



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

Presenter: Arman Melkonyan
Associate Director, Statistical Programming

Agenda

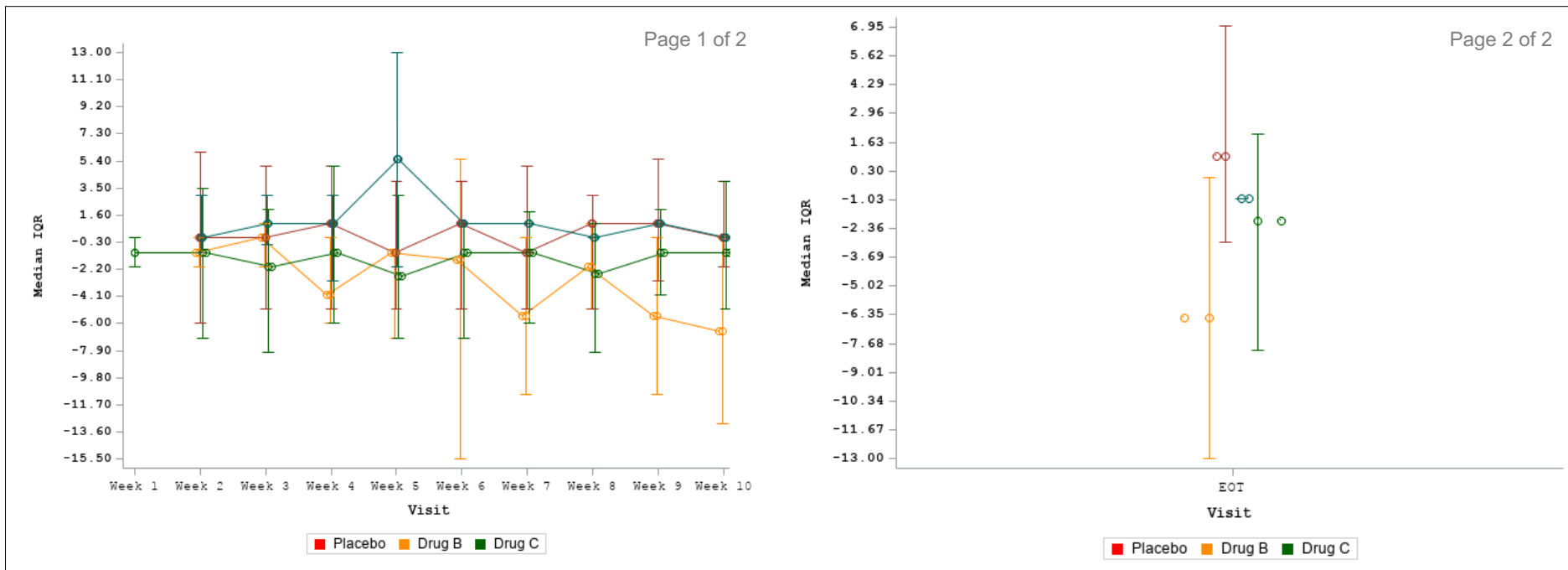
- Introduction
 - Smart Pagination Overview
 - Zero Impact: Visual Contrast
 - Adaptive Plot Generation Macro
 - Coding and Output Overview
- 

Why is it Important?

Smart Pagination Overview

- Clarity of the Visualization
- Reduced Underutilized Pages

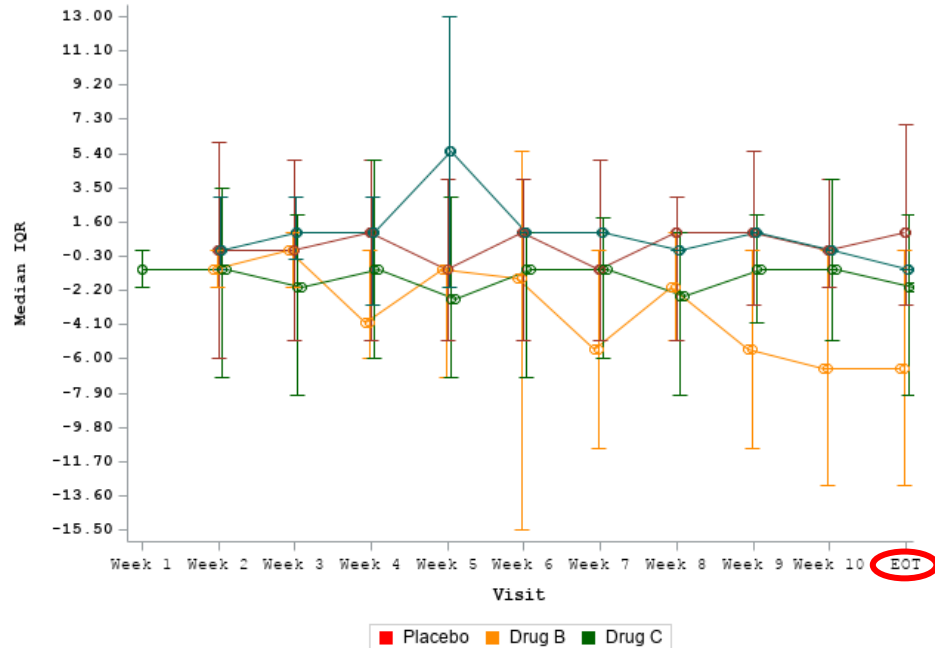
Figure 1



Smart Pagination Overview

- Clarity of the Visualization
- Reduced Underutilized Pages

Figure 2

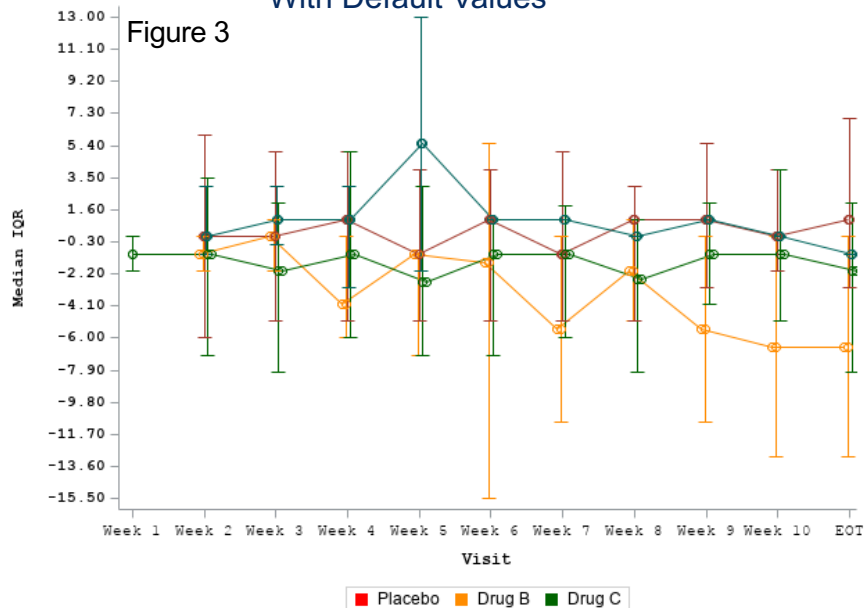


Zero Impact: Visual Contrast

- Accurate Representation
- Visual Balance
- Contextual Understanding

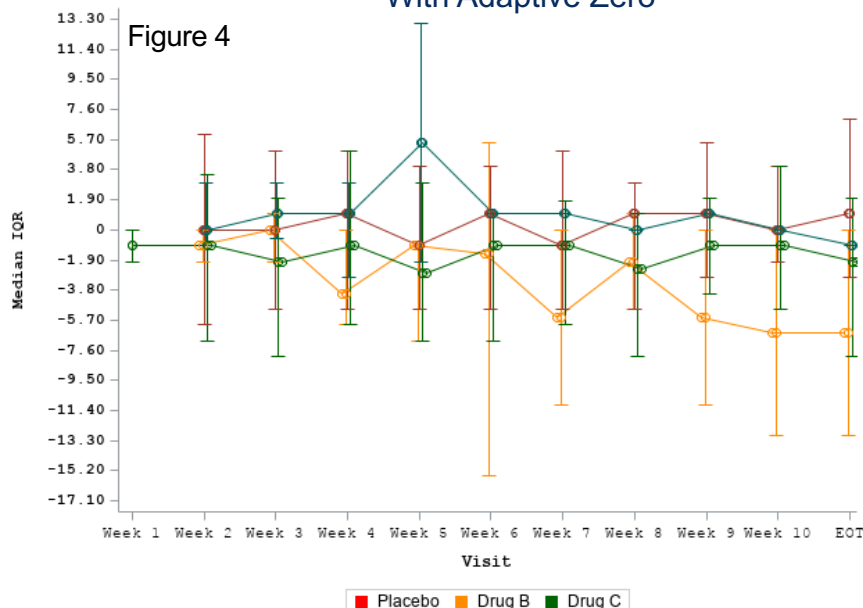
With Default Values

Figure 3



With Adaptive Zero

Figure 4



Adaptive Plot Generation Macro

Parameter Overview

```
%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 */
                        );
```

Adaptive Plot Generation Macro

Macro Parameter Validation

Ensuring Parameter Existence

```
MLOGIC(ADAPTPLOTGEN): %IF condition &dsetin eq _ is TRUE
MLOGIC(ADAPTPLOTGEN): %PUT ERROR:(AdaptPlotGen) MACRO
    PARAMETER DSETIN IS A REQUIRED PARAMETER, PROVIDE A
    DATASET NAME.
ERROR:(AdaptPlotGen) MACRO PARAMETER DSETIN IS A REQUIRED
PARAMETER, PROVIDE A DATASET NAME.
MLOGIC(ADAPTPLOTGEN): %GOTO EXIT (label resolves to EXIT).
MLOGIC(ADAPTPLOTGEN): Ending execution.
```

Variable Existence in Sorting

```
MLOGIC(ADAPTPLOTGEN): %IF condition
    %check(variable=&plotgroup) = 1 is TRUE
MLOGIC(ADAPTPLOTGEN): %PUT ERROR:(AdaptPlotGen) MACRO
    PARAMETER PLOTGROUP DOES NOT APPEAR IN SORTVARS MACRO
    PARAMETER.
ERROR:(AdaptPlotGen) MACRO PARAMETER PLOTGROUP DOES NOT APPEAR
IN SORTVARS MACRO PARAMETER.
MLOGIC(ADAPTPLOTGEN): %GOTO EXIT (label resolves to EXIT).
MLOGIC(ADAPTPLOTGEN): Ending execution.
```

Checking Parameter Type

```
MLOGIC(ADAPTPLOTGEN): %IF condition
    %vartype(&dsetin,&ANALYSISVAR)=C is TRUE
MLOGIC(ADAPTPLOTGEN): %PUT ERROR:(AdaptPlotGen) MACRO
    PARAMETER &ANALYSISVAR IS NOT NUMERIC.
SYMBOLGEN: Macro variable ANALYSISVAR resolves to avalc
ERROR:(AdaptPlotGen) MACRO PARAMETER avalc IS NOT NUMERIC.
MLOGIC(ADAPTPLOTGEN): %GOTO EXIT (label resolves to EXIT).
MLOGIC(ADAPTPLOTGEN): Ending execution.
```

Blocking Dataset Replacement

```
MLOGIC(ADAPTPLOTGEN): %IF condition %quppercase(&dsetout) eq
    %quppercase(%scan(&dsetin, 1, !!)) is TRUE
MLOGIC(ADAPTPLOTGEN): %PUT ERROR:(AdaptPlotGen) THE OUTPUT
    DATASET IS NAMED IDENTICALLY TO THE INPUT DATASET.
ERROR:(AdaptPlotGen) THE OUTPUT DATASET IS NAMED IDENTICALLY TO
THE INPUT DATASET.
MLOGIC(ADAPTPLOTGEN): %GOTO EXIT (label resolves to EXIT).
MLOGIC(ADAPTPLOTGEN): Ending execution.
```

Conditional Note for Values Exceeding a Threshold

```
MLOGIC(ADAPTPLOTGEN): %IF condition &numxaxis>&maxvis is TRUE
MLOGIC(ADAPTPLOTGEN): %PUT NOTE:(AdaptPlotGen) VALUE OF MACRO
    VARIABLE NUMAXIS IS GREATER THAN MAXIMUM NUMBER OF VISITS
    FOR &SUBGRPVAL.
NOTE:(AdaptPlotGen) VALUE OF MACRO VARIABLE NUMAXIS IS GREATER
THAN MAXIMUM NUMBER OF VISITS FOR Activated Partial
Thromboplastin Time (sec) - Coagulation
```

Coding Overview

Creating Macro Variables for Smart Pagination

```
/* Counting Unique Number of Subgroups */
proc sql noprint;
    select count(distinct(&subgrp)) into :subgrpn trimmed from &prefix.&dsetin.1;
quit;
```

Number of Subgroups

```
/* Iteration of Subgroups */
%do i=1 %to &subgrpn.;
```

```
/* Counting Unique Number of Visits and Calculation of the Remainder */
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;
```

Maximum Number of Visits

Remainder Calculation

```
/* Conditions for Visit Number Adjustment */
```

```
%if &mod ge 1 and &mod lt %eval(&numxaxis/2) %then %do;
    %let maxvi = %sysevalf(&maxvis- &mod);
%end;
%else %do;
    %let maxvi = &maxvis;
%end;
```

Conditional Visit Number Adjustments

Coding Overview

Navigating High-Visit Figures with Smart Pagination

- Data Overview
- Remainder Consideration
- Dynamic Page Adjustment

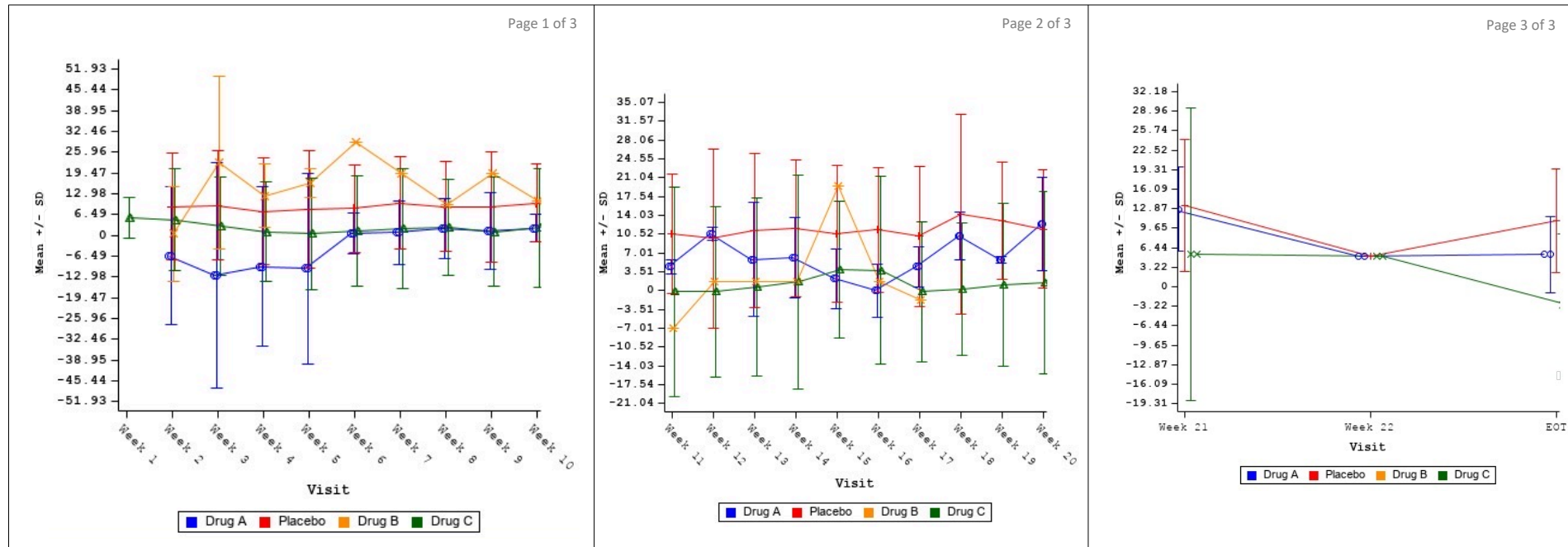
```
%do j=1 %to &maxvi.%by &numxaxis;

    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;
            +&mod.
        %end;
    ));
run;
```

Adaptive Plot Generation Macro

Output of High-Visit Figure Without Smart Pagination

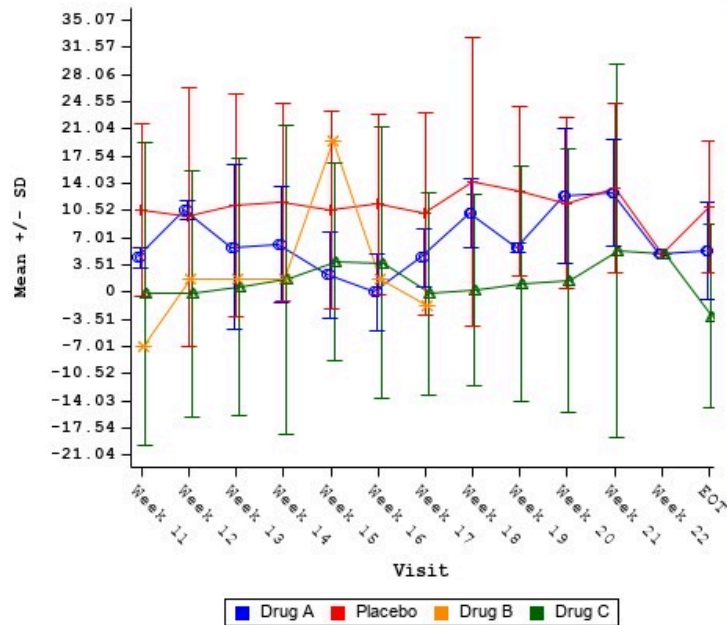
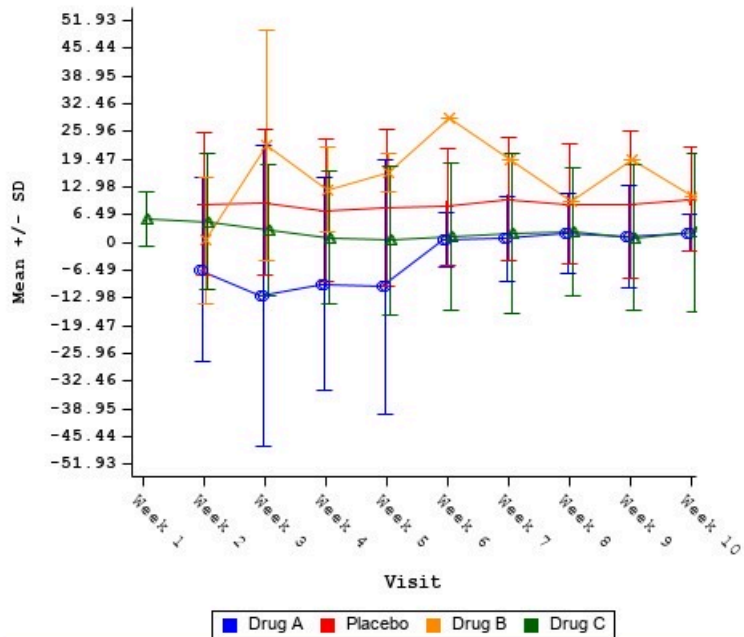
Figure 5



Adaptive Plot Generation Macro

Output of High-Visit Figure with Smart Pagination

Figure 6



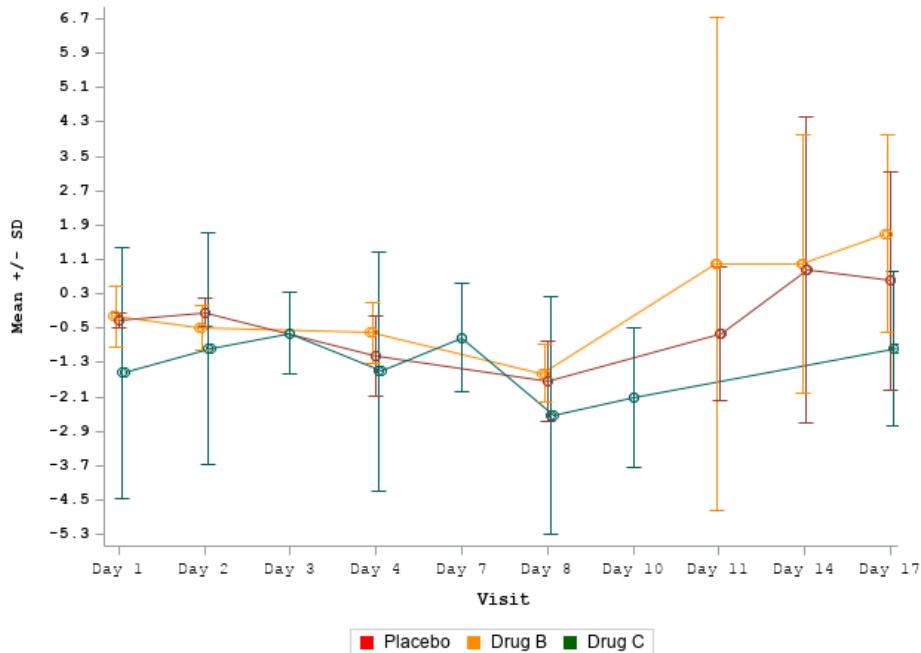
Adaptive Plot Generation Macro

Output Without Adaptive Zero

Figure 7



ZERO's secret: Can we Unlock it?



Output 1

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16
-5.3	-4.5	-3.7	-2.9	-2.1	-1.3	-0.5	0.3	1.1	1.9	2.7	3.5	4.3	5.1	5.9	6.7

Adaptive Plot Generation Macro

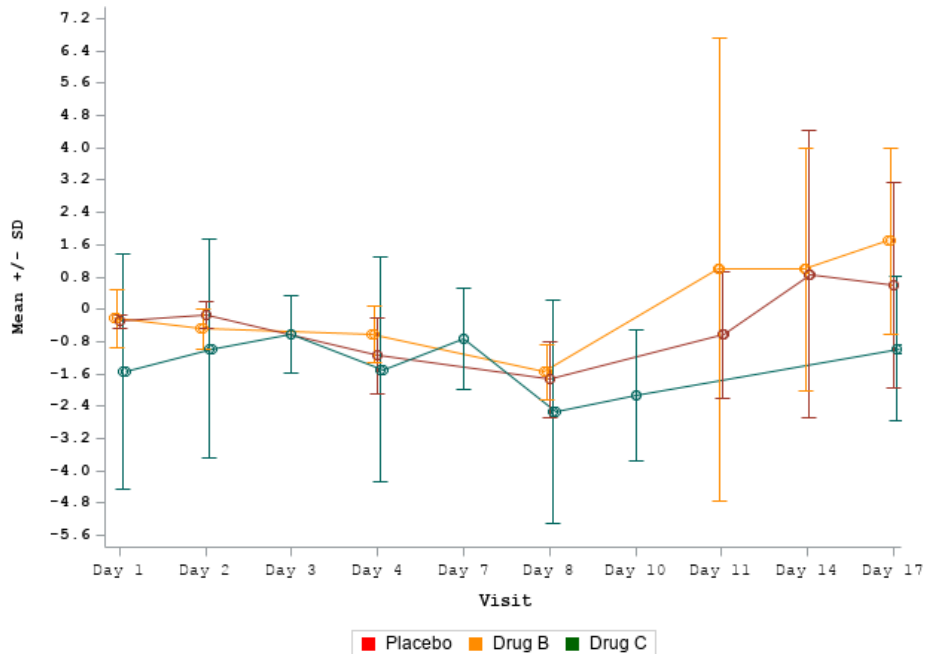
Output With Adaptive Zero

Figure 8

Derivation of Adaptive Zero

```
%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;
```

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



Output 2

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17
-5.6	-4.8	-4	-3.2	-2.4	-1.6	-0.8	0	0.8	1.6	2.4	3.2	4	4.8	5.6	6.4	7.2

Conclusion

- By dynamically calculating the optimal distribution of visits per page, the method prevents underutilized pages and ensures an organized and visually appealing representation.
- The program's ability to include zero within the Y-axis range enhances the clarity of visualizations, especially in cases where both positive and negative values are present.
- The two-stage approach, as outlined in the paper, addresses specific challenges encountered by researchers, programmers, and data analysts when dealing with complex clinical research datasets.

Thank You

Questions?

 arman.melkonyan@statecs.com

 <https://statecs.com>

