of the AE, MILD is the severity, and 0 means the Placebo treatment group.

2. Filters the ADAE data set to only the combination of interest, and then outputs that filtered data set.

3. Produces HTML file of the subject level listing for the combination of interest.

4. Increments the value of **x** so that the scan function can extract each of the 53 combinations during the do loop process, and therefore 53 subject level listings can be outputted.

PRODUCING GRAPHS OF AE COUNTS, FOR EACH TREATMENT AND SEVERITY GROUP

Figure 1 below is an example of a graph showing the number of subjects with at least one AE for each treatment and severity group. This particular graph shows the Number of Placebo Subjects for the Application Site Dermatitis AE.

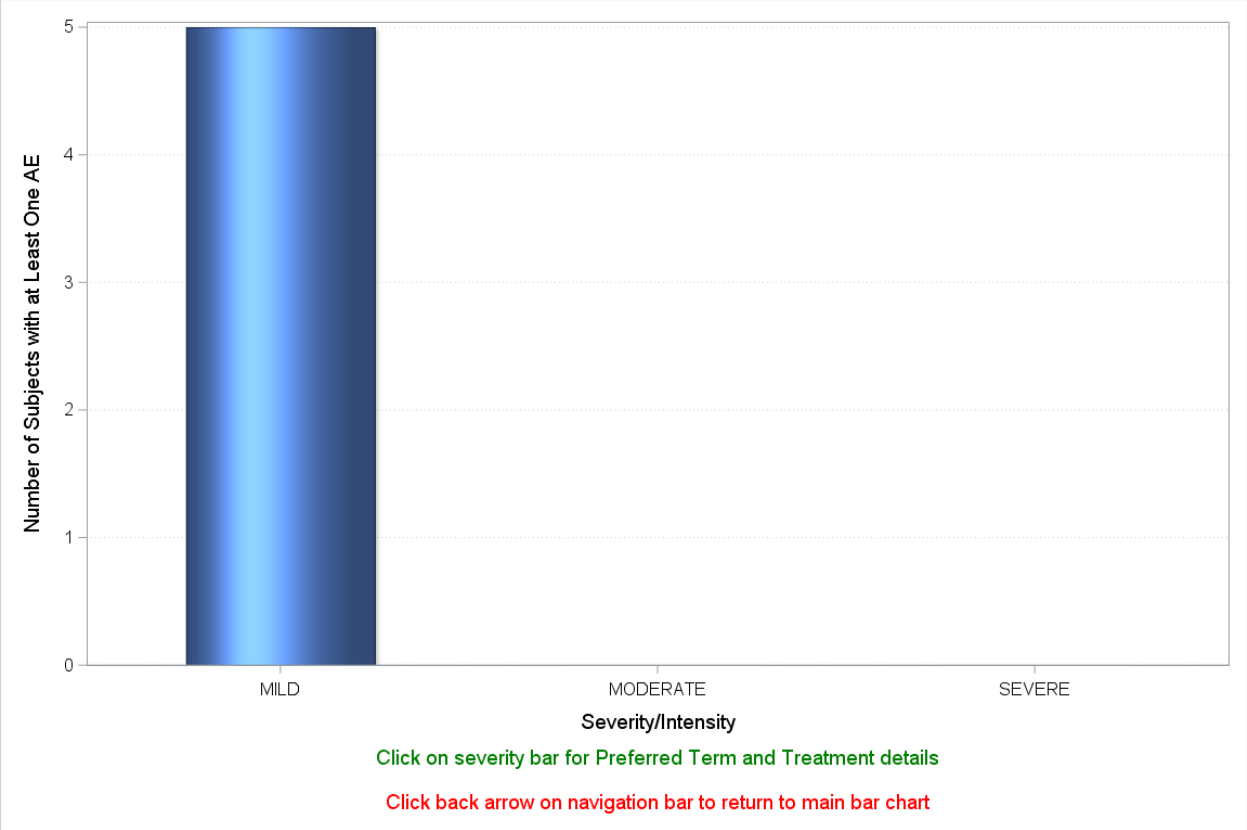


Figure 1: Number of Placebo Subjects with at Least One Application Site Dermatitis by Severity

Program 2 below shows how to produce graphs of the AE counts, for each AE, treatment and severity group illustrated in Figure 1.

```
/* generate a supporting graph for each preferred term */
%macro genchart(trt = , color =);
   /* create a template for the supporting AE graph - within in macro so bar colors
can align with main graph */
   proc template;
      define statgraph sevfreq&trt;
         begingraph;
            entryfootnote textattrs = (size = 12 pt color = green) "<Text1>";
            entryfootnote " ";
            entryfootnote textattrs = (size = 12 pt color = red) "<Text2>";
            layout overlay / yaxisopts = (label = "<Label1>"
                                          griddisplay = on
                                          gridattrs = (color = lightgray
                                          pattern = dot));
               barchart category = AESEV response = COUNT / barwidth = 0.5
                                                            fillattrs =
                                                           (color = &color)
                                                            dataskin = sheen
                                                            url = url1;
            endlayout;
         endgraph;
       end;
   run;

   %let y = 1;
   %let upt = %scan(&allprefterm, &y, '#');

   %do %while (&upt ne   );

      %if %sysfunc(find(&upt, &trt)) > 0 %then %do;
         /* Specify the image output filename. */
         ods graphics / reset imagemap = on imagename = "&upt"
           drilltarget = "_self" height = 8in width = 12in;

         /* Generate the graph using ODS HTML. */
         ods html path = outp file = "&upt..html";
         proc sgrender data = adaesev template = sevfreq&trt;
            where pttrt = "&upt";
            by AEDECOD TRTA;
         run;
         ods html close;
      %end;

      %let y = %eval(&y + 1);
      %let upt = %scan(&allprefterm, &y, '#');
   %end;
%mend genchart;

options mprint mlogic;

%genchart(trt = 0,  color = cornflowerblue)
%genchart(trt = 54, color = indianred)
%genchart(trt = 81, color = cadetblue)
```

Program 2: Code to Produce Graphs of AE Counts

1. Creates the GTL needed to produce the bar charts. Due to space limitations in the above code, the details regarding the text in <Text1>, and <Text2> can be seen in the Appendix.

2. The URL option indicates that each bar on the bar chart has an active link, and the hyperlink is the value in the **URL1** variable. When the bar is clicked you will then be navigated to the respective subject level listing.

3 The IMAGEMAP = ON option enables the data tips and drill-down options to work.

4 Creates the HTML file of the bar chart of AE counts, for each of the 53 combinations of AE, treatment and severity group in this example.

5 Since an output needs to be produced for each of the AE and severity level combinations, then when associating the AE data with the template, we need to ensure we are referencing the correct template and subsetting the data for the appropriate set of records.

PRODUCING MAIN BAR CHART

Figure 2 shows the main bar chart of the percentages of subjects with at least one AE, by AE and treatment group.

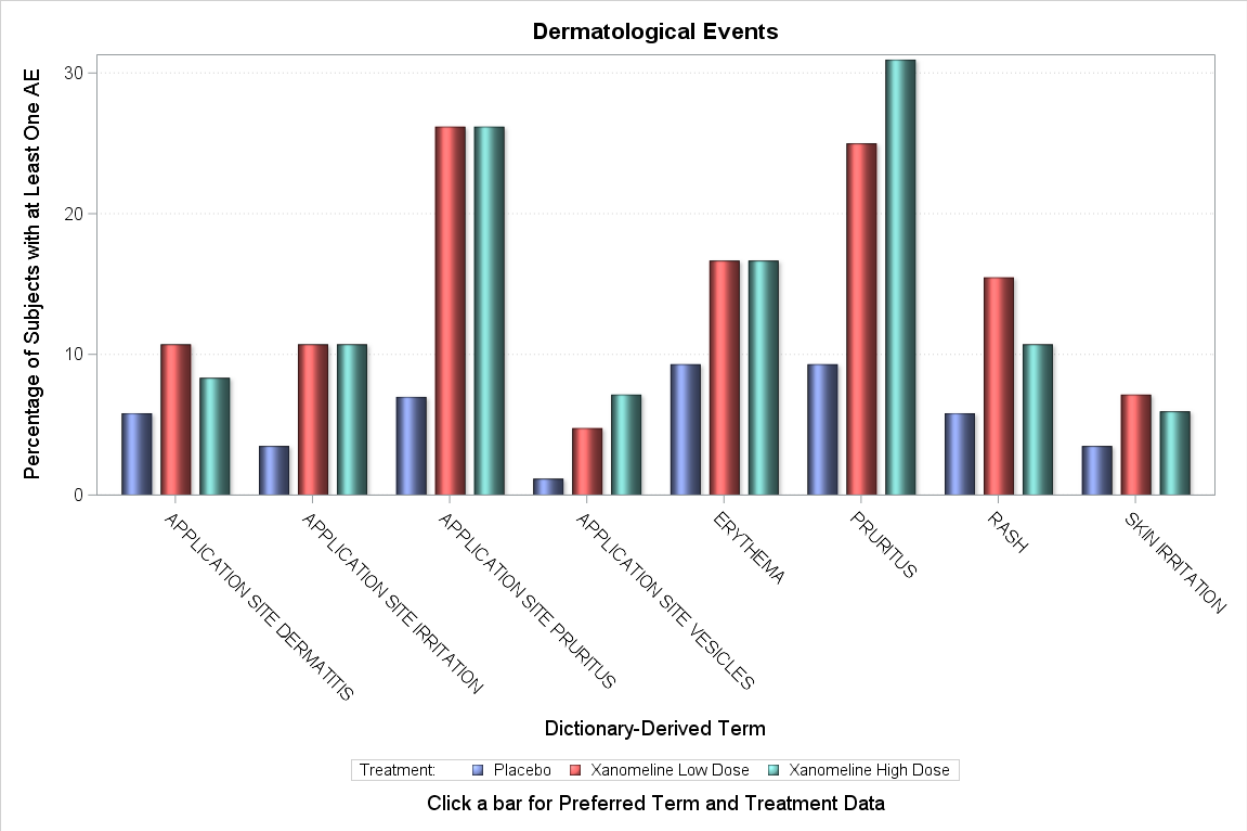


Figure 2: Main Bar Chart

Program 3 creates the main bar chart of the percentages of subjects with at least one AE, by AE and treatment group shown in Figure 2.

```
proc template;
   define statgraph ptfreq;
      begingraph;
         entrytitle "Dermatological Events";
         entryfootnote textattrs = (size = 12 pt) "<Text1>";
         layout overlay / yaxisopts = (label = "<Text2>"
                                       griddisplay = on
                                       gridattrs = (color = lightgray pattern
                                         = dot));
            barchart category = AEDECOD response = PERCENT / name =
                                                              "dermevent"
                                                             group = TRTA
                                                             groupdisplay =
                                                             cluster
                                                             barwidth = 0.75
                                                             dataskin = sheen
                                                             url = url0;
            discretelegend "dermevent" / title = "Treatment:";
         endlayout;
      endgraph;
    end;
run;

/* Enable image mapping in the HTML output and specify a base image name */
ods graphics / reset imagemap = on imagename = "aes" drilltarget = "_top" height =
8in width = 12in;

/* Generate the drill-down graph using ODS HTML */
ods html path = outp file = "aes.html";
proc sgrender data = ptcnt2 template = ptfreq;
run;
ods html close;
```

(1)

(2)

Program 3: Code for Producing Main Bar Chart of Percentages of Subjects with at least one AE, by AE and Treatment.

(1) The variable URL0 is used in the URL option. This variable contains the values of the URL name associated with each AE and treatment group by severity bar charts, and when you click on the associated bar you are then navigated to the appropriate severity bar charts, which are created in Program 2.

(2) This template is different from the template in Program 2, because this template produces a bar chart of the percentage of subjects with at least one AE, by AE and treatment. Therefore, we need to make sure we are using the correct template and that we are not subsetting the data since this is the main bar chart.

CREATING THE INPUT DATA SET

Now that the templates and macros have been created to produce the necessary output, we have an idea of how many URLs are needed and how these URLs will look. We can then make sure that our input data set has the correct variables that will hold these URLs that are needed so that we can build our supporting listings and graphs and have our drill-down effect. Refer to Program 4 for illustration of how the URL values were created and stored in the input data set.

```
/* add a URL column for the drill-down links to the Adverse Event data set */

data adae;
   length url0 url1 pttrt pttrtsev $500;
   set ADAM.ADAE;

   /* need to change the spaces to underscores to build URLs */
   PREFTERM = lowcase(tranwrd(tranwrd(strip(AEDECOD), ' ', '_'), '-', '_'));

   /* build URLs for each preferred term */
   pttrt = catx('_', PREFTERM, TRTAN);
   url0 = cats(pttrt, '.html');

   /* build URLs for each preferred term by relation */
   pttrtsev = catx('_', PREFTERM, AESEV, TRTAN);
   url1 = cats(pttrtsev, '.html');
run;
```

Program 4: Building the Input Data Set with URL Values

LAB DRILL-DOWN EXAMPLE

Like adverse events, being able to obtain additional information regarding portions of the graph gives better insight into the data. The following link shows a Laboratory plot, which you can drill down and further investigate the data - https://www.krissharris.co.uk/sgf/2019/3261/output/labs/ALT_AVAL.html.

In order to achieve the drill-down ability, we need to create templates that will be used to produce output of all the underlying reports (i.e., supporting graphs and/or supporting listings). The only limit in the number of supporting outputs is based on how much detail you want. When the templates are rendered to produce the supporting outputs, each of the supporting outputs need to be saved as an HTML file. It is ideal to give each file a consistent naming convention so that it can be easily referenced within the parent output.

After you have decided on a naming convention for each of the supporting outputs, then a data set can be created that will contain variables that will be used to store the URLs associated with each HTML file. Note that if there are different types of supporting outputs then creating separate variables to capture the URL for each type of supporting output is preferred.

Understanding what the end goal in mind will help to build the necessary templates and the data set. It is best to map out what the expectations are of the desired graph. For example, with the lab data we want to be able to:

- Select from one of three lab parameters (i.e., Alanine Aminotransferase, Aspartate Aminotransferase and Gamma Glutamyl Transferase) as well as select what data value to use (i.e., the analysis value or the change from baseline)
- Create a series plot of the means with +/- one standard deviation ‘whiskers’ over time for each treatment once we determine the parameter and data value
- From the series plots, have the ability to either
    - select a mean value at a time point for a treatment and see all the underlying data as a listing, or
    - select the time point to see boxplots overlaid with scatterplots by treatment
- From the boxplots, have the ability to select a treatment and see all the underlying data as a listing.

Essentially, we should be able to get to the listing of data for a time point and treatment from either the series plot or from the boxplots. Based on the list of desired outcomes, this graph has several layers of interaction.

CREATE THE SUPPORTING LISTINGS

Based on the criteria listed above we can determine that we need supporting listings for both the analysis value and change from baseline for each lab parameter at each time point and treatment combination. In our example, we have three lab parameters, three treatments and five time points which indicates we need 45 (i.e., 3 * 3 * 5) supporting listings. The number of listings can add up quickly and trying to create these one at a time can be cumbersome. Writing a macro to loop through the data to help create these listings is beneficial and will also help with creating a consistent naming convention for the files. Although we are creating listings, they should still be saved as HTML files so that they can be referenced in the parent output.

To help with the generation of all the listings a temporary variable (PRM_VIS_TRT) was created that concatenated the parameter code (PARAMCD), analysis visit number (AVISITN) and the treatment code (TRTAN) separated by an underscore (refer to Data Display 1). PRM_VIS_TRT is used as the filename for each supporting listing.

This allowed for the creation of a macro variable (refer to Program 5 that contained all the possible PRM_VIS_TRTs concatenated separated with '#' so that a macro could be created that would loop through each filename and extract only the records associated with that file name (refer to Program 6). The filename was also used to build the URL which would be used in the parent graph to link the point on the graph to the supporting listing.

PARAMCD	AVISITN	TRTAN	PRM_VIS_TRT	URL1
ALT	0	0	ALT_0_0	ALT_0_0.html
ALT	0	54	ALT_0_54	ALT_0_54.html
ALT	0	81	ALT_0_81	ALT_0_81.html
ALT	2	0	ALT_2_0	ALT_2_0.html
ALT	2	54	ALT_2_54	ALT_2_54.html
ALT	2	81	ALT_2_81	ALT_2_81.html
ALT	4	0	ALT_4_0	ALT_4_0.html
ALT	4	54	ALT_4_54	ALT_4_54.html
ALT	4	81	ALT_4_81	ALT_4_81.html
ALT	6	0	ALT_6_0	ALT_6_0.html
ALT	6	54	ALT_6_54	ALT_6_54.html
ALT	6	81	ALT_6_81	ALT_6_81.html
ALT	8	0	ALT_8_0	ALT_8_0.html
ALT	8	54	ALT_8_54	ALT_8_54.html
ALT	8	81	ALT_8_81	ALT_8_81.html

Data Display 1: Creation of PRM_VIS_TRT

```
/* create a macro variable of all unique url values so that we can loop through */
proc sql noprint;
   select distinct prm_vis,
                   prm_vis_trt
          into :allprm_vis          separated by "#",
               :allprm_vis_trt      separated by "#"
   from adlb;
quit;
```

Program 5: Create Macro Variables to Loop Through Data to Create Supporting Outputs

```
/* generate the supporting listing for each lab test by time and treatment */
%macro genlist;
   %let x = 1;
   %let uprm = %scan(&allprm_vis_trt, &x, '#');

   %do %while (&uprm ne   );
      %let visnum = %scan(&uprm, 2, '_');
      proc sort data = adlb
                 out = adlb___&x;
         where prm_vis_trt = "&uprm";
         by PARAM AVISIT TRTA USUBJID ADT;
      run;

      ods html path = outp file = "&uprm..html";
      title color = red "Click back arrow on navigation bar to return to previous output";
      proc report data = adlb___&x;
         by PARAM AVISIT TRTA;
         columns USUBJID ADT AVAL ANRIND
             %if &visnum ne 0 %then %do; BASE CHG BNRIND %end; A1LO A1HI;
         define USUBJID  / display;
         define ANRIND   / display '';
         %if &visnum ne 0 %then %do;
            define BNRIND   / display '';
         %end;
      run;
      ods html close;

      %let x = %eval(&x + 1);
      %let uprm = %scan(&allprm_vis_trt, &x, '#');
   %end;
%mend genlist;

%genlist
```

Program 6: Creation of Supporting Listings for Each Parameter, Analysis Visit and Treatment

(1) Utilizing the macro variable that contains all the possible filenames (i.e., PRM_VIS_TRT values), we can extract each filename individually and set up a %DO %WHILE to loop through all the possible filenames.

(2) Since the filename was created as a temporary variable in the source data set, it can be used to subset the data for the desired records.

(3) Prior to creating the listings, the appropriate ODS destination must be opened and any titles or footnotes should be specified. Each listing will be linked to the parent output using a URL so the output must be HTML. The titles and footnotes can contain instructions on how to navigate through the various drill-down outputs. Once the output is produced the destination should be closed.

(4) Using the data subset created in (2) the REPORT procedure is used to produce the actual supporting listing. Note that for AVISITN = 0 the variables associated with baseline (BASE, CHG and BNRIND) are not displayed since they would not provide any additional information.

Output 2 shows two samples for supporting listings that are generated.

Click back arrow on navigation bar to return to previous output

Parameter=Alanine Aminotransferase (U/L) Analysis Visit=Baseline Actual Treatment=Placebo

Unique Subject Identifier	Analysis Date	Analysis Value		Analysis Range 1 Lower Limit	Analysis Range 1 Upper Limit
01-701-1015	26DEC2013	27		6	34
01-701-1023	22JUL2012	23		6	43
01-701-1047	22JAN2013	22		6	32
01-701-1118	27FEB2014	15		6	43
01-701-1130	09FEB2014	15		6	35
01-701-1153	06SEP2013	13		6	32

Click back arrow on navigation bar to return to previous output

Parameter=Alanine Aminotransferase (U/L) Analysis Visit=Week 2 Actual Treatment=Placebo

Unique Subject Identifier	Analysis Date	Analysis Value		Baseline Value	Change from Baseline		Analysis Range 1 Lower Limit	Analysis Range 1 Upper Limit
01-701-1015	16JAN2014	41		27	14		6	34
01-701-1023	27AUG2012	30		23	7		6	43
01-701-1047	25FEB2013	16		22	-6		6	32
01-701-1118	26MAR2014	7		15	-8		6	43
01-701-1130	01MAR2014	14		15	-1		6	35
01-701-1153	08OCT2013	14		13	1		6	32

Output 2: Sample Supporting Listings for Analysis Visit 0 (Baseline) and Analysis Visit 2 (Week 2)

CREATE THE SUPPORTING GRAPHS

The supporting graph that needs to be created is a boxplot with a scatterplot overlaid for each treatment. Since the output contains all three treatments on one graph, there is no need to have an individual output at the treatment level. Thus, we only need to create supporting graphs for each parameter and analysis visit but we also need to consider the data type. In the supporting listings we did not consider the data type, but for the graphs we do.

Notice that in the supporting listings, the analysis value as well as the change from baseline is included thus there is no need to distinguish between data types for the listing. However, for the boxplot and scatterplot graphs we need to produce this using either analysis value or change from baseline based on the what was used to produce the parent series plot.

In our example, we have two data types, three parameters and 5 analysis visits which gives us 30 (i.e., 2 * 3 * 5) supporting graphs. However, for change from baseline we do not have supporting graphs for baseline since change from baseline for baseline is 0 and has no meaningful use. Thus, for our example we only have 27 supporting graphs.

Like the supporting listings, we will use a macro to loop through the data to create the boxplot with scatterplot overlays graphs. But we need to create a new temporary variable that will contain the filenames for these supporting graphs. To allow for consistency in the filename we concatenate the parameter code, the analysis visit and data type (i.e., AVAL or CHG) and these aforementioned values are separated by an underscore (refer to Data Display 2).

PARAMCD	AVISITN	PRM_VIS_AVAL	URL1AVAL
ALT	0	ALT_0_AVAL	ALT_0_AVAL.html
ALT	2	ALT_2_AVAL	ALT_2_AVAL.html
ALT	4	ALT_4_AVAL	ALT_4_AVAL.html
ALT	6	ALT_6_AVAL	ALT_6_AVAL.html
ALT	8	ALT_8_AVAL	ALT_8_AVAL.html
AST	0	AST_0_AVAL	AST_0_AVAL.html
AST	2	AST_2_AVAL	AST_2_AVAL.html
AST	4	AST_4_AVAL	AST_4_AVAL.html
AST	6	AST_6_AVAL	AST_6_AVAL.html
AST	8	AST_8_AVAL	AST_8_AVAL.html
GGT	0	GGT_0_AVAL	GGT_0_AVAL.html
GGT	2	GGT_2_AVAL	GGT_2_AVAL.html
GGT	4	GGT_4_AVAL	GGT_4_AVAL.html
GGT	6	GGT_6_AVAL	GGT_6_AVAL.html
GGT	8	GGT_8_AVAL	GGT_8_AVAL.html

PARAMCD	AVISITN	PRM_VIS_CHG	URL1CHG
ALT	2	ALT_2_CHG	ALT_2_CHG.html
ALT	4	ALT_4_CHG	ALT_4_CHG.html
ALT	6	ALT_6_CHG	ALT_6_CHG.html
ALT	8	ALT_8_CHG	ALT_8_CHG.html
AST	2	AST_2_CHG	AST_2_CHG.html
AST	4	AST_4_CHG	AST_4_CHG.html
AST	6	AST_6_CHG	AST_6_CHG.html
AST	8	AST_8_CHG	AST_8_CHG.html
GGT	2	GGT_2_CHG	GGT_2_CHG.html
GGT	4	GGT_4_CHG	GGT_4_CHG.html
GGT	6	GGT_6_CHG	GGT_6_CHG.html
GGT	8	GGT_8_CHG	GGT_8_CHG.html

Data Display 2: Creation of PRM_VIS_AVAL and PRM_VIS_CHG

The boxplot does not have the capability to link to a supporting output. However, we would like the ability to do this. It is possible to drill down from the boxplot by using the scatterplot overlay, however this is not ideal, because this would create URL links from every possible data point on the scatter plot. Ideally, we would like to have one URL that is tied to the entire portion of the graph that encompasses the desired boxplot and corresponding scatterplot. Therefore, we need to utilize another type of plot that will allow for this feature. The HIGHLOWPLOT does have an option for using URLs. To leverage the use of the HIGHLOWPLOT, we need to determine the minimum and maximum value for each data type, parameter and treatment (refer to Data Display 3 and Data Display 4).

USUBJID	TRTAN	AVISITN	PARAMCD	AVAL	MIN	MAX
01-701-1015	0	0	ALT	27	7	69
01-701-1023	0	0	ALT	23	7	69
01-701-1047	0	0	ALT	22	7	69
01-701-1118	0	0	ALT	15	7	69
01-701-1130	0	0	ALT	15	7	69
01-701-1153	0	0	ALT	13	7	69
01-701-1203	0	0	ALT	13	7	69
01-701-1234	0	0	ALT	18	7	69
01-701-1345	0	0	ALT	13	7	69
01-701-1363	0	0	ALT	9	7	69
01-701-1387	0	0	ALT	12	7	69
01-701-1392	0	0	ALT	23	7	69

Data Display 3: Determining MIN and MAX for Each Parameter and Treatment for Analysis Value

USUBJID	TRTAN	AVISITN	PARAMCD	CHG	MIN	MAX
01-701-1015	0	2	ALT	14	-14	54
01-701-1023	0	2	ALT	7	-14	54
01-701-1047	0	2	ALT	-6	-14	54
01-701-1118	0	2	ALT	-8	-14	54
01-701-1130	0	2	ALT	-1	-14	54
01-701-1153	0	2	ALT	1	-14	54
01-701-1203	0	2	ALT	1	-14	54
01-701-1234	0	2	ALT	12	-14	54
01-701-1345	0	2	ALT	-4	-14	54
01-701-1363	0	2	ALT	2	-14	54
01-701-1387	0	2	ALT	0	-14	54
01-701-1392	0	2	ALT	-3	-14	54

Data Display 4: Determining MIN and MAX for Each Parameter and Treatment for Change from Baseline Value

After the necessary data components are determined, the template can be created that will produce the desired output with the necessary link to the supporting listing.

```
/* create a template for the supporting lab graphs */
   proc template;
      define statgraph drill&var; (1)
         begingraph;
            dynamic _byval2_ _byval4_;
            entrytitle halign = center _byval2_ " at " _byval4_;
            entryfootnote textattrs = (size = 11 pt color = green) "Click on Boxplot
for Lab Result Details";
            entryfootnote " ";
            entryfootnote textattrs = (size = 11 pt color = red) "Click back arrow on
navigation bar to return to previous output";
            layout overlay / yaxisopts = (griddisplay = on
                                          gridattrs = (color = lightgray
                                                       pattern = dot));

               /* produce boxplot */
               boxplot x = TRTA y = &var / tip = (none)
                                           extreme = false;

               /* produce scatterplot */
               scatterplot x = TRTA y = &var / jitter = auto;

               /* produce highlow plot */
               highlowplot x = TRTA low = min high = max / type = bar
                                                           display = (fill)
                                                           datatransparency = 1 (3)
                                                           url = url1
                                                           rolename = (url = url1)
                                                           tip = (url);

            endlayout;
         endgraph;
      end;
   run;
```

(2) (4)

Program 7: Template to Produce Boxplot with Scatter Plot Overlay and Transparent High-low Plot for Link to Supporting Listing

(1) Because the source data value can be either the analysis value (AVAL) or the change from baseline (CHG), a different template is needed to reference the correct variable. Thus, the template name is based on the variable name so that a template for each can be generated.

(2) In order to identify each supporting group, the DYNAMIC statement along with ENTRYTITLE statement is used. In addition, ENTRYFOOTNOTE statements can be used to provide navigation instructions.

(3) The purpose of the high-low plot is to allow for one URL link to the supporting listing, therefore it is not necessary for the plot to be visible. The URL feature is the only desired feature that needs to be utilized, therefore, the plot was made transparent, so it does not appear in the graph.

(4) Recall that we created a variable with the URL name in Data Display 1. To indicate the URL that should be used, the URL = option allows for the specification of the variable that contains the correct URL. In addition, ROLENAME option allows for the defining of user-defined roles which can be used in the TIP option. The TIP option indicates what is to be displayed when the mouse hovers over the data. In this example, we want to display the URL that links to the supporting listing.

Once the template is defined, the data then needs to be associated with it. Similar to what was done when producing the supporting listings (refer to Program 5 and Program 6), a macro variable is created that contains all the possible filenames so that the filename for each supporting graph can be extracted in a %DO %WHILE loop and the corresponding graph is produced (refer to Program 8).

```
%macro genbox;
   %let y = 1;
   %let uprmvis = %scan(&&&allprm_vis_&var, &y, '#');

   %do %while (&uprmvis ne   );
      ods graphics / reset imagemap = on imagename = "&uprmvis"
                (1) drilltarget = "_self" height = 8in width = 12in;

      ods html path = outp file = "&uprmvis..html";          (2)
      proc sgrender data = adlb_&var template = drill&var;
         where prm_vis_&var = "&uprmvis";
         by PARAMCD PARAM AVISITN AVISIT;
      run;
      ods html close;

      %let y = %eval(&y + 1);
      %let uprmvis = %scan(&&&allprm_vis_&var, &y, '#');
   %end;
%mend genbox;
```

Program 8: Associating Boxplot and Scatter Plot Template with the Data

(1) By default, any drill-down output will open in a new browser window. Specifying DRILLTARGET = "_self" indicates that we want the drill-down output to open in the same window. Note that when specifying the target window for the drill-down output, the value must be in lower case and enclosed in parentheses. Possible values for DRILLTARGET are "_blank" (default value), "_top", "_parent", "_self" and "*frame-name*"**.**

(2) The height and width of the graph can be adjusted to the desired size.

Figure 3 is an illustration of what the supporting graph looks like.

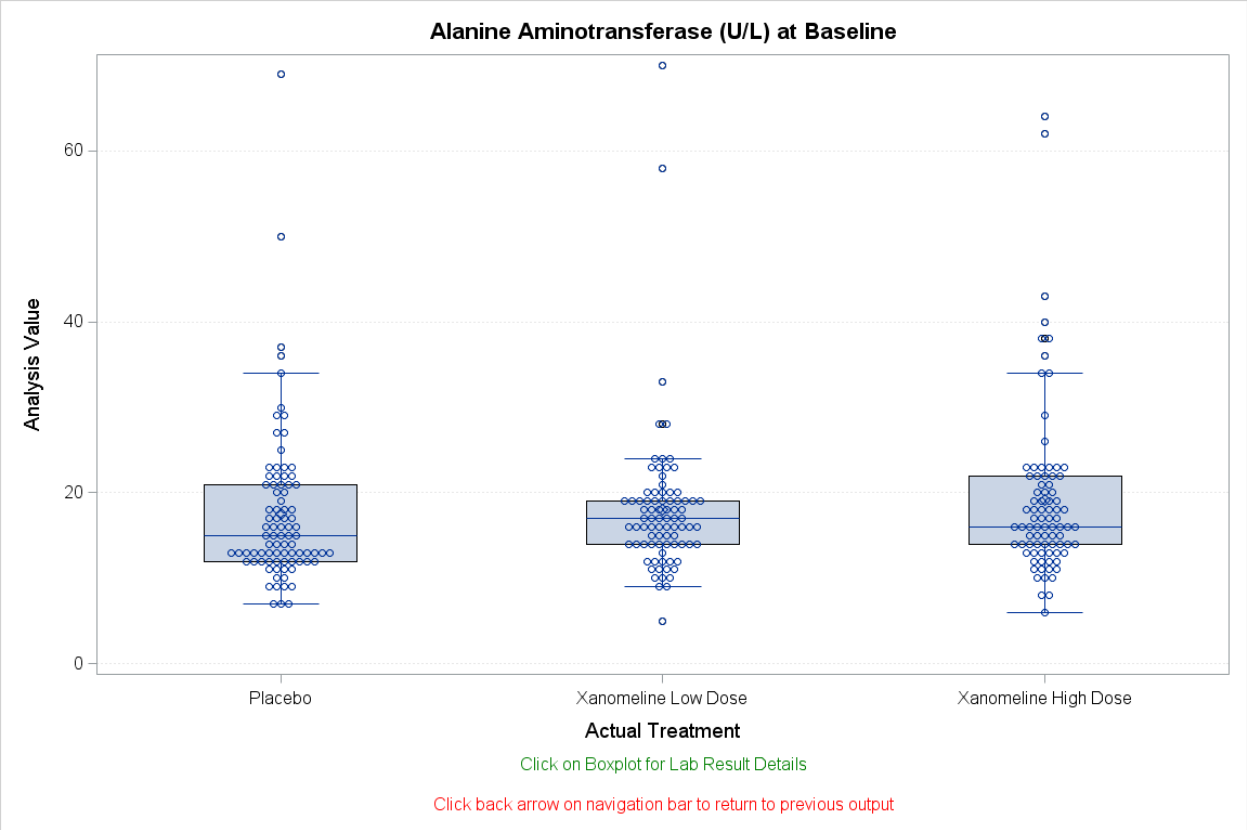


Figure 3: Sample Supporting Graph of Boxplot with Scatter Plot Overlay for Each Treatment for the Indicated Data Type, Parameter and Analysis Visit

CREATE THE MAIN GRAPH

After all the supporting outputs have been produced, a template then needs to be developed that will produce the main graph. In our example, the main graph is the series plot over time for each parameter and treatment combination for the indicated data type. Because there are two data types and three parameters, that is 6 graphs.

Again, we use the macro facility to help build templates for each of the main graphs (refer to Program 9).

```
   /* create a template for the main lab graph */
   proc template;
      define statgraph series&var;
         begingraph;
            entrytitle "&varlabel";
            entryfootnote textattrs = (size = 11 pt color = green) "Click a Visit
Column for Boxplot of Data";
            entryfootnote " ";
            entryfootnote textattrs = (size = 11 pt color = green) "Click on Mean for
Lab Result Details";
            entryfootnote " ";
            entryfootnote textattrs = (size = 11 pt color = red) "Click back arrow on
navigation bar to return to previous output";
            layout overlay / xaxisopts = (label = "Analysis Visit"
                                          type = discrete
                                          discreteopts = (tickvaluelist =
                                                         (&base '2' '4' '6' '8')))
                             yaxisopts = (label = "Mean &varlabel"
                                          type = linear
                                          linearopts = (viewmin = &min
```

```
                                                    viewmax = &max)
                                        griddisplay = on
                                        gridattrs = (color = lightgray
                                                     pattern = dot));

            highlowplot x = AVISITN low = min high = max / type = bar
                                                           display = (fill)
                                                           url = url1&var
                                                           datatransparency = .95 (1)
                                                           rolename = (url=url1&var)
                                                           tip = (url);

            seriesplot x = AVISITN y = mean / display = all
                                              group = TRTA
                                              name = "trt";

            scatterplot x = AVISITN y = mean / group = TRTA
                                          (2) url = url1
                                              yerrorupper = eval(mean + se)
                                              yerrorlower = eval(mean - se);

            discretelegend "trt" / title = "Treatment:";
         endlayout;
      endgraph;
   end;
run;
```

Program 9: Template to Produce Series Plot Over Time for Each Data Type and Parameter

(1) In the supporting graph, we set the high-low plot to be 100% transparent because we did not want it to show up on the graph. We only wanted to utilize the URL feature. However, for the main graph, we want the high-low plot to be semi-transparent. This will create vertical bars over the visit tick mark so that the user can click anywhere in that space to get to the supporting graph.

(2) One of the desired criteria for the graph is to be able to get to the data corresponding to the data listing from the main graph (i.e., bypass the supporting graph). Since SCATTERPLOT does have the URL option, we are able to use the SCATTERPLOT to create both the desired whisker plots as well as link to the supporting listing.

Note that the titles and footnotes can be used to provide instructions on navigation.

To associate the data with the template for the main graph, we would follow the same approach as illustrated in Program 8.

Figure 4 illustrates the series plot for the analysis value for the lab parameter Alanine Animotransferase and what would be displayed if we hover over the semi-transparent high-low plot. Notice that if we hover over the semi-transparent high-low plot it shows the name of the URL for the supporting output.

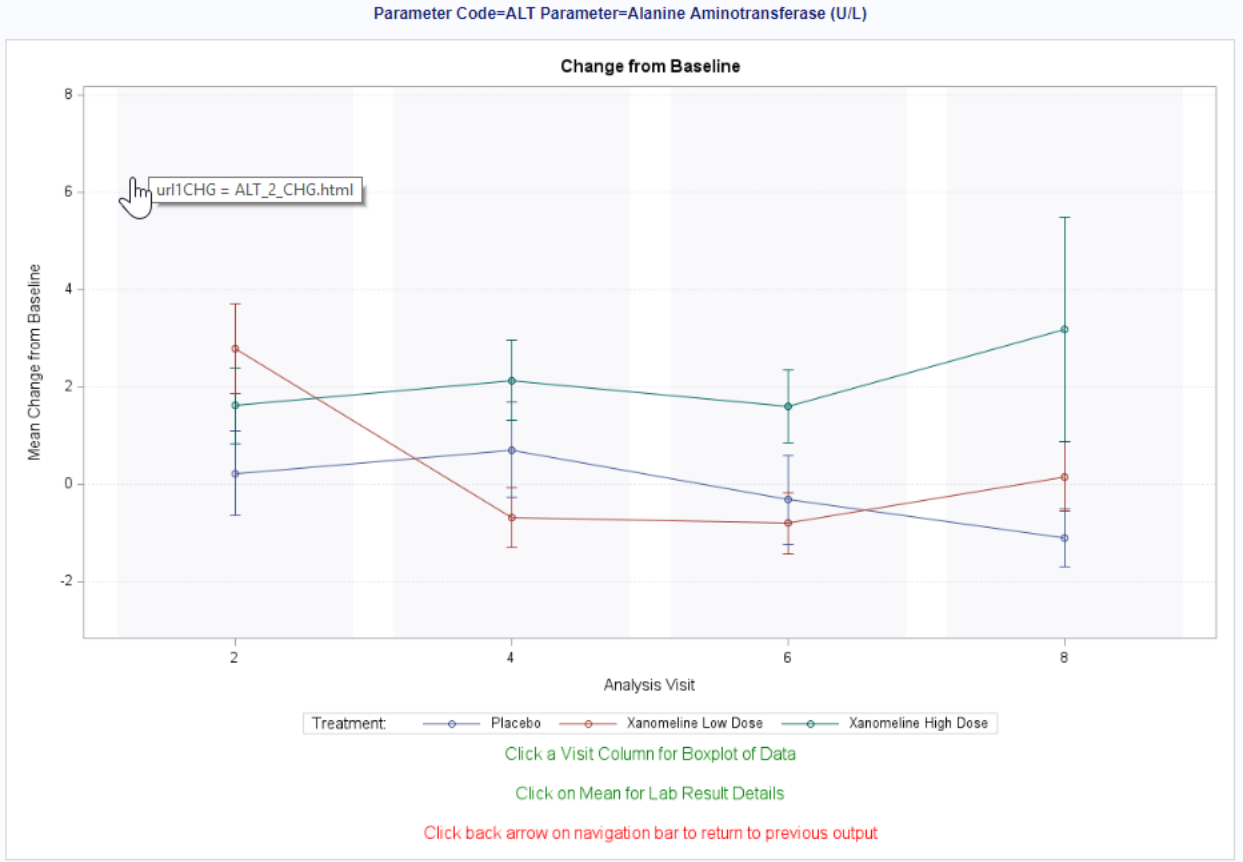


Figure 4: Sample Main Graph of Series Plot Over Time for Each Treatment for the Analysis Value and Indicated Parameter

Notice that for Figure 5 that when we hover over the actual mean value data point, that instead of the corresponding URL for the supporting listing being displayed, it shows the analysis visit value, the treatment value and the mean change from baseline value. If the TIP option is not specified, then it will display the default values.

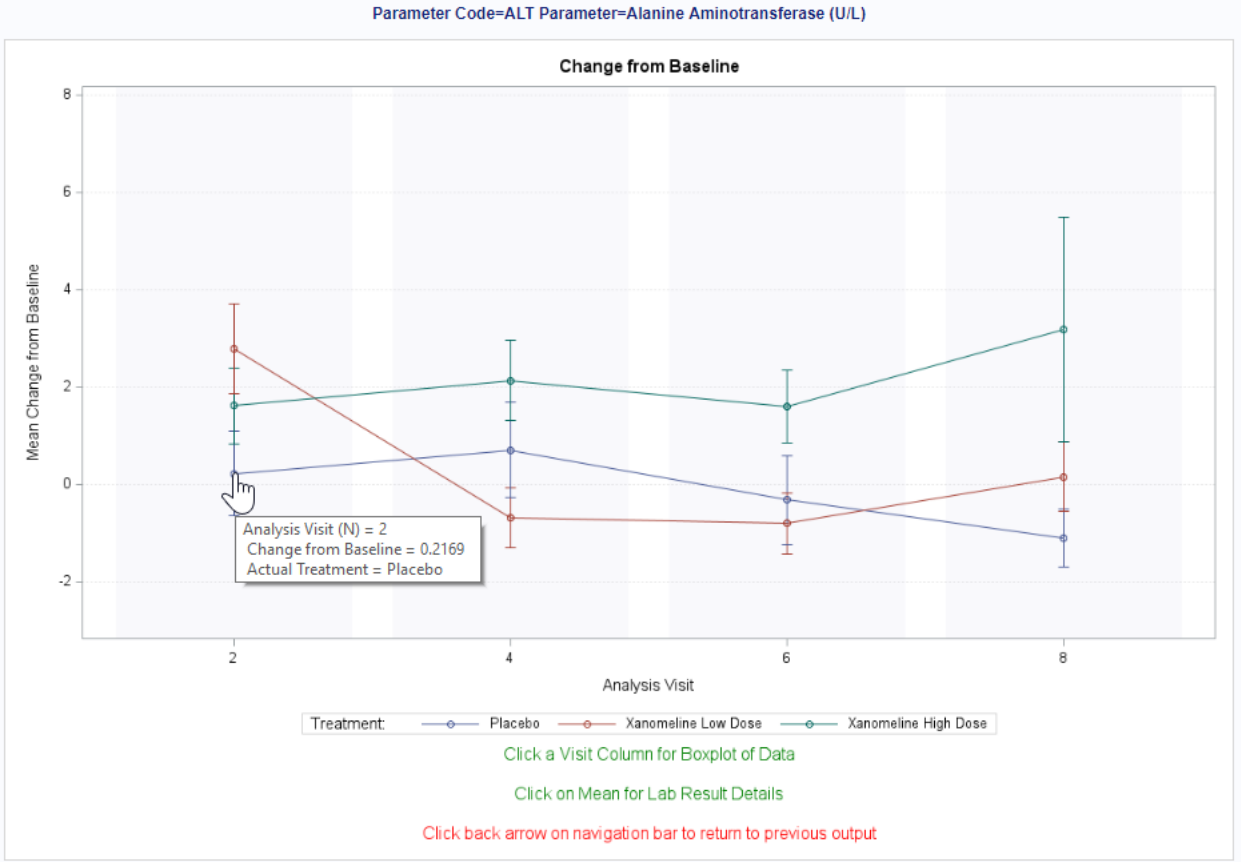


Figure 5: Sample Main Graph of Series Plot Over Time for Each Treatment for Change from Baseline and Indicated Parameter

BUILDING THE SELECTION MENU

So now we have built all the supporting output and the main graphs, how do we select the ones we want to see without having to find the main graph of interest? We can build a menu. To build a menu we can create a data set with the necessary information (refer to Data Display 5). The menu can then be associated with a template that is created using BLOCKPLOT and TEXTPLOT (refer to Program 10).

PARAM	PARAMCD	ORD	VAR	VARLBL	URLMENU
Gamma Glutamyl Transferase (U/L)	GGT	1	AVAL	Analyis Value	GGT_AVAL.html
Gamma Glutamyl Transferase (U/L)	GGT	1	CHG	Change from Baseline	GGT_CHG.html
Aspartate Aminotransferase (U/L)	AST	2	AVAL	Analyis Value	AST_AVAL.html
Aspartate Aminotransferase (U/L)	AST	2	CHG	Change from Baseline	AST_CHG.html
Alanine Aminotransferase (U/L)	ALT	3	AVAL	Analyis Value	ALT_AVAL.html
Alanine Aminotransferase (U/L)	ALT	3	CHG	Change from Baseline	ALT_CHG.html

Data Display 5: Menu to Select Parameter and Data Type

```
/* create a template that will be used to select the parameter and variable for
review */
proc template;
   define statgraph textplot;
      begingraph;
         entrytitle textattrs = (size = 18 pt) "Mean of Analysis Value or Change from
Baseline for Indicated Parameters";
         entrytitle " ";
         entrytitle textattrs = (size = 16 pt color = green) "Select Type of Analysis
and Parameter";
         entrytitle " ";
         entrytitle " ";
         layout overlay / xaxisopts = (display = none
                                       label = " ")
                          yaxisopts = (display = none
                                       label = " "
                                       linearopts = (viewmin = 0
                                                     viewmax = 4))
                          walldisplay = none;

            blockplot x = VAR block = varlbl / valuehalign = center
                                               valuevalign = top
                                               valueattrs = (size = 16 pt
                                                             weight = bold
                                                             style = italic
                                                             color = blue)
                                               display = (values);

            textplot x = VAR y = ord text = PARAM / textattrs=(weight = bold
                                                               size = 14 pt)
                                                    fillattrs = (transparency = 0.9)
                                                    url = urlmenu
                                                    rolename = (url = urlmenu)
                                                    tip = (urlmenu)
                                                    position = center;

         endlayout;
      endgraph;
   end;
run;
```

(1)

(2)

Program 10: Template to Create the Selection Menu

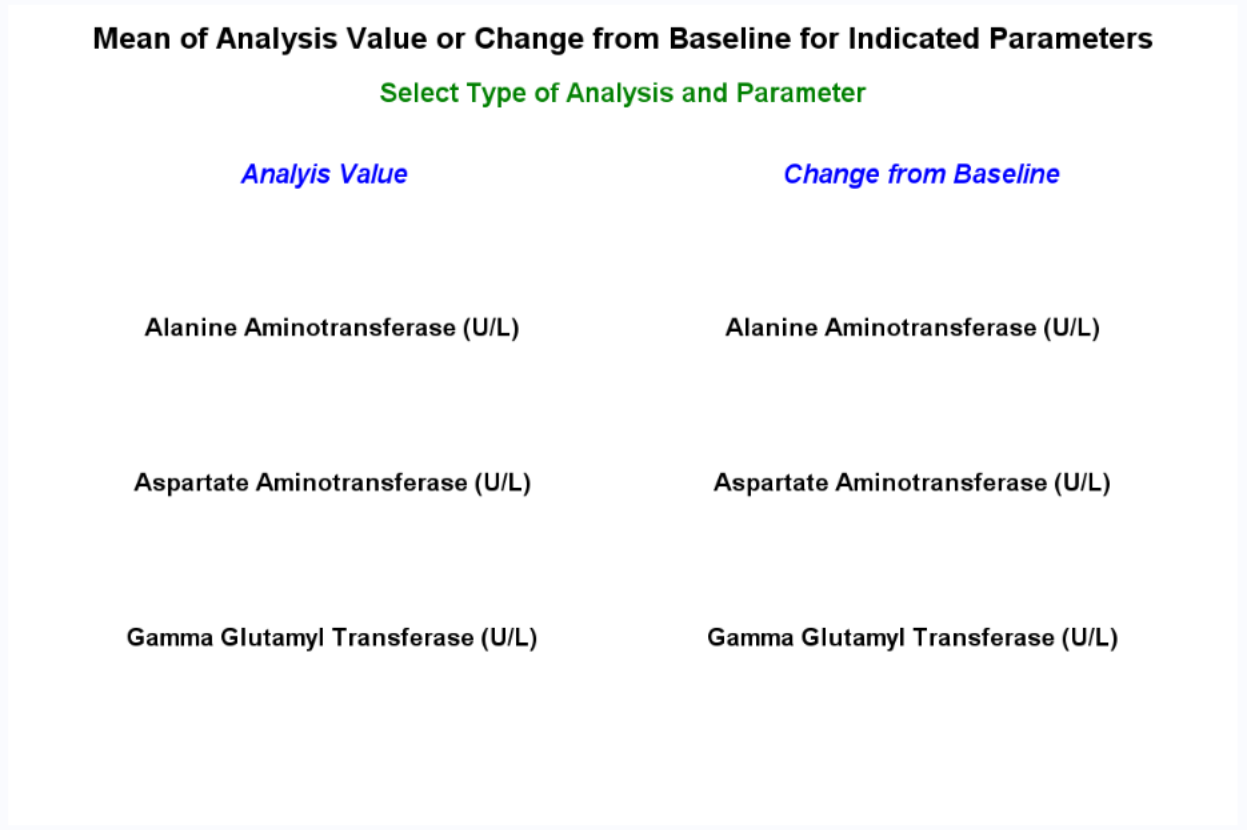
(1) BLOCKPLOT statement is used to create the Data Type headers. These headers have no function other than being displayed.

(2) TEXTPLOT statement allows the indicated text to be displayed at the specified location X and Y on the graph. It is not required that X and Y be numeric values. In our example, X is a character variable that has values of 'AVAL' or 'CHG'. This is so that it will line up under the corresponding Data Type headers (i.e., TEXTPLOT., "Analysis Value" and "Change from Baseline"). Like some of the other plots shown in this example, the TEXPLOT statement has the URL option to allow for linking to another output.

The Selection Menu will be rendered like any other type of graph that uses a template (refer to Program 11). Output 3 illustrates what the Selection Menu looks like once the template is associated with the data. The Selection Menu can also be viewed here:
https://www.krissharris.co.uk/sgf/2019/3261/output/labs/all_labs.html

```
ods graphics / reset border = off imagemap = on imagename = "all_labs"
               drilltarget = "_top" height = 8in width = 12in;

ods html path = outp file = "all_labs.html";
proc sgrender data = menu template = textplot;
run;
ods html close;


ods graphics / reset imagemap = off;
```

Program 11: Associating the Selection Menu Data with the Corresponding Template



Output 3: Selection Menu

TABLE OF CONTENTS WITH GRAPH

In some cases, it may be beneficial to have a table of contents appear alongside your graph. This is especially beneficial when you have more than one type of output that is part of the HTML file that is being generated or if the output you are generating is based on a by group variable. The table of contents will allow you to navigate to the portion of interest within the output. The table of contents and the desired output can also be encompassed into one frame.

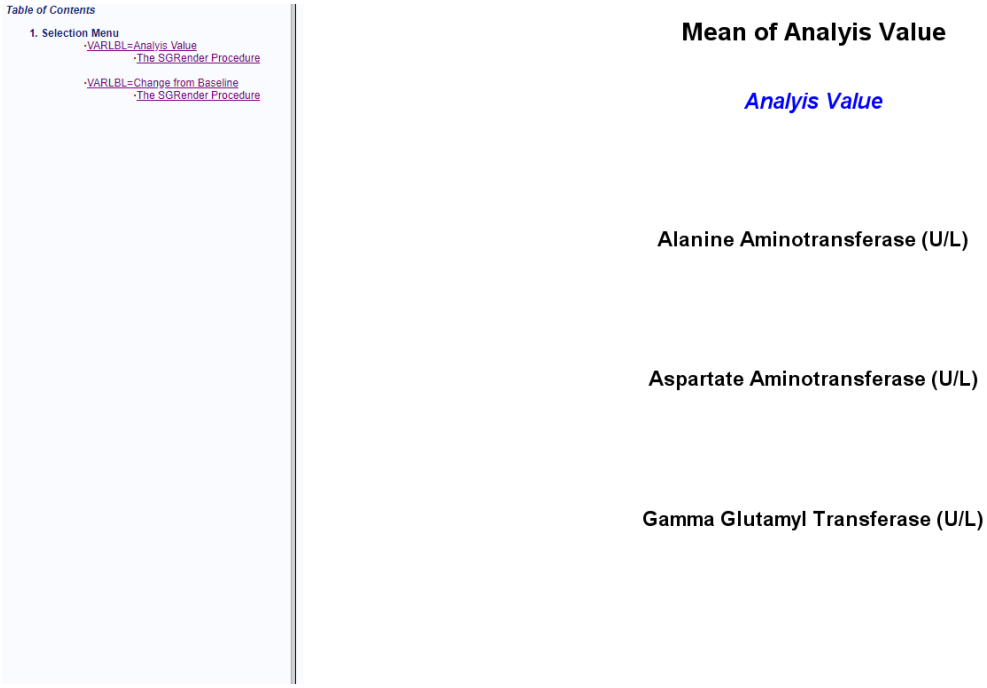
The table of contents along with the graph within the same frame is to be rendered like any other type of graph that uses a template (refer to Program 12). Output 4 illustrates what the entire frame would look like. The frame includes the table of contents along with the Selection Menu looks once it is split into separate pages based on the by group.

```
/* enable image mapping in the HTML output and specify a base image name */
ods graphics / reset border = off imagemap = on imagename = "all_labs" drilltarget =
"_SELF" height = 8in width = 12in;
options nobyline;

/* generate the drill-down graph using ODS HTML */

ods html path = outp contents = "__TOC.html"
                     body = "all_labs.html"
                     frame = "__TOC_LABS.html";

ods proclabel "Selection Menu";

proc sgrender data = menu template = textplot;
   by VARLBL;
run;

/* disable image mapping and open an output destination. */
ods graphics / reset imagemap = off;
```

(1) (2) (3) (4)

Program 12: Creating a Table of Contents and Output within one Frame

1. CONTENTS option on the ODS HTML statement indicates that a table of contents should be created. An entry for each procedure or each by group is generated
2. BODY represents the actual output that is being generated. Multiple pages and or procedures can be contained within the BODY. The CONTENTS helps to navigate to different portions of the output. FILE can also be used to create the main html body of the file.
3. FRAME is used to place the table of contents alongside the main html body.
4. Rather than using the default labels for each procedure a label can be specified. This can be very useful when the same procedure is used to produce different outputs within the main html body. For example, if you are using SGRENDER to produce different graphs, then PROCLABEL can be used to specify a name for each graph.



Output 4: Table of Contents with Selection Menu

DROP-DOWN SELECTION MENU

There is an example on the SAS Website (SAS Institute Inc, 2019), which shows how to create a drill-down graph. The example was modified to allow for the drop-down selection menu. An example of using a drop-down selection can be seen here - https://www.krissharris.co.uk/sgf/2019/3261/output/filter/SOFA_all.html. You will notice that as you select a different region, the plot changes. This is achieved by also using HTML, JavaScript and CSS.

In order to use the selection menu on your graphs, you need to output all possible graphs first, similar to what was done for the AE and Lab examples. For the sofa sales graphs, three graphs are outputted based on three regions: east and west combined, east, and west. Program 13 shows how to output the graphs for each region.

```
%macro genchart(product=, region=, regioncls=, outname=);
  /* Specify the image output filename. */
  ods graphics / reset=all imagemap=on imagename="&product._&outname"; (1)

  /* Generate the graph using ODS HTML. */
  ods _all_ close;
  ods html path=outp file="&product._&outname..html" text="&filterlink." (2)
     headtext='<script src="myScript.js"></script>  <link rel="stylesheet"
     type="text/css" href="myScript.css">';
  proc sgrender data=sales template=drilldown;
    where product = "&product" and &regioncls.;     (3)
    dynamic product="&product" region="&region";
  run;
  ods html close;
%mend genchart;

/* Use the macro to generate the supporting graphs. */
%genchart(product=SOFA, region=%str(West and East), regioncls=%str(region in
('EAST', 'WEST')), outname=all);
%genchart(product=SOFA, region=%str(West), regioncls=%str(region in
('WEST')), outname=west);
%genchart(product=SOFA, region=%str(East), regioncls=%str(region in
('EAST')), outname=east);
```

Program 13: Creating outputs for Each Region

(1) Using the standard ODS GRAPHICS statement, we can output the graph.

(2) Using custom HTML code to help build the selection box and to reference the JavaScript and Cascading Style Sheet (CSS). The FILTERLINK macro variable contains the code, which is used for the selection box, which shown in Program 14.

(3) Within the SGRENDER procedure we can have a where clause that will allow us to select the region for the outputs.

The HTML code to create the selection box is in Program 14 Admittedly, being familiar with HTML code is useful when creating these types of interactive graphics.

```
%let filterlink=<center><div class='dropdown'>
  <button onclick='myFunction()' class='dropbtn'>Select Region</button>
  <div id='myDropdown' class='dropdown-content'>
    <a href='SOFA_all.html'>Regions=All</a>
    <a href='SOFA_west.html'>Regions=West</a>
    <a href='SOFA_east.html'>Regions=East</a>
  </div>
</div>
</center>;
```

Program 14: HTML Needed to Create the Selection Box.

It is not necessary to explain the JavaScript code and CSS that was for the selection box example. The main point of those files is just to be referenced in the HTML code, because they control the appearance of the selection box and the function. The JavaScript code and CSS can be found at the links below and in the Appendix:

- JavaScript - https://www.krissharris.co.uk/sgf/2019/3261/output/filter/myScript.js
- CSS - https://www.krissharris.co.uk/sgf/2019/3261/output/filter/myScript.css

CONCLUSION

SAS can produce interactive graphics when it is combined with HTML. Interactive graphics allow you to drill-down or filter you graphs and allows you to understand the data better.

REFERENCES

CDISC. (2013). *CDISC*. Retrieved from SDTM/ADaM Pilot Project: http://www.cdisc.org/system/files/members/article/application/zip/updated_pilot_submission_package.zip

Harris, K. (2017). Hands-On Graph Template Language (GTL): Part A. *SAS Global Forum.* Denver: SAS Global Forum.

Harris, K., & Watson, R. (2018). Great Time to Learn GTL. *PharmaSUG.* Seattle: PharmaSUG.

SAS Institute Inc. (2019, 02 25). *Create a Drill-Down Graph*. Retrieved from SAS® 9.4 Graph Template Language: User's Guide, Fifth Edition: https://documentation.sas.com/?docsetId=grstatug&docsetTarget=p1vl3qpgkcj6stn1okehgp0nbqvg.htm&docsetVersion=9.4&locale=en

Watson, R. (2019). Great Time to Learn GTL: A Step-by-Step Approach to Creating the Impossible. *SAS Global Forum.* Dallas: SAS Global Forum.

CONTACT INFORMATION

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

Kriss Harris
SAS Specialists Limited
italjet125@yahoo.com
http://www.krissharris.co.uk

Richann Watson
DataRich Consulting
richann.watson@datarichconsulting.com
https://www.datarichconsulting.com

APPENDIX

PROGRAM 2

<Text1>

Click on severity bar for Preferred Term and Treatment details

<Text2>

Click back arrow on navigation bar to return to main bar chart

PROGRAM 3

<Text1>

Click a bar for Preferred Term and Treatment Data

<Text2>

Percentage of Subjects with at Least One AE

JAVASCRIPT

```
function myFunction() {
  document.getElementById("myDropdown").classList.toggle("show");
}

// Close the dropdown menu if the user clicks outside of it
window.onclick = function(event) {
  if (!event.target.matches('.dropbtn')) {
    var dropdowns = document.getElementsByClassName("dropdown-content");
    var i;
    for (i = 0; i < dropdowns.length; i++) {
      var openDropdown = dropdowns[i];
      if (openDropdown.classList.contains('show')) {
        openDropdown.classList.remove('show');
      }
    }
  }
}
```

CSS

```
/* Dropdown Button */
.dropbtn {
  background-color: #3498DB;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}

/* Dropdown button on hover & focus */
.dropbtn:hover, .dropbtn:focus {
  background-color: #2980B9;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
  position: relative;
  display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

/* Links inside the dropdown */
.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #ddd}

/* Show the dropdown menu (use JS to add this class to the .dropdown-content
container when the user clicks on the dropdown button) */
.show {display:block;}
```