Paper DV11

Interactive Data Visualization: A Tool for Common Clinical Study Figures and SAP Figure Shells

Ji Qi, BioPier Inc., Burlington, USA
Qisen Luan, BioPier Inc., Burlington, USA

ABSTRACT

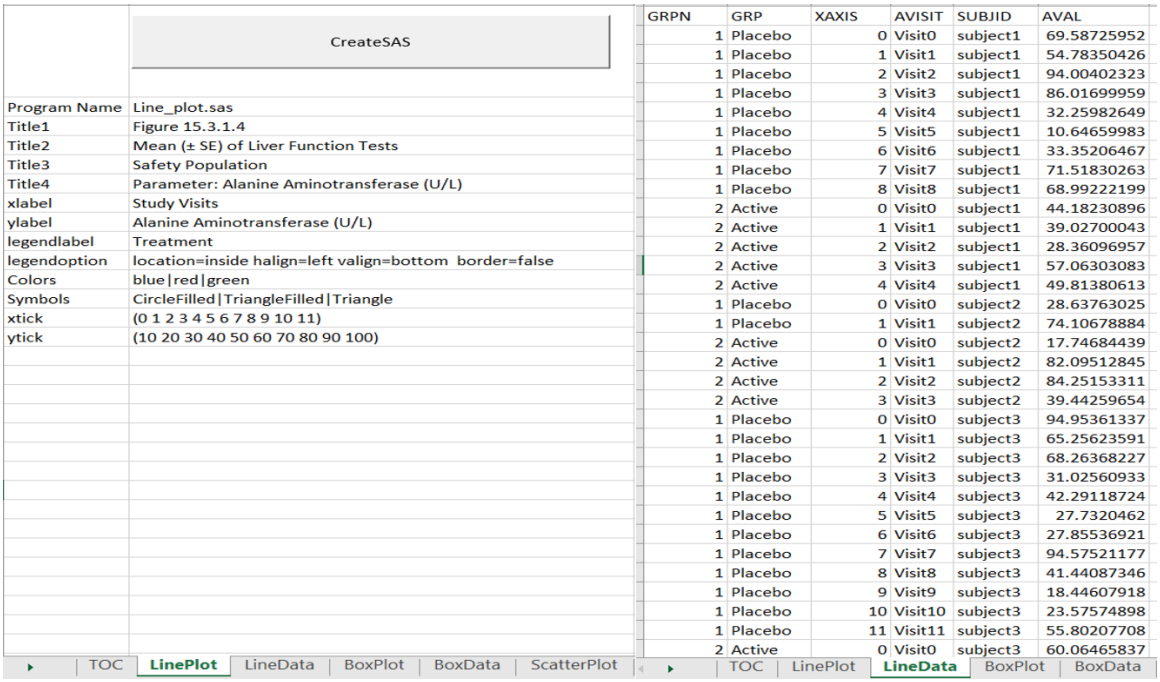
Figures are essential and critical elements in clinical trial data analysis and reporting. We propose a tool for the interactive generation of mock-up biostatistical figures using Excel and SAS. An excel file holds sample preset data and figure attributes, such as title, axis labels, tick definitions, groups, group attributes, etc. Actual values of data or attributes could be modified following the samples. The SAS program code for each type of figure can be generated automatically from the Excel file utilizing Visual Basic for Application (VBA) scripts, thus producing the sample outputs. The figures produced by this tool could help statisticians choose the right type of figures for visualizing different data, tailor figure details, generate mock-up shells, and quickly generate additional figures for reporting/presentation purposes if needed. The SAS code generated in the designing stage could be shared with programmers to produce the figures for study reports later.

INTRODUCTION

In clinical study data reporting and analysis, figures are unique and essential in that they help visualize critical changes within a treatment group or comparisons among treatment groups easily. But usually, it is not as easy for statisticians to produce figure shells for the Statistical Analysis Plan (SAP). They first need to determine the types of figures to utilize for the best presentation of different data types and the details within the figures. Then the next challenge is to convey that information clearly in the figure shell. To accomplish the SAP figure shell task, a tool that can generate different types of figures quickly with easily adjustable attributes would be of great help. This paper will demonstrate a tool we developed for various common figure types in clinical trials.

USER INTERFACE AND WORKFLOW

As shown in Figure 1, this tool is based on Microsoft® Excel. Each type of figure utilizes two tabs: one for figure attributes (panel a) and another for data (panel b). Both are pre-filled with example data, and users can adjust the values according to their needs.



	CreateSAS
Program Name	Line_plot.sas
Title1	Figure 15.3.1.4
Title2	Mean (± SE) of Liver Function Tests
Title3	Safety Population
Title4	Parameter: Alanine Aminotransferase (U/L)
xlabel	Study Visits
ylabel	Alanine Aminotransferase (U/L)
legendlabel	Treatment
legendoption	location=inside halign=left valign=bottom border=false
Colors	blue|red|green
Symbols	CircleFilled|TriangleFilled|Triangle
xtick	(0 1 2 3 4 5 6 7 8 9 10 11)
ytick	(10 20 30 40 50 60 70 80 90 100)

TOC | LinePlot | LineData | BoxPlot | BoxData | ScatterPlot

GRPN	GRP	XAXIS	AVISIT	SUBJID	AVAL
1	Placebo	0	Visit0	subject1	69.58725952
1	Placebo	1	Visit1	subject1	54.78350426
1	Placebo	2	Visit2	subject1	94.00402323
1	Placebo	3	Visit3	subject1	86.01699959
1	Placebo	4	Visit4	subject1	32.25982649
1	Placebo	5	Visit5	subject1	10.64659983
1	Placebo	6	Visit6	subject1	33.35206467
1	Placebo	7	Visit7	subject1	71.51830263
1	Placebo	8	Visit8	subject1	68.99222199
2	Active	0	Visit0	subject1	44.18230896
2	Active	1	Visit1	subject1	39.02700043
2	Active	2	Visit2	subject1	28.36096957
2	Active	3	Visit3	subject1	57.06303083
2	Active	4	Visit4	subject1	49.81380613
1	Placebo	0	Visit0	subject2	28.63763025
1	Placebo	1	Visit1	subject2	74.10678884
2	Active	0	Visit0	subject2	17.74684439
2	Active	1	Visit1	subject2	82.09512845
2	Active	2	Visit2	subject2	84.25153311
2	Active	3	Visit3	subject2	39.44259654
1	Placebo	0	Visit0	subject3	94.95361337
1	Placebo	1	Visit1	subject3	65.25623591
1	Placebo	2	Visit2	subject3	68.26368227
1	Placebo	3	Visit3	subject3	31.02560933
1	Placebo	4	Visit4	subject3	42.29118724
1	Placebo	5	Visit5	subject3	27.7320462
1	Placebo	6	Visit6	subject3	27.85536921
1	Placebo	7	Visit7	subject3	94.57521177
1	Placebo	8	Visit8	subject3	41.44087346
1	Placebo	9	Visit9	subject3	18.44607918
1	Placebo	10	Visit10	subject3	23.57574898
1	Placebo	11	Visit11	subject3	55.80207708
2	Active	0	Visit0	subject3	60.06465837

TOC | LinePlot | LineData | BoxPlot | BoxData

a. Attributes Tab **b. Data Tab**

Figure 1. User Interface

After setting the input, a simple click on the 'Create SAS' button on the top of the attributes tab will generate and store the SAS program for the corresponding figure at the same location as the tool under the program name specified in the excel. In our example shown In Figure 1, the output program name is Line_plot.sas.

The code generated contains three parts: a) attributes setup; b) data read-in (only partially shown for simplicity); c) post-processing and graph generation.

a.
```
%let title1 = Figure 15.3.1.4;
%let title2 = Mean (± SE) of Liver Function Tests;
%let title3 = Safety Population;
%let title4 = Parameter: Alanine Aminotransferase (U/L);
%let xlabel = Study Visits;
%let ylabel = Alanine Aminotransferase (U/L);
%let legendlabel = Treatment;
%let legendoption = location=inside halign=left valign=bottom  border=false;
%let colors = blue|red|green;
%let symbols = CircleFilled|TriangleFilled|Triangle;
%let xtick = (0 1 2 3 4 5 6 7 8 9 10 11);
%let ytick = (10 20 30 40 50 60 70 80 90 100);
```

b.
```
data final;
    length GRP $200.;
    length AVISIT $200.;
    length SUBJID $200.;
    GRPN=1; GRP="Placebo"; XAXIS=0; AVISIT="Visit0"; SUBJID="subject1"; AVAL=69.5872595231049;  output;
    GRPN=1; GRP="Placebo"; XAXIS=1; AVISIT="Visit1"; SUBJID="subject1"; AVAL=54.7835042630322;  output;
    GRPN=1; GRP="Placebo"; XAXIS=2; AVISIT="Visit2"; SUBJID="subject1"; AVAL=94.0040232324973;  output;
    GRPN=1; GRP="Placebo"; XAXIS=3; AVISIT="Visit3"; SUBJID="subject1"; AVAL=86.0169995897449;  output;
```

c.
```
proc sort data = final; by grpn grp xaxis avisit; run;
proc univariate data = final noprint;
    by grpn grp xaxis avisit;
    var aval;
    output out=final2 mean=yaxis stdmean=interval;
run;
data final3; set final2; yaxisl=yaxis-interval; yaxish=yaxis+interval; run;
proc sort data = final2(where=(grpn > .)) out = grps nodupkey; by grpn; run;
data _null_;
   set grps end=eof;
   call symput ('grps'||compress(put(_n_,best12.)), trim(put(grp,$100.)));
   if eof then call symput('tot_grp', compress(put(_n_, best12.)));
run;
%macro mc1;
proc template;
    define statgraph LinePlot;
        dynamic TITLE1 TITLE2 TITLE3 TITLE4 TITLE5;
        begingraph / border=false designwidth=9in designheight=4.8in;
            discreteattrmap name="grp" / ignorecase=true;
            %do i = 1 %to &tot_grp.;
            value "&&grps&i.." / markerattrs=GraphData&i(color=%scan(&colors, &i,|) symbol=%scan(&symbols, &i,|) size=7)
            lineattrs=GraphData&i(color=%scan(&colors, &i,|)) fillattrs=(color=%scan(&colors, &i,|)) textattrs=(color=%scan(&colors, &i,|));
            %end;
            enddiscreteattrmap;
            discreteattrvar attrvar=grp var=grp attrmap="grp";
            entrytitle TITLE1; entrytitle TITLE2; entrytitle TITLE3; entrytitle TITLE4; entrytitle TITLE5;
              layout overlay /
                xaxisopts=(LABELATTRS=(SIZE=8pt) label="&xlabel." type=linear linearopts=(tickvaluelist=&xtick. tickvaluefitpolicy=rotate ))
                yaxisopts=(LABELATTRS=(SIZE=8pt) label="&ylabel." linearopts=(tickvaluelist=&ytick. ));
                seriesplot  x=xaxis y=yaxis / group=grp  name='a' includemissinggroup=false;
                scatterplot x=xaxis y=yaxis / group=grp  name='b' YerrorLower=yaxisl YerrorUpper=yaxish ;
                MERGEDLEGEND 'a' 'b' / title="&legendlabel." down=2 &legendoption.;
            endlayout;
        endgraph;
    end;
run;
%mend mc1; %mc1;
proc sgrender data=final3 template=LinePlot;
    dynamic TITLE1="&title1" TITLE2="&title2" TITLE3="&title3" TITLE4="&title4";
run;
```

Figure 2. Excerpts of Output SAS Codes

The last step is simply submitting the SAS codes to generate the figure. In our example, the output line plot is shown in Figure 3.

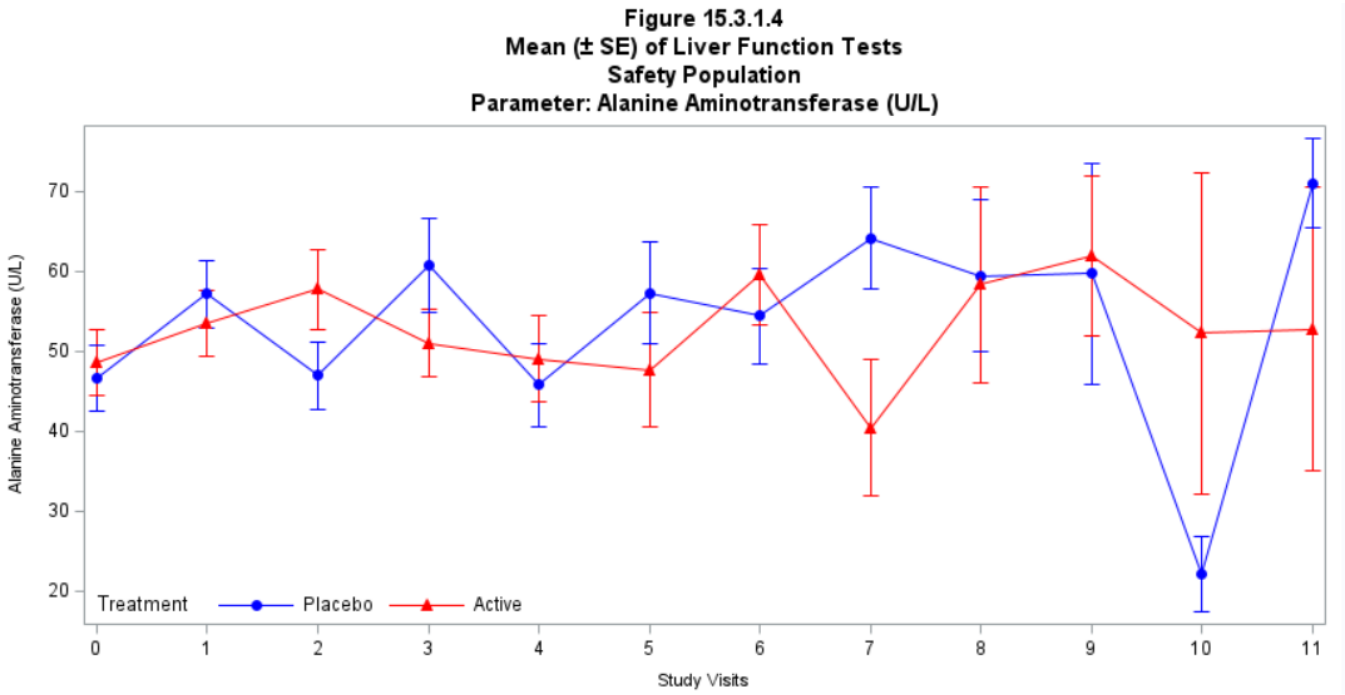


Figure 3. Line Plot Example

The straightforward process for the creation of sample figures by this tool is summarized in the flow chart below:

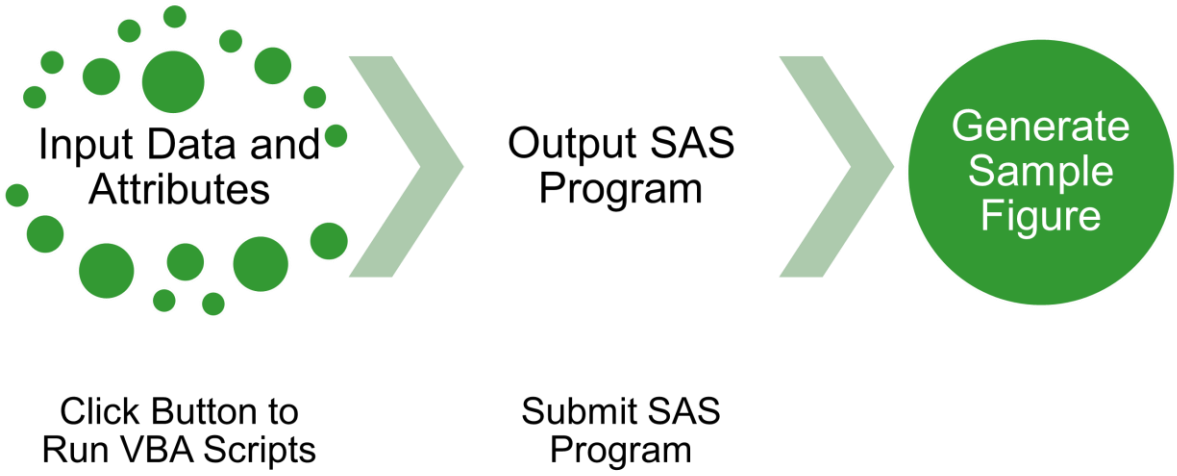


Figure 4. Sample Figure Creation Process

BEHIND THE SCENES: USE VBA TO GENERATE SAS PROGRAM

The one-click generation of the figure SAS program is realized by VBA scripts. We will briefly discuss the mechanisms below.

INITIATE VBA MACRO AND ADD CREATE SAS BUTTON

1) In the excel file that holds the figure attributes and data, press ALT+F11 to launch the VBA editor. Double-click on 'This Workbook' on the left and enter scripts 'Sub Demom() End Sub' to initiate a macro named ThisWorkbook.Demom for this whole excel workbook.

2) To add a button in the attributes tab, go to Files-> Options-> Customize ribbon to add the 'Developer' tab to the display ribbon. Then insert a button by going to the developer tab -> insert -> select a Form Controls button. Place it in the same column as the attribute values (in our case, cell B1).

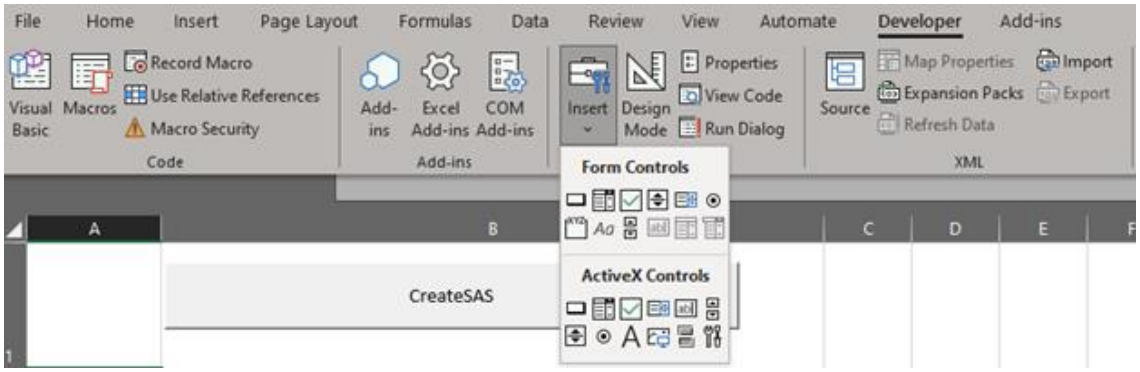


Figure 5. Insert the 'Create SAS' Button

3) A dialog window will prompt the user to assign a macro to the button. Or the macro 'ThisWorkbook.Demom' could be assigned to the button from the right-click menu. The text on the button could also be edited from the right-click menu.

READ THE ATTRIBUTES TAB INFORMATION WITH VBA

1) Use the following code to find the column of the attributes:

```
With ActiveSheet.Buttons(Application.Caller).TopLeftCell
ProgCol = .Column
End With
```

Then find each row of desired information with the following code as an example:

```
Rowa = Cells.Find(What:="Program Name", After:=ActiveSheet.Cells(1, 1), LookAt:=xlPart,
LookIn:=xlValues, SearchOrder:=xlByRows, SearchDirection:=xlPrevious, MatchCase:=False).Row
```

With the current values of ProgCol and Rowa, the 'Program Name' information can be read by VBA at Cells(Rowa, ProgCol).

2) Get the path of the current workbook and prepare the output for SAS programs.

```
Set pgm = Cells(Rowa, ProgCol)
path = ThisWorkbook.path & "\" & pgm
Open path For Output As FileNum
```

Then use the following code to output the SAS program. The example below only output the first row in Figure 2. Notice to use 'Chr(34)' to replace all double quotation marks going into SAS codes when creating the VBA code.

```
Print #FileNum, "%let title1 = " & title1 & ";"
```

READ THE DATA TAB INFORMATION WITH VBA

1) 'Activate' data tab to switch the active sheet from the attributes tab to the data tab.

2) Find the total number of columns and rows in the data region.

3) For each column, detect the variable data type as string or numeric. If the variable type is string (VarType returns 8), set the variable length to $200 in the output SAS code. Otherwise, a variable is set as numeric.

4) Read through each row of data. Any numeric variable with a missing entry is set to the value "."; any string variable with a missing entry is set to the value """, otherwise we output the entry value as is.

Sample VBA scripts for this part can be found in Appendix 1.

FEATURE HIGHLIGHT: INTERACTIVE VISUALIZATION

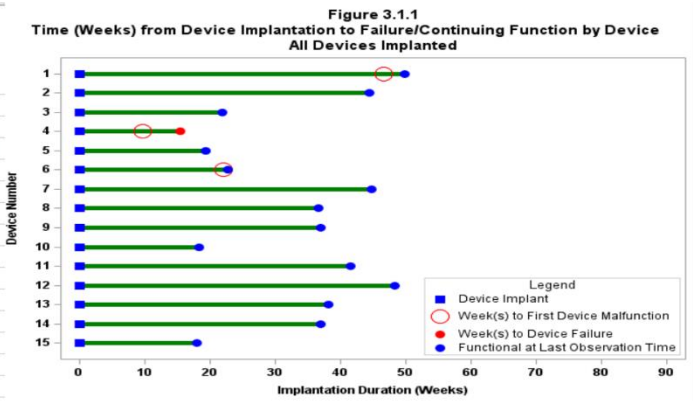
Now let's take a closer look at the key feature of our tool: interactive adjustments of the figure output. Figure 6 compares the swimmer plot outputs before and after attribute adjustments. In this example, all attributes, including titles, axis labels, legend labels, legend options, axis tick values, marker shapes, marker sizes, marker colors, line colors, and line thickness, are modified to illustrate the flexibilities of the tool. Although there is no change to the data, the resulting presentations are very different. Comparing panels b and d, we would like to point out some key differences that impact the completeness and clarity of the output. Firstly, the location of the legend. Placing the long legend inside the graph area saves space vertically, so each page could fit more rows of data or show more clearly. But this also requires adjustments in the x-axis values. The device implantation duration values in the data are only up to about 50 weeks. To avoid the overlap of the legend with the data line,

the maximum x-axis value is set to 90. The x-axis tick interval is decided by the range. For a range of 90, an interval of 10 is clear; whereas for a range of 50, an interval of 5 works better. Secondly, the marker symbols and colors. For device number 6, the time points for the first malfunction and the last observation time are close, so the two symbols overlap. It is clear from the comparison that the symbol pair in panel b is easier to see than that in panel d.

The flexibility of our tool is beyond quick visual checks on varying attributes. Changes in the input data could also be well accommodated. Looking back at Figures 1 and 3, we can see that the default values for 'colors' and 'symbols' are set up for three groups, but the data and output only have two groups. Still, the tool runs smoothly. On the other hand, if more groups are needed in the data, they will be properly presented by simply assigning new symbols and colors (see Figure 7 below).

	CreateSAS
Program Name	Swimmer_plot.sas
Title1	Figure 3.1.1
Title2	Time (Weeks) from Device Implantation to Failure/Continuing Function by Device
Title3	All Devices Implanted
xlabel	Implantation Duration (Weeks)
ylabel	Device Number
legendlabel	Legend
legendoption	border=on valign=bottom halign=right location=inside
xtickvalue	xmin=0 xmax=90 xtick=10
x1attr	SYMBOL=squarefilled size=8pt color=blue
x2attr	SYMBOL=circle size=13pt color=red
x3attr	SYMBOL=circlefilled size=8pt color=red
x4attr	SYMBOL=circlefilled size=8pt color=blue
lineattr	color=green thickness=4pt

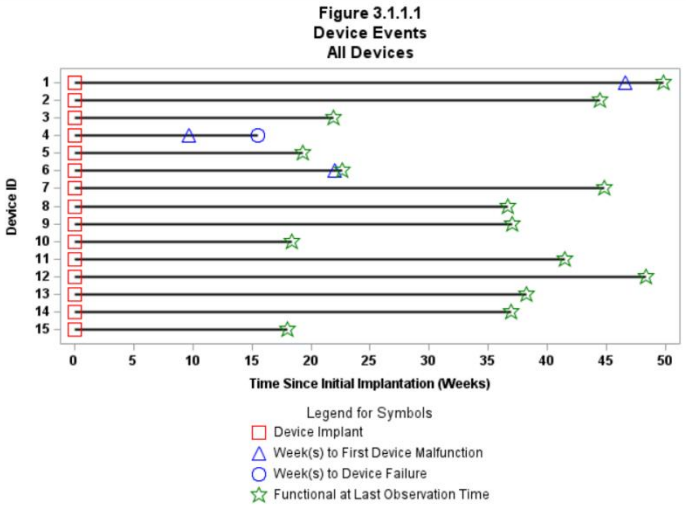
a. Default Attributes



b. Swimmer Plot Output with Default Attributes

	CreateSAS
Program Name	Swimmer_plot2.sas
Title1	Figure 3.1.1.1
Title2	Device Events
Title3	All Devices
xlabel	Time Since Initial Implantation (Weeks)
ylabel	Device ID
legendlabel	Legend for Symbols
legendoption	border=off valign=bottom halign=center location=outside
xtickvalue	xmin=0 xmax=50 xtick=5
x1attr	SYMBOL=square size=10pt color=red
x2attr	SYMBOL=triangle size=10pt color=blue
x3attr	SYMBOL=circle size=10pt color=blue
x4attr	SYMBOL=star size=10pt color=green
lineattr	color=black thickness=2pt

c. Adjusted Attributes



d. Swimmer Plot Output with Adjusted Attributes

Figure 6 Interactive Attributes Adjustment Example: Swimmer Plot

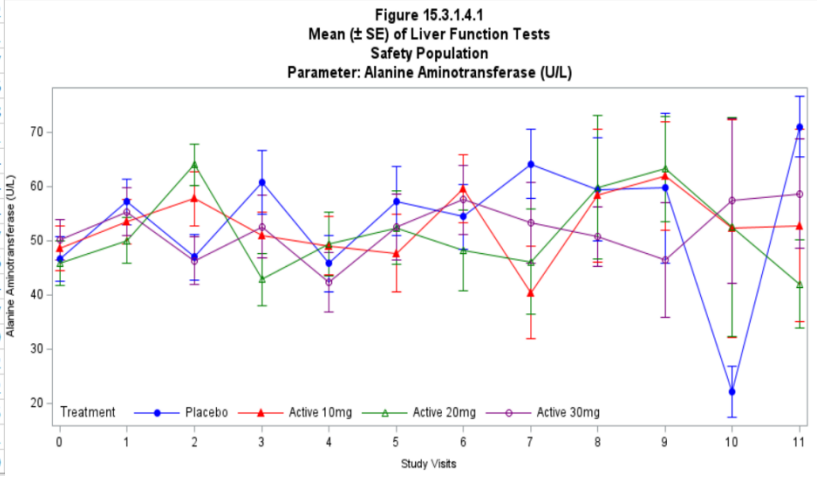
1	GRPN	GRP	XAXIS	AVISIT	SUBJID	AVAL
432	4	Active 30m	0	Visit0	subject1	59.58725952
433	4	Active 30m	1	Visit1	subject1	44.78350426
434	4	Active 30m	2	Visit2	subject1	84.00402323
435	4	Active 30m	3	Visit3	subject1	76.01699959
436	4	Active 30m	4	Visit4	subject1	22.25982649
437	4	Active 30m	5	Visit5	subject1	0.646599834
438	4	Active 30m	6	Visit6	subject1	23.35206467
439	4	Active 30m	7	Visit7	subject1	61.51830263
440	4	Active 30m	8	Visit8	subject1	58.99222199
441	3	Active 20m	0	Visit0	subject1	21.63763025
442	3	Active 20m	1	Visit1	subject1	67.10678884
443	3	Active 20m	2	Visit2	subject1	87.95361337
444	3	Active 20m	3	Visit3	subject1	58.25623591
445	3	Active 20m	4	Visit4	subject1	61.26368227
446	4	Active 30m	0	Visit0	subject2	20.7320462
447	4	Active 30m	1	Visit1	subject2	20.85536921
448	3	Active 20m	0	Visit0	subject2	87.57521177
449	3	Active 20m	1	Visit1	subject2	34.44087346
450	3	Active 20m	2	Visit2	subject2	75.74615278
451	3	Active 20m	3	Visit3	subject2	3.993787161
452	4	Active 30m	0	Visit0	subject3	57.27961864
453	4	Active 30m	1	Visit1	subject3	43.57632714
454	4	Active 30m	2	Visit2	subject3	31.72614014
455	4	Active 30m	3	Visit3	subject3	22.24322137
456	4	Active 30m	4	Visit4	subject3	74.37215526
457	4	Active 30m	5	Visit5	subject3	10.59711944
458	4	Active 30m	6	Visit6	subject3	9.673654677
459	4	Active 30m	7	Visit7	subject3	57.92462329
460	4	Active 30m	8	Visit8	subject3	61.28494942
461	4	Active 30m	9	Visit9	subject3	77.14251102
462	4	Active 30m	10	Visit10	subject3	79.76455595
463	4	Active 30m	11	Visit11	subject3	66.71405171
464	3	Active 20m	0	Visit0	subject3	11.03947309

a. Updated Line Plot Data

CreateSAS

Program Name	Line_plot2.sas
Title1	Figure 15.3.1.4.1
Title2	Mean (± SE) of Liver Function Tests
Title3	Safety Population
Title4	Parameter: Alanine Aminotransferase (U/L)
xlabel	Study Visits
ylabel	Alanine Aminotransferase (U/L)
legendlabel	Treatment
legendoption	location=inside halign=left valign=bottom border=false
Colors	blue|red|green|purple
Symbols	CircleFilled|TriangleFilled|Triangle|Circle
xtick	(0 1 2 3 4 5 6 7 8 9 10 11)
ytick	(10 20 30 40 50 60 70 80 90 100)

b. Updated Line Plot Attributes



c. Updated Line Plot Output

Figure 7 Interactive Data/Attributes Adjustment Example: Line Plot

DEMONSTRATION: SAMPLE FIGURE LIBRARY

In the first tab of our tool, the Table of Contents is included, which shows the eight types of figures currently in our library (Figure 8). The selection of figure types and sample figures for those not shown already are detailed below.

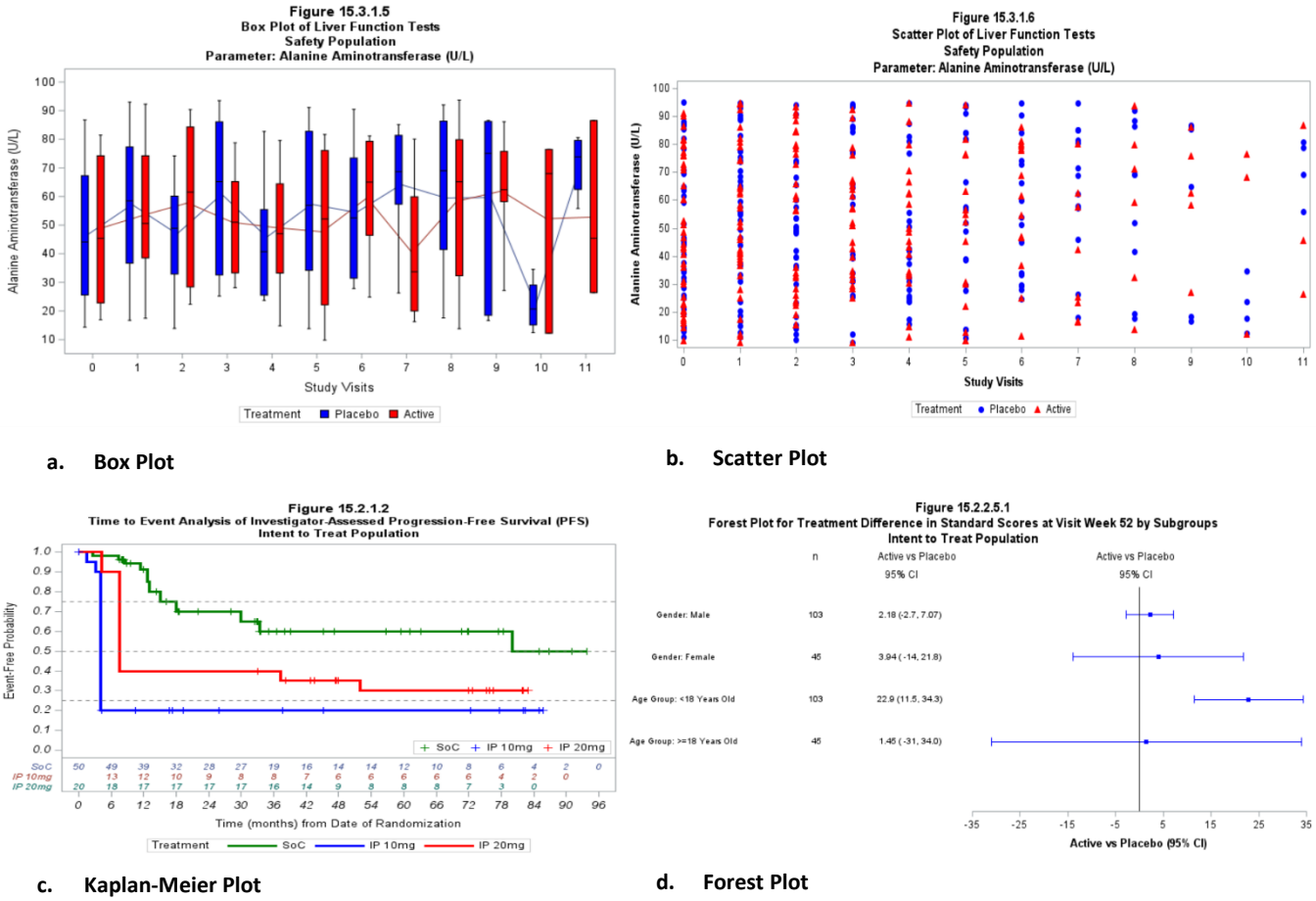
Table of Contents

Figure 8 Table of Contents of Current Figure Types

By surveying all 77 reports of randomized controlled trials in five general medical journals from November 2006 to January 2007, Pocock SJ et al. have found that the types of figures mostly used in clinical trials reports are flow diagrams, Kaplan-Meier plots, forest plots, and repeated measures plots [1]. The flow diagram is an integral part of the CONSORT guidelines, adopted by most major journals. Its aim is to display the flow of participants through each stage, specifically for each randomized group reporting the numbers randomly assigned, receiving intended treatment, completing the study protocol, and analyzed for the primary outcome [1]. Since it is usually not part of the figure output defined by the clinical study SAP, it is not

included in the tool. Kaplan-Meier plots are commonly used to present time-to-event results. Forest plots are typically used for displaying subgroup analysis results. Sample figures for these two types are in Figure 9 lower panels.

Repeated measures plots display observed measurement values at multiple time points, typically the baseline and follow-up time points, or changes from baseline at different time points. In the paper from Pocock SJ et al. [1], the repeated measure plots referred to line plots, but other formats could be used too. Box plots at different time points, sometimes also connected with a line through the mean values, could visualize the same information about the trend of changes, with additional information for data distribution and outlier at each time point. Similarly, box plots at each time point could be replaced by scatterplots, which show the complete information on data distribution. The sample line plot has been shown above, and the sample box plot and scatter plot are in Figure 9 upper panels. Besides visualizing repeated measures data, scatter plots can also visualize X-Y relationships, for example, the relationship between estimated changes and actual changes.



a. Box Plot

b. Scatter Plot

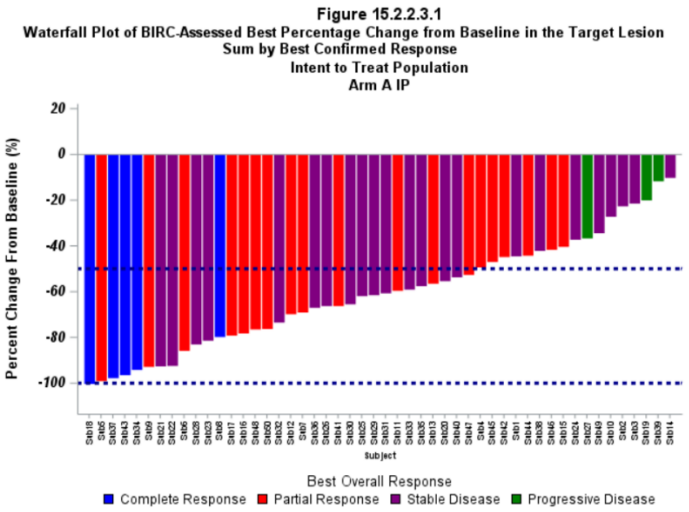
c. Kaplan-Meier Plot

d. Forest Plot

Figure 9 Box Plot, Scatter Plot, Kaplan-Meier Plot and Forest Plot Samples From the Tool

Multiple papers focusing on the graphs in oncology trials or cancer research data reporting mentioned that waterfall plots and swimmer plots are common graphs to visualize each patient's response to tumor interventions [2,3,4]. Moreover, swimmer plots can also display other individual patient/device data over time, such as adverse events, device surgeries and functional status, etc. The sample swimmer plot is in Figure 6 above, and the sample waterfall plot is in Figure 10.

Heatmaps are a data visualization tool that uses color to represent data values in a two-dimensional image. They are powerful in mapping cancer genotypes and phenotypes [3]. They can also be used to represent adverse events by treatment, metabolic abnormalities, the infection rate in different populations, etc. [5]



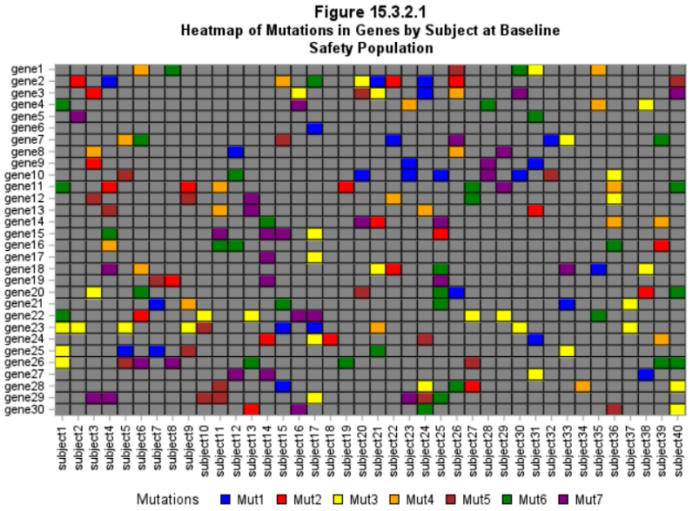


Figure 10 Waterfall Plot and Heatmap Samples From the Tool

DISCUSSION

Now we know the workflow and features of this tool, let's discuss more how it helps with clinical trials, including the generation of SAP figure shells and actual data analysis.

CHOOSE THE RIGHT FIGURE TYPE

For some study data, the decision on figure types is straightforward; for example, the Kaplan-Meier plot for time-to-event data, as discussed earlier in the paper. But for other data, it may require more effort for such decisions. In the case of repeated measures data, line plots, box plots, and scatter plots could be potential options. The choice in each study depends on the sample size, the number of treatment groups and time points, the endpoint measure, etc., to make the output figure as informative as possible but be clear and succinct at the same time. With the tool, if a dummy dataset tailored to the specific study is input into the data tab, sample outputs in different figure types could be quickly generated and facilitate the statistician to land on a confident decision.

DECIDE ON FIGURE DETAILS AND CONVEY THE INFORMATION IN THE SHELL

It is better to display the desired figure attributes as expected in figure shells, although the sample graphs are not reflecting actual data. Otherwise, plenty of explanations will be required in the programming notes or other communications. From our experience, we have received shells with lengthy text descriptions of the figure only, a minimal figure with just the axes paired with text descriptions, or a sample figure borrowed from similar studies with text for the desired changes. We also took the approach of borrowing sample figures from other studies for the task of figure shell generation. With those shells, the client often asked our team to generate a draft figure and then commented to make changes to the attributes. Now that we have this new tool, the details can be tried out during the interactive process to achieve a satisfactory presentation, as discussed above in the feature highlight section, and illustrated in Figure 6. Then a sample graph according to study-specific expectations could be generated and put into the shell. This could greatly simplify the back-and-forth process for the outputs review and update and shorten the timeline.

REUSE SAS CODE TO FACILITATE ANALYSIS

Figure 2 above demonstrates the SAS program output from the data and attributes settings in Figure 1. When it is time to produce the figure output, the programmer can replace the dummy data (panel b) with actual study data and reuse the rest of the codes. Slight modifications might be necessary to account for the differences in the data, for example, data ranges, or repeat the figure for multiple parameters. But overall, the saving in effort and time is significant.

CONCLUSION

In summary, in this paper, we demonstrated a tool to interactively visualize data and adjust attributes as needed in a user-friendly interface. Besides the types of figures showcased in this paper, this tool is extensible to other figures following the same idea. For statisticians, this tool will help decide the type and details for figures to use, and the sample figure output could be used to generate mock-up shells with exact expectations. For programmers, the output SAS program code from the designing stage could be a time-saver for output generation. Also, this tool could help quickly generate additional figures for reporting/presentation purposes if needed.

REFERENCES

[1] Pocock SJ, Travison TG, Wruck LM. *Figures in clinical trial reports: current practice & scope for improvement.* Trials. 2007 Nov 19;8:36. doi: 10.1186/1745-6215-8-36.rials. 2007 Nov 19;8:36.

[2] Murali Kanakenahalli, Avani Kaja. *Graphical Presentation of Clinical Data in Oncology Trials.* PharmaSUG 2015 - Paper DV10

[3] Chia PL, Gedye C, Boutros PC, Wheatley-Price P, John T. *Current and Evolving Methods to Visualize Biological Data in Cancer Research.* J Natl Cancer Inst. 2016 May 31;108(8):djw031. doi: 10.1093/jnci/djw031. Print 2016 Aug.PMID: 27245079

[4] Liqin Wang. *Graphical Presentation of Clinical Data in Oncology Studies Using R.* PharmaSUG China 2019 - Paper DV-067

[5] Pavani Potuganti, and Sree Nalluri. *Graphical Representation of Clinical Data using Heat Map Graphs.* PharmaSUG 2021 - Paper DV-078

ACKNOWLEDGMENTS

We would like to thank Lixin Gao (CEO of Biopier Inc.) for conceptualizing the tool and providing feedbacks on the optimization.

CONTACT INFORMATION

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

Ji Qi, Ph.D.
BioPier Inc.
80 Blanchard Road #104
Burlington, MA 01803
Work Phone: 781-354-1924
Email:jiqi@biopier.com
Web: http://biopier.com/

Qisen Luan
BioPier Inc.
80 Blanchard Road #104
Burlington, MA 01803
Work Phone: 781-354-1924
Email:qluan@biopier.com
Web: http://biopier.com/

APPENDIX 1

```
Sheets("LineData").Activate
datarow = Cells.Find(What:="*", After:=ActiveSheet.Cells(1, 1), LookAt:=xlPart,
LookIn:=xlValues, SearchOrder:=xlByRows, SearchDirection:=xlPrevious, MatchCase:=False).Row
datacol = Cells.Find(What:="*", After:=ActiveSheet.Cells(1, 1), LookAt:=xlPart,
LookIn:=xlValues, SearchOrder:=xlByColumns, SearchDirection:=xlPrevious,
MatchCase:=False).Column
ReDim Header(0 To datacol)
Set dataheader = ActiveSheet.Range(Cells(1, 1), Cells(1, datacol))
Set datarange = ActiveSheet.Range(Cells(2, 1), Cells(datarow, datacol))
For l = 1 To datarange.Columns.Count
    If VarType(datarange.Cells(1, l)) = 8 Then
        Print #FileNum, "    length " & dataheader.Cells(1, l) & " $200.;"
        Header(l) = 8
    ElseIf VarType(datarange.Cells(1, l)) = 0 Then
        For lc = 1 To datarange.Rows.Count
            If VarType(datarange.Cells(lc, l)) = 8 Then
                Print #FileNum, "    length " & dataheader.Cells(1, l) & " $200.;"
                Header(l) = 8
                Exit For
            ElseIf VarType(datarange.Cells(lc, l)) = 5 Then
                Header(l) = 5
                Exit For
            End If
        Next lc
    Else
        Header(l) = 5
    End If
Next l
For j = 1 To datarange.Rows.Count
    For k = 1 To datarange.Columns.Count
        If VarType(datarange.Cells(j, k)) = 0 And Header(k) = 5 Then
            LineText = IIf(k = 1, "    ", LineText) & dataheader.Cells(1, k) & "=.; "
        ElseIf VarType(datarange.Cells(j, k)) = 0 And Header(k) = 8 Then
            LineText = IIf(k = 1, "    ", LineText) & dataheader.Cells(1, k) & "=" & Chr(34)
& Chr(34) & "; "
        Else
            LineText = IIf(k = 1, "    ", LineText) & dataheader.Cells(1, k) & "=" &
IIf(Header(k) <> 8, "", Chr(34)) & datarange.Cells(j, k) & IIf(Header(k) <> 8, "", Chr(34)) &
"; "
        End If
    Next k
    Print #FileNum, LineText & " output;"
Next j
```