PharmaSUG 2019 - Paper DV-021

Applying an Experimental GTL Feature to CONSORT Diagrams

Shane Rosanbalm, Rho, Inc.

ABSTRACT

SAS added an experimental feature to the TEXTPLOT statement in GTL as part of 9.4 M3. When the OUTLINE option was invoked, this experimental feature allowed the user to capture a dataset with information about where the outline was being drawn using the OUTFILE and OUTID options.

This paper is about the application of the experimental OUTFILE and OUTID options in an attempt to make the creation of CONSORT diagrams a little less labor intensive.

INTRODUCTION

A brief history of CONSORT diagrams in SAS.

- 2010
  - The CONSORT statement, an evidence-based, minimum set of recommendations for reporting randomized trials, was published. This included a recommended Flow diagram of the progress through the phases of a parallel randomized trial of two groups (that is, enrollment, intervention allocation, follow-up, and data analysis).
- 2013
  - Art Carpenter and Dennis Fisher publishes a paper at WUSS which describes how to produce CONSORT diagrams programmatically. The process utilized an RTF template with placeholder text where the counts should go. The RTF file is then scanned and TRANSLATE functions are used to update the placeholder text with the desired counts.
- 2016
  - Sanjay Matange published a blog post about creating CONSORT diagrams using SGPLOT. The process involves creating datasets with coordinates for where various parts of the diagram should be drawn (e.g., the corners of the rectangles are specified as 4 coordinate pairs and then a POLYGON statement draws the rectangle).
- 2018
  - Sanjay Matange and Prashant Hebbar jointly publish a PharmaSUG paper which further fleshes out the ideas first published in the 2016 blog post.
  - Shane Rosanbalm publishes a SESUG paper in which efficiencies are added to the SGPLOT approach. The process involves calling some helper macros that assist in the construction of the various datasets with coordinates.

LAYOUT INEFFICIENCIES

When creating a CONSORT diagram with SAS using the SGPLOT approach, laying out the diagram is rather labor intensive and fiddly. Specifying the coordinates correctly on the first try is virtually impossible. And the dependencies of all of the boxes on one another often causes cascades of side effects when you reposition or resize any one box. Anything that could reduce the amount of time and effort involved in performing the initial layout of the diagram would be useful.

USING AN EXPERIMENTAL GTL FEATURE

As part of 9.4 M3, the options OUTFILE and OUTID were added to the GTL statement TEXTPLOT. Specifying these options result in an output dataset being created with information about where text outlines have been drawn. The idea behind this paper is to leverage the information in this outlines dataset to generate specifications for an initial rough layout for a CONSORT diagram. The goal is not to

generate perfect specifications -- that seems a bit unrealistic. But if we can get the positions and sizes of each the boxes even *close* to right, this should speed up the specification process quite a bit.

GETTING STARTED WITH EXCEL

The process that I've created to facilitate the initial layout of CONSORT diagrams begins with Excel. This is where you enter your quick and dirty information about your diagram. You need only specify four pieces of information for each box:

1. A numeric boxId.
2. The row that the box goes in.
3. The col(umn) that the box goes in.
4. The roughText to be displayed in the box.

	boxId	row	col	roughText
2	1	1	1.5	Assessed for Eligibility (n=xxx)
3	2	2	1.5	Randomized (n=xxx)
4	3	1.5	2.5	Excluded (n=xx)~* Reason text about yeah long (n=xx)~* Reason 2 (n=xx)~* Reason 3 (n=xx)
5	11	3	1	Allocated to xxxxxxx (n=xx)~* Received allocated drug (n=xx)~* Did not receive allocated drug (n=x)
6	12	3	2	Allocated to xxxxxxx (n=xx)~* Received allocated drug (n=xx)~* Did not receive allocated drug (n=x)
7	13	3	3	Allocated to xxxxxxx (n=xx)~* Received allocated drug (n=xx)~* Did not receive allocated drug (n=x)
8	21	4	1	Discontinued drug (n=xx) due to:~* Reason text (n=x)~* Reason text (n=x)~* Reason text (n=x)~* Reason text (n=x)
9	22	4	2	Discontinued drug (n=xx) due to:~* Reason text (n=x)~* Reason text (n=x)~* Reason text (n=x)~* Reason text (n=x)
10	23	4	3	Discontinued drug (n=xx) due to:~* Reason text (n=x)~* Reason text (n=x)~* Reason text (n=x)~* Reason text (n=x)
11	31	5	1	FAS (n=xx)~* Excluded from FAS~ (n=x)~* Safety set (n=xx)~* Excluded from SS~ (n=x)
12	32	5	2	FAS (n=xx)~* Excluded from FAS~ (n=x)~* Safety set (n=xx)~* Excluded from SS~ (n=x)
13	33	5	3	FAS (n=xx)~* Excluded from FAS~ (n=x)~* Safety set (n=xx)~* Excluded from SS~ (n=x)
14	41	6	0.75	Something~good (n=xx)
15	42	6	1.25	Something~bad (n=xx)
16	43	6	1.75	Something~good (n=xx)
17	44	6	2.25	Something~bad (n=xx)
18	45	6	2.75	Something~good (n=xx)
19	46	6	3.25	Something~bad (n=xx)

Figure 1. Sample Excel file with rough CONSORT contents

You'll notice a couple of things about these Excel specifications.

- The roughText truly is rough.
    - Having roughly the right length of text strings is much more important than having the exact wording.
- The position of each box is specified with a row and col pair.
    - The integer values correspond to the boxes that form the main structure of the diagram. Use integers for boxes that are meant to line up with other boxes.
    - The non-integer values are for the fiddly bits of the diagram that don't fit nicely into a row or column. Use non-integer values to communicate that a box goes *somewhere over there*.

A THROW-AWAY PLOT

The SAS program reads in the Excel file and creates a really rough draft layout using the row and col values.

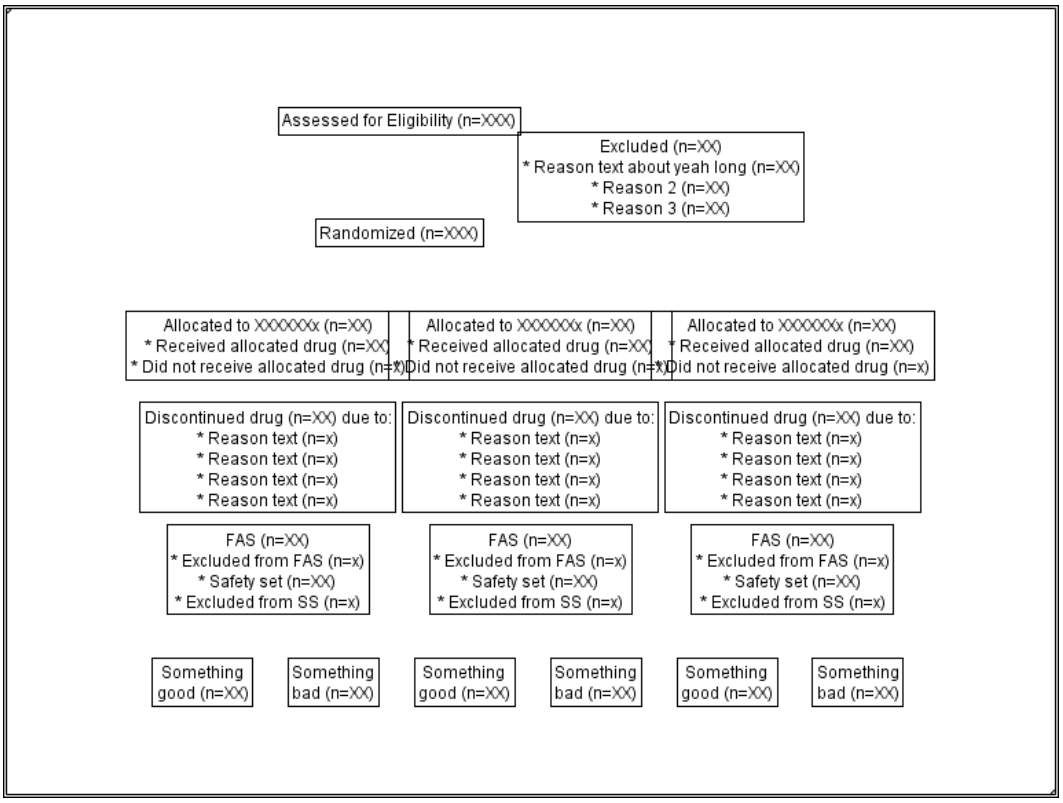

Figure 2. Ugly draft layout using ROW and COL values for positioning

This initial layout is a throw-away. It's only purpose is to allow us to create the OUTFILE which contains the information about the width and height of each of the outlines that were drawn. The OUTFILE is created as a CSV, which when viewed in Excel looks like this.

1	NAME	OUTID	WIDTH	HEIGHT	DESCENT	RELATIVEV	RELATIVEH	RELATIVEC	DATAWID	DATAHEIG	DATADESCENT
2	TEXT	1	146.1832	58.09091	5.003497	0.191339	0.101558	0.008747	19.13393	10.15575	0.874737
3	TEXT	2	102.1257	19.02273	5.003497	0.133672	0.033257	0.008747	13.36723	3.325652	0.874737
4	TEXT	3	179.2264	58.09091	5.003497	0.23459	0.101558	0.008747	23.45896	10.15575	0.874737
5	TEXT	11	183.2317	45.06818	5.003497	0.239832	0.078791	0.008747	23.9832	7.879052	0.874737
6	TEXT	12	183.2317	45.06818	5.003497	0.239832	0.078791	0.008747	23.9832	7.879052	0.874737
7	TEXT	13	183.2317	45.06818	5.003497	0.239832	0.078791	0.008747	23.9832	7.879052	0.874737
8	TEXT	21	159.2003	71.11364	5.003497	0.208377	0.124325	0.008747	20.83773	12.43245	0.874737
9	TEXT	22	159.2003	71.11364	5.003497	0.208377	0.124325	0.008747	20.83773	12.43245	0.874737
10	TEXT	23	159.2003	71.11364	5.003497	0.208377	0.124325	0.008747	20.83773	12.43245	0.874737
11	TEXT	31	105.1296	84.13636	5.003497	0.137604	0.147092	0.008747	13.76042	14.70915	0.874737
12	TEXT	32	105.1296	84.13636	5.003497	0.137604	0.147092	0.008747	13.76042	14.70915	0.874737
13	TEXT	33	105.1296	84.13636	5.003497	0.137604	0.147092	0.008747	13.76042	14.70915	0.874737
14	TEXT	41	61.07199	136.2273	5.003497	0.079937	0.23816	0.008747	7.993716	23.81596	0.874737
15	TEXT	42	111.1374	32.04545	5.003497	0.145468	0.056024	0.008747	14.54678	5.602352	0.874737
16	TEXT	43	61.07199	32.04545	5.003497	0.079937	0.056024	0.008747	7.993716	5.602352	0.874737
17	TEXT	44	58.06806	32.04545	5.003497	0.076005	0.056024	0.008747	7.600532	5.602352	0.874737
18	TEXT	45	61.07199	32.04545	5.003497	0.079937	0.056024	0.008747	7.993716	5.602352	0.874737
19	TEXT	46	58.06806	32.04545	5.003497	0.076005	0.056024	0.008747	7.600532	5.602352	0.874737

Figure 3. Resulting CSV file with information about where each OUTLINE was drawn

IMPROVING THE VERTICAL SPACING OF THE BOXES

Now that we know the height of each of the boxes (DATAHEIGHT), it becomes a simple algebra problem to calculate how much total vertical space the boxes use. We then divide the remaining space up between the rows to equally space the boxes vertically.

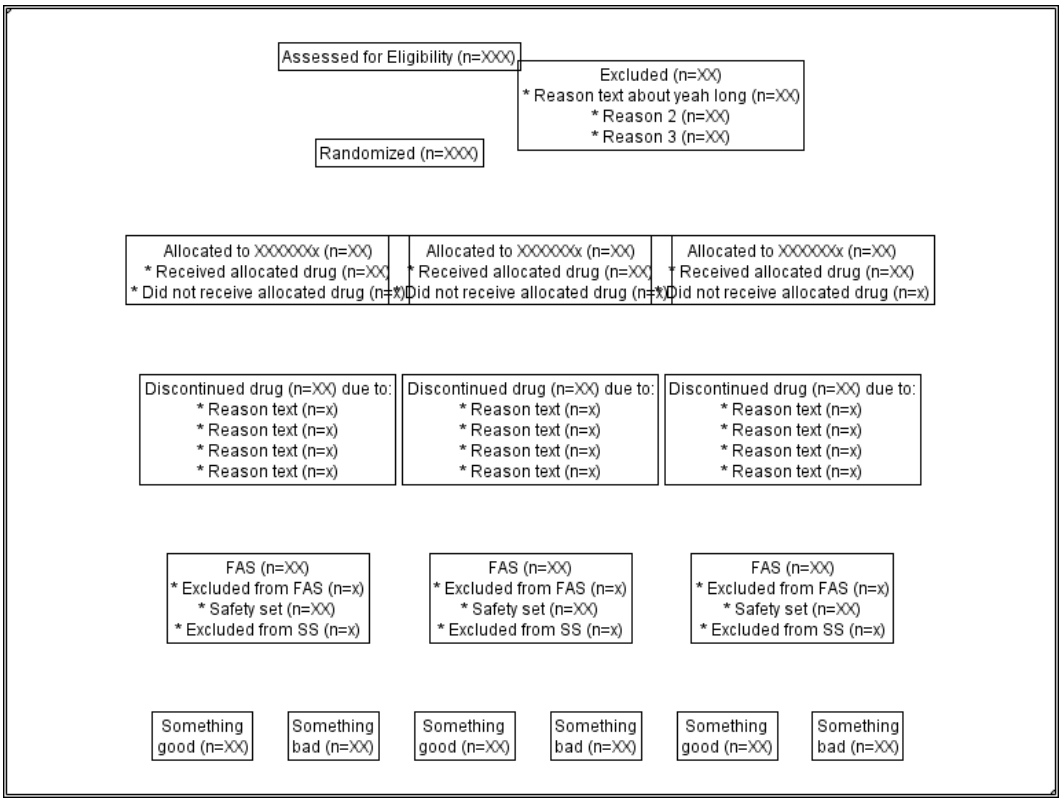

Figure 4. Improving the vertical spacing of the boxes

IMPROVING THE HORIZONTAL SPACING OF THE BOXES

We also now know the width of each of the boxes (DATAWIDTH), so it is equally as simple to calculate how much total horizontal space the boxes use. We then divide the remaining space up between the columns to equally space the boxes horizontally.

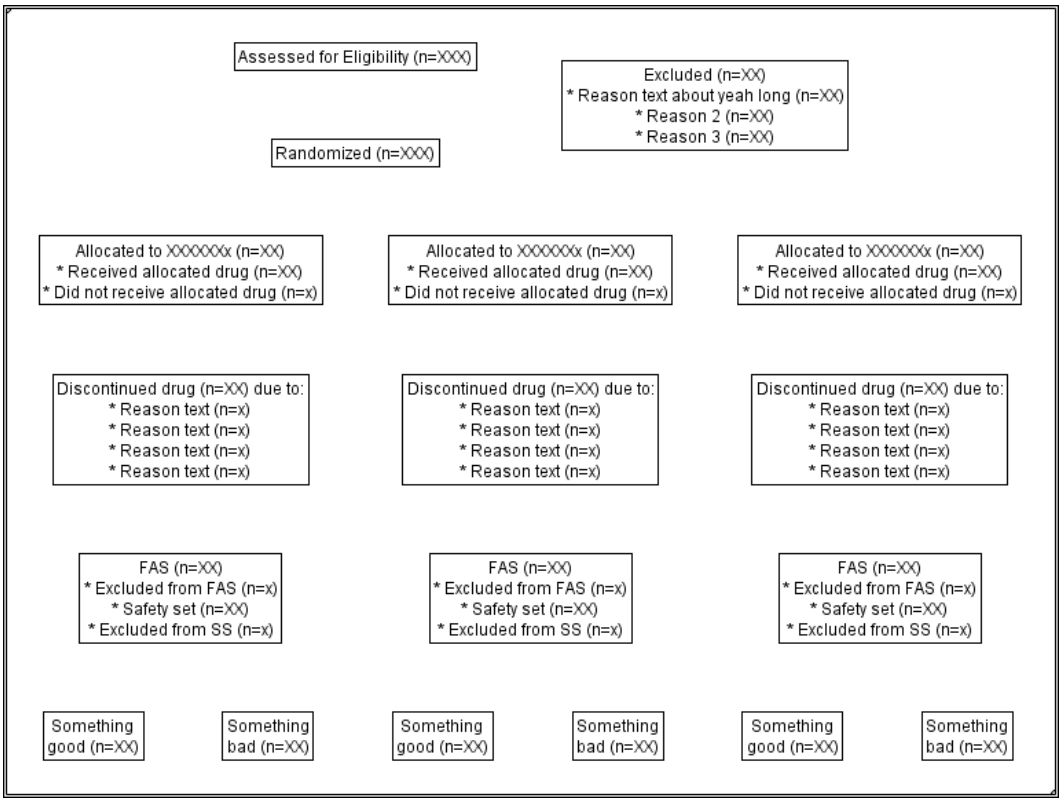

Figure 5. Improving the horizontal spacing of the boxes

Now that we know roughly where the boxes should go, we need to be able to incorporate this information back into the SGPLOT-based process for generating CONSORT diagrams. Fortunately, the SAS program ends with the creation of a putstring variable, the content of which can be easily copy/pasted into the datalines of the emptyBoxes data step in the SGPLOT-based CONSORT diagram program.

THE FINAL RECOMMENDATION

	putstring
1	01, 33.33, 4.39, 19.13, 3.33
2	02, 33.33, 16.51, 13.37, 3.33
3	03, 66.67, 7.04, 23.46, 10.16
4	11, 16.67, 28.62, 23.98, 7.88
5	12, 50.00, 28.62, 23.98, 7.88
6	13, 83.33, 28.62, 23.98, 7.88
7	21, 16.67, 45.29, 20.84, 12.43
8	22, 50.00, 45.29, 20.84, 12.43
9	23, 83.33, 45.29, 20.84, 12.43
10	31, 16.67, 66.51, 13.76, 14.71
11	32, 50.00, 66.51, 13.76, 14.71
12	33, 83.33, 66.51, 13.76, 14.71
13	41, 8.33, 90.00, 7.99, 5.60
14	42, 25.00, 90.00, 7.60, 5.60
15	43, 41.67, 90.00, 7.99, 5.60
16	44, 58.33, 90.00, 7.60, 5.60
17	45, 75.00, 90.00, 7.99, 5.60
18	46, 91.67, 90.00, 7.60, 5.60

Figure 6. Copy/paste the resulting configuration metadata into an efficient program

To see the putstring values in action, check out the program smooth.sas in this repository. The smooth program makes use of some macros that were originally developed in the sas-consort-sgplot repository. The current repository's versions of the macros have been modified ever so slightly from the original to better fit with this repository's top-down approach to y values (the original SGPLOT-based approach used a bottom-up approach to y values).

DOWNLOAD

The material presented in this paper is available for download at GitHub.

https://github.com/srosanba/sas-consort-experimental

Once at the link, simply select the green [Clone or download] button at right and select the [Download ZIP] option. This contains the ROUGH.XLSX for doing your rough layout, ROUGH.SAS to transform the Excel row/col values into recommendations, and SMOOTH.SAS in which you can paste the recommendations and generate an efficient SGPLOT-based CONSORT diagram.

CONCLUSION

Generating the layout for a CONSORT diagram created with SGPLOT is not trivial. The above process allows the user to quickly generate rough specifications using nothing more than row and column values for positioning. Incorporating this into your CONSORT process should reduce the amount of time spent generating the initial layout.

REFERENCES

Carpenter A. and Fisher D. 2013. "Reading and Writing RTF Documents as Data: Automatic Completion of CONSORT Flow Diagrams" *Proceedings of MWSUG 2018*. Available at https://www.mwsug.org/proceedings/2013/RX/MWSUG-2013-RX05.pdf

Mallavarapu A. and Shults D. 2016. "CONSORT Diagram: Doing it with SAS" *Proceedings of PhUSE 2016.* Available at https://www.lexjansen.com/phuse/2016/pp/PP03_ppt.pdf

Hebbar P. and Matange S. 2018. "CONSORT Diagrams with SG Procedures" *Proceedings of PharmaSUG 2018*. Available at https://www.lexjansen.com/pharmasug/2018/DV/PharmaSUG-2018-DV24.pdf

He H. 2018. "An Exploratory Way to Draw a Flow Diagram for the Subject Disposition of a Two-Arm Randomized Trial Using SAS® 9.4 M3" *Proceedings of PharmaSUG 2018*. Available at https://www.lexjansen.com/pharmasug/2018/AD/PharmaSUG-2018-AD28.pdf

Rosanbalm S. 2018 “CONSORT Diagrams in SGPLOT: Adding Efficiencies” *Proceedings of SESUG 2018*. Available at https://www.lexjansen.com/sesug/2018/SESUG2018_Paper-271_Final_PDF.pdf

CONTACT INFORMATION

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

Shane Rosanbalm
Rho, Inc.
srosanba@gmail.com
https://github.com/srosanba/sas-consort-experimental