Paper DV05

Expanding Visual Clarity: Multi-Page Figure Partitioning and Adaptive Zero-Value Depiction using PROC SGPLOT

Arman Melkonyan, STATECS LLC, Yerevan, Armenia
Yeghishe Grigoryan, STATECS LLC, Yerevan, Armenia

ABSTRACT

In recent years, the demand for visually appealing and informative graphical representations of data has increased significantly. When dealing with large datasets containing numerous subgroups and visits, it becomes essential to present the information in a clear and accessible manner. This paper introduces a novel approach to enhance visual clarity by partitioning complex figures into multiple pages and adaptively depicting zero-values using PROC SGPLOT in SAS. The proposed method focuses on partitioning figures across multiple pages based on subgroups, such as parameters, and adaptively depicting zero-value on the Y-axis, ensuring a more accurate representation of data. The method dynamically calculates the optimal distribution of visits per page, minimizing the occurrence of underutilized pages that would otherwise display a sparse set of data points. Additionally, the adaptive zero-value depiction addresses the challenges posed by data sets containing both positive and negative values, thus ensuring a more comprehensive representation of data. The implementation of this approach significantly improves the visual clarity of figures and has the potential to enhance data-driven decision-making across various industries and applications.

INTRODUCTION

In the fast-paced world of clinical research, data visualization plays a crucial role in effectively communicating complex information and facilitating informed decision-making. Researchers, programmers, and data analysts frequently encounter challenges when visualizing data, particularly when dealing with high-visit figures and displaying dynamically Zero-Value on Y-Axis. This article presents a groundbreaking SAS macro program that transforms the generation of line plot figures, introducing innovative methods for enhanced data visualization. This flexible macro program seamlessly integrates with the BDS (Basic Data Structure) ADaM (Analysis Data Model) datasets, incorporating essential components such as subgroups, visits and analysis variables. The article unravels in two key stages: Navigating High-Visit Figures with Smart Pagination and Consistent Zero-Value Display and Customizable Y-Axis Division.

STAGE 1: NAVIGATING HIGH-VISIT FIGURES WITH SMART PAGINATION

The initial stage delves into an ingenious method for managing high-visit figures through dynamic partitioning of data across multiple pages. By leveraging the capabilities of SAS PROC SGPLOT, the macro program adeptly allocates the data according to the preferred number of visits to be displayed on the X-axis per page. This adaptive distribution guarantees an attractive and accurate presentation of datasets encompassing a vast quantity of visits.

```
%macro AdaptPlotGen   (DSETIN      =    /* Input Dataset  */
                      ,DSETOUT     =    /* Output Dataset */
                      ,NUMXAXIS    = 10 /* Number of Visits on X-Axis */
                      ,SORTVARS    =    /* Sorting Variables for Descriptive Statistics */
                      ,ANALYSISVAR =    /* Required Numeric Variable for Analysis */
                      ,SUBGRP      =    /* Subgroup Variable */
                      ,TICKNUM     = 15 /* Minimum Number of Ticks on Y-Axis */
                      ,PLOTGROUP   =    /* Grouping Variable for Values of Data */
                      ,FIGTYPE     =    /* Type of Figure (Mean, Median) */
                      ,YAXISLABEL  =    /* Label for Y-Axis */
                      ,XAXISLABEL  =    /* Label for X-Axis */
                      ,DEBUGFL     = Y  /* Y (Delete Temporary Datasets) or N (Keep Temporary Datasets) */
                      ,YAXISDEC    = 2  /* Number of Decimals on Y-Axis */
                      );
```

Figure 1. Macro Parameters

The %AdaptPlotGen macro program enables dynamic generation of adaptive line plots, taking input parameters which are described above, to provide versatile data visualization tailored to user requirements.

```
proc sql noprint;
      select count(distinct(&subgrp)) into :subgrpn trimmed from &prefix.&dsetin.1;
quit;

%do i=1 %to &subgrpn.;
    proc sql noprint;
       select count(distinct(&avisit)) into :maxvis trimmed
       from &prefix.&dsetin.1
       where &subgrp.n=&i;

       select mod(&maxvis,&numxaxis) into :mod trimmed
       from &prefix.&dsetin.1
       where &subgrp.n=&i;
    quit;
```

Figure 2. Calculating Maximum Visits and Modulus for Smart Pagination

In the first stage of the macro program, the logic behind splitting the figure across multiple pages is applied, focusing on printing the figure output for each subgroup (&subgrp) The process comprises the following steps:

- **COUNTING UNIQUE VISITS AND SUBGROUPS**
  The program starts by calculating the unique number of visits and unique number of subgroups present in the Analysis dataset. This information is essential for determining how to distribute the data across multiple pages.
- **CALCULATING THE REMAINDER**
  The remainder calculation is a pivotal step in optimizing the distribution of visits across pages for a more meaningful presentation. This suboptimal outcome is addressed by the program, which calculates the remainder to understand the distribution of the remaining visits. This strategic approach ensures that the presentation is tailored to the actual data distribution, preventing unnecessary page breaks and fostering a more cohesive and efficient display.

```
%do j=1 %to &maxvi %by &numxaxis;
     data &prefix.&dsetin.2;
            set &prefix.&dsetin.1(
            where=(&subgrp.n=&i and &j.-(&j=1)<=ord<&j+&numxaxis
            %if &j eq %eval(&maxvis-&mod-&numxaxis+1)
            and &mod ge 1 and &mod lt %eval(&numxaxis/2) %then %do;
                  %if %sysevalf(&j+&numxaxis+&mod<=(&maxvis+1)) %then %do;
                        +&mod
                  %end;
            %end;));
     run;
```

Figure 3. Thoughtful Data Selection

- **STRATEGIC VISIT ARRANGEMENT**

  In the process of arranging visits into pages, the program focuses on creating a balanced presentation. It considers whether there is a leftover portion when dividing the total visits by the desired number of visits per page. If such a remainder exists, the program adapts the total visit count accordingly. When the program approaches the final page, it integrates any remaining visits to optimize the overall presentation. Crucially, the program uses a strategic approach to determine whether the remainder is less than half of the desired number of visits per page. If it is, the program adjusts the total visits by subtracting the remainder, ensuring a harmonized distribution. On the other hand, if the remainder is equal to or greater than half of the desired visits, the program maintains the original total visit count without any modifications. This meticulous process ensures that the presentation is tailored to the nature of the data, promoting clarity and cohesiveness.

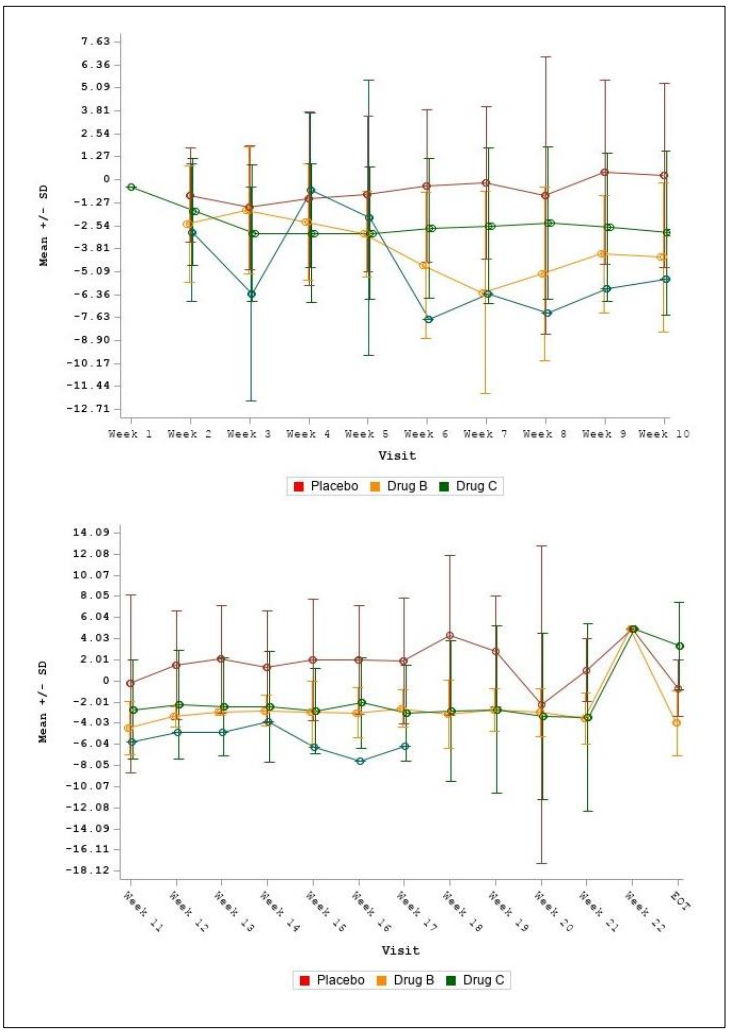

Figure 4. Output of High-Visit Figure with Smart Pagination

STAGE 2: ADAPTIVE ZERO-VALUE DISPLAY AND CUSTOMIZABLE Y-AXIS DIVISION

In the second stage of the process, the program focuses on calculating descriptive statistics for the dataset, including Minimum, Maximum, (Mean and Standard Deviation if the type of the figure Is Mean +-SD, otherwise if type is Median IQR the procedure calculates Median, Q1, Q3). These values are essential for understanding the overall trends and distribution of the data. Additionally, the program calculates the upper and lower bounds using the (Mean and Standard Deviation or Median, Q1, Q3) values. This step helps to define the range of the data, which is useful for visualizing the data in line plots.

Upon completing the calculation of descriptive statistics, the program extracts essential information from the dataset to generate specific macro variables. These macro variables encompass distinct values of visit, the maximum value of the upper bounds, and the minimum value of the lower bounds. This information is essential for the subsequent steps, where the program will utilize these macro variables to dynamically adapt the plot display based on the unique visits, upper, and lower bounds.

```
data &prefix.numaxis&j.;
      by1=input(put(%sysevalf((&max.-&min.)/&ticknum.),8.&yaxisdec.),best.);
                array range[%eval(&ticknum.+1)] a1-a%eval(&ticknum.+1);
                do i=1 to %eval(&ticknum.+1);
                    if i=1 then do;
                        range[i]=&min.;
                        z=abs(range[i]);
                        if range[i]=abs(range[i]) then sign=1;
                    end;
                    else do;
                        range[i]=input(put(range[i-1],best.),best.)+by1;
                        if abs(range[i])<z then do;
                            z=abs(range[i]);
                            if range[i]=abs(range[i]) then sign=1;
                        end;
                    end;
                  end;
      %if %sysevalf(&max.>=0) and %sysevalf(&min.<=0) %then %do;
             if z^=0 then do i=1 to %eval(&ticknum.+1);
                    if not missing(range[i]) and sign=1 then range[i]=range[i]-z;
                    else if not missing(range[i]) and sign=. then range[i]=range[i]+z;
             end;
      %end;
```

Figure 5. Algorithmic Approach to Y-Axis Tick Marks

The process comprises the following steps:

1. The part of the macro program mentioned in Figure 5 performs a series of calculations to determine the interval between tick marks on the Y-axis. This interval is computed by finding the difference between the minimum and maximum values specified. The resulting value serves as the spacing parameter for evenly distributing tick marks along the Y-axis. Subsequently, the macro generates a set of variables representing these tick marks. The initial variable is assigned the minimum value, and subsequent variables are created by adding the computed interval to the preceding variable. This systematic approach ensures a uniform distribution of tick marks.

   To accurately position the 0 point on the Y-axis, the macro identifies the tick mark with the value closest to 0. This is achieved by evaluating the absolute values of each tick mark and selecting the one with the minimum absolute value. The chosen tick mark's value is then used as a reference point. Once the closest tick mark to 0 is determined, the program dynamically adjusts all tick mark values. This adjustment involves subtracting the identified tick mark value from each variable representing the tick marks. The goal is to align the 0 point (Figure 6) precisely on the Y-axis.

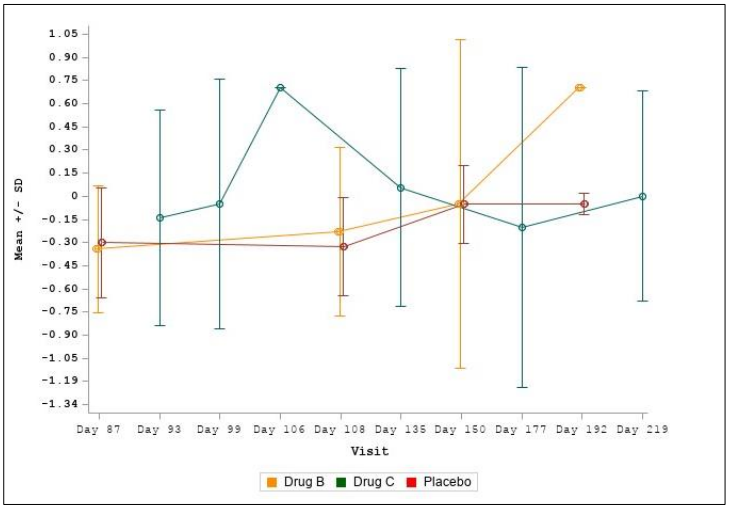

Figure 6. Graphical Representation of Truncation

2. In the realm of figure visualization, incorporating zero as a reference point on the Y-axis holds significance, especially in datasets featuring a mix of positive and negative values. This practice enhances the clarity of the figure, enabling viewers to discern the relevance and significance of zero values within the dataset. However, situations may arise where the dataset comprises solely positive or negative values, rendering the display of zero on the Y-axis unnecessary. To address this, the macro (Figure 7) incorporates a feature that intelligently determines when it is more appropriate to include or exclude zero on the Y-axis. Specifically, the macro considers cases where all values are either negative or positive, yet the interval between the first Y-axis value and zero is less than or equal to the interval between ticks. In such scenarios, the macro recommends displaying zero on the Y-axis for optimal visualization as shown in Figure 8.

```
%if %sysevalf(&min.>0) and %sysevalf(&min<=&by1) %then %do;
    if z^=0 then do i=1 to %eval(&ticknum.+1);
           if not missing(range[i]) and sign=1 then range[i]=range[i]-z;
    end;
%end;
%if %sysevalf(&max.<0) and %sysevalf(&absmax<=&by1) %then %do;
    if z^=0 then do i=1 to %eval(&ticknum.+1);
           if not missing(range[i]) and sign=. then range[i]=range[i]+z;
    end;
%end;
```

Figure 7. Additional Conditions for Showing Zero

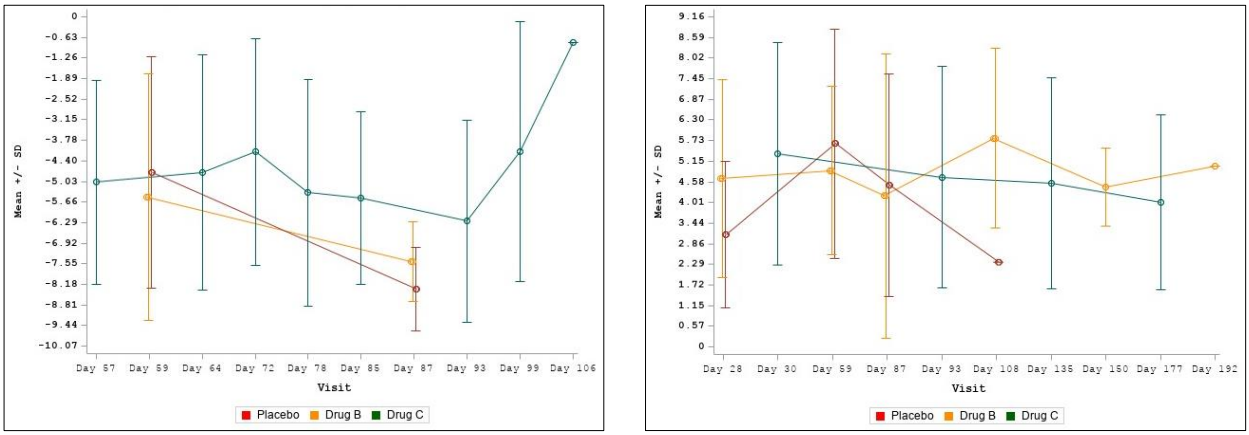

Figure 8. Smart Zero Display for Clear Figure Visualization

3. After obtaining all the essential Y-axis values, including the 0 point, the program addresses potential truncation issues that may occur during the adjustment process. Specifically, situations may arise where the first Y-axis value is greater than the specified minimum value, or the last Y-axis value is less than the specified maximum value. Such scenarios can lead to the truncation of error bars in the resulting figure. To mitigate these truncation concerns, the program takes additional measures. It calculates two supplementary values that provide insights into which part of the Y-axis data may be subject to truncation (Figure 9). There are instances where both of these supplementary values are necessary to ensure accurate representation, and there are ideal cases where neither of them needs to be added. This careful consideration and adjustment process (Figure 10) aims to prevent any truncation issues and contribute to the precision and integrity of the resulting figure.

```
if a%eval(&ticknum.+1)<%sysevalf(&max.) then
       %eval(&ticknum.+2)=a%eval(&ticknum.+1)+by1;
if a1>%sysevalf(&min.) then a0=a1-by1;
```

Figure 9. Truncation Handling for Clear Figure Visualization

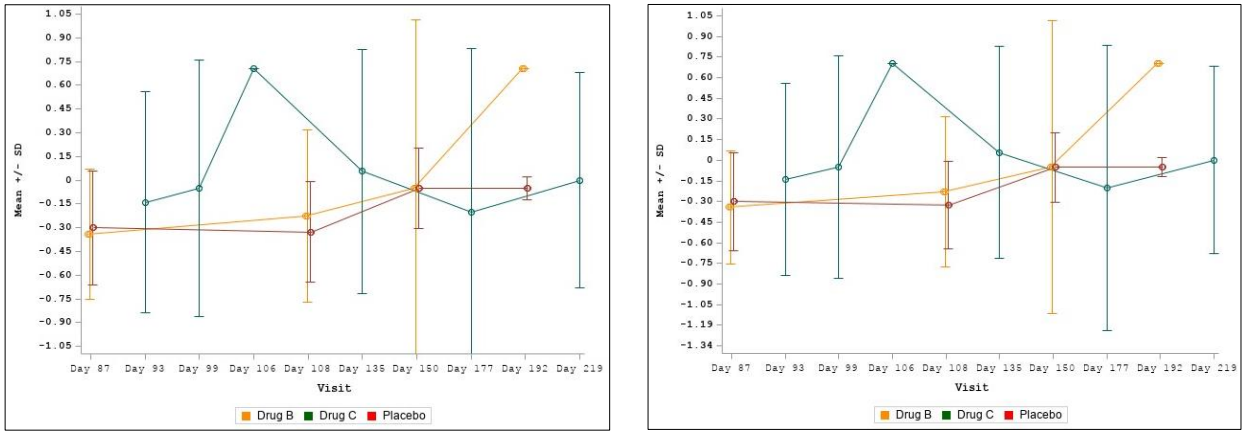

Figure 10. Truncation Handling for Clear Figure Visualization

CONCLUSION

This paper presents a SAS macro program designed for enhanced data visualization in high-visit figures. The two-stage approach begins with smart pagination, dynamically distributing data across pages based on subgroups, optimizing the presentation by considering remainders. The second stage focuses on consistent zero-value display and customizable Y-axis division, ensuring precision in aligning the zero point and addressing potential truncation issues. This comprehensive solution offers improved visual clarity, making it a valuable tool for navigating complex datasets with large numbers of visits.

REFERENCES

[1] SAS Institute Inc. 2017. Base SAS® 9.4 Procedures Guide, Seventh Edition. Cary, NC: SAS Institute Inc.
[2] Burlew, Michele M. 2014. SAS® Macro Programming Made Easy, Third Edition. Cary, NC: SAS Institute Inc.
[3] SAS Institute Inc. 2016. SAS® 9.4 ODS Graphics: Procedures Guide, Sixth Edition. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Arman Melkonyan
STATECS LLC
Email: arman.melkonyan@statecs.com
www.statecs.com