#SASGF

VIRTUAL

SAS® GLOBAL FORUM 2021

AMERICAS | MAY 18 – 20 ASIA PACIFIC | MAY 19 – 20 EMEA | MAY 25 – 26

Methods of a Fully Automated CONSORT Diagram Macro

Jeffrey Meyers, Mayo Clinic

Jeff has worked in the Cancer Center Statistics division of Mayo Clinic for 11 years where he has focused on developing macros, creating graphics, and producing reports.

Jeff has presented his macro programs at several conferences including PharmaSUG, Midwest SUG and SAS Global Forum.

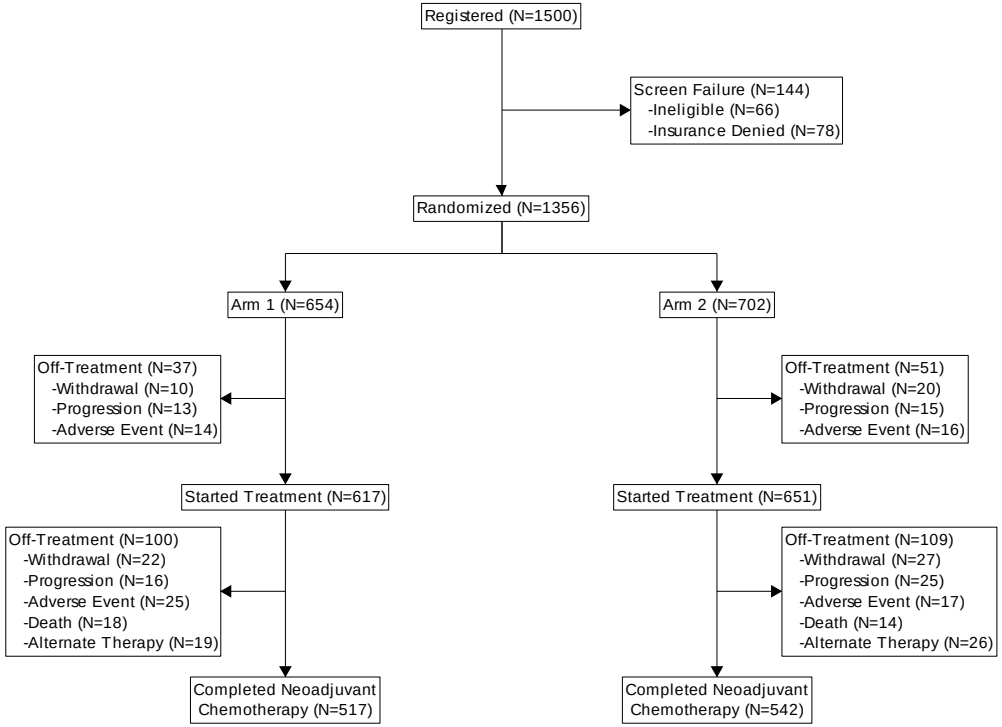
Contents

- What is a CONSORT diagram?
- Short %CONSORT macro introduction
- Graphics setup
- Automation methods
- Final steps

What is a CONSORT Diagram?

What is a CONSORT

Example



%CONSORT Introduction

%CONSORT Introduction

Key Points

- Fully automates making a CONSORT diagram with user input data
  - Calculates numbers, creates and fills in textboxes, aligns textboxes, and draws connecting lines
- Multiple options for customizing font size/color, textbox outline/fill colors, tweaking alignment, adding annotation
- Outputs to multiple image types

%CONSORT Introduction

Required Input

- DATA: input data set name
- ID: variable for unique patient identifier. Only one row-per-patient is allowed in the input data set
- NODE: One or more variables that represent a node of the CONSORT
  - Variables serve two purposes: acting as a logical operator (missing/non-missing) and containing the value that will be listed in the textbox
  - Order that variables are listed left-to-right will be printed top-to-bottom in the CONSORT

%CONSORT Introduction

Key Optional Input

- SPLIT: Variables that will make the CONSORT paths branch (or split)
  - One SPLIT variable can be listed per NODE. SPLIT variables are carried forward and do not need to be repeated
  - Values of the SPLIT variables will be used in the textboxes
- OFFREASON:
  - One or more variables can be specified per NODE containing the reason a patient went off-treatment
  - Last variables are carried forward to remaining NODEs
  - Label of variable is used as header for textbox and values of variable are used in a list within the textbox

%CONSORT Introduction

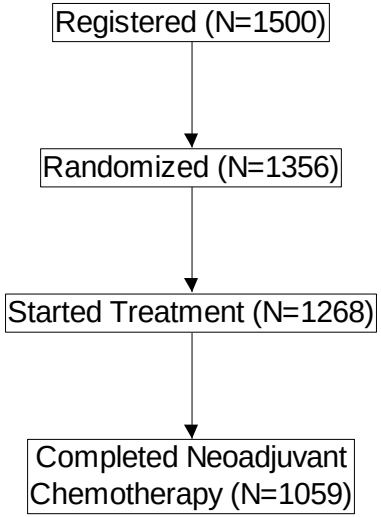
EXAMPLE Data Set

id	Treatment Arm	Sex	Smoking Status	Screen Failure	reg	rand	treated	neo	rt	surg	adj	comp	Off-Treatment
1	Arm 1	Female	Former		Registered	Randomized	Started Treatment	Completed Neoadjuvant~Chemotherapy		Completed Surgery	Started Adjuvant Therapy	Completed All Therapy	
2	Arm 2	Male	Current		Registered	Randomized	Started Treatment	Completed Neoadjuvant~Chemotherapy	Completed Neoadjuvant RT	Completed Surgery	Started Adjuvant Therapy	Completed All Therapy	
3	Arm 1	Female	Current		Registered	Randomized	Started Treatment	Completed Neoadjuvant~Chemotherapy		Completed Surgery	Started Adjuvant Therapy		Withdrawal
4	Arm 2	Male	Current		Registered	Randomized	Started Treatment	Completed Neoadjuvant~Chemotherapy	Completed Neoadjuvant RT	Completed Surgery	Started Adjuvant Therapy	Completed All Therapy	
5	Arm 2	Male	Current		Registered	Randomized	Started Treatment	Completed Neoadjuvant~Chemotherapy	Completed Neoadjuvant RT	Completed Surgery	Started Adjuvant Therapy	Completed All Therapy	

Code available in paper to create EXAMPLE data set

%CONSORT Introduction

Basic Example

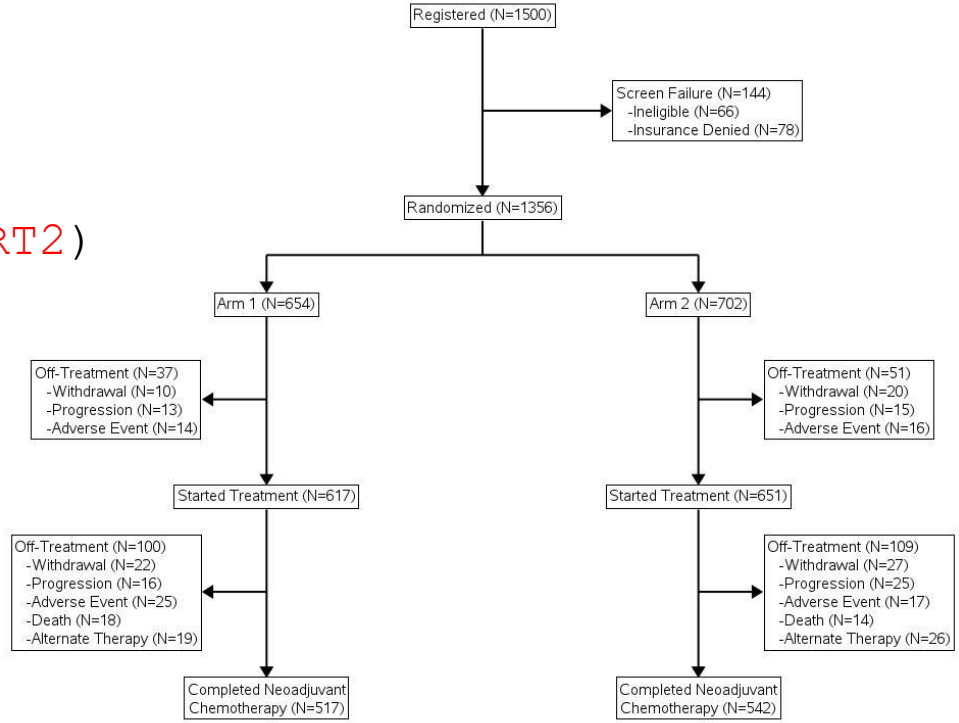



```
%CONSORT(DATA=EXAMPLE, ID=ID, NODE=REG RAND TREATED NEO)
```

%CONSORT Introduction

SPLIT/OFFRSN Example

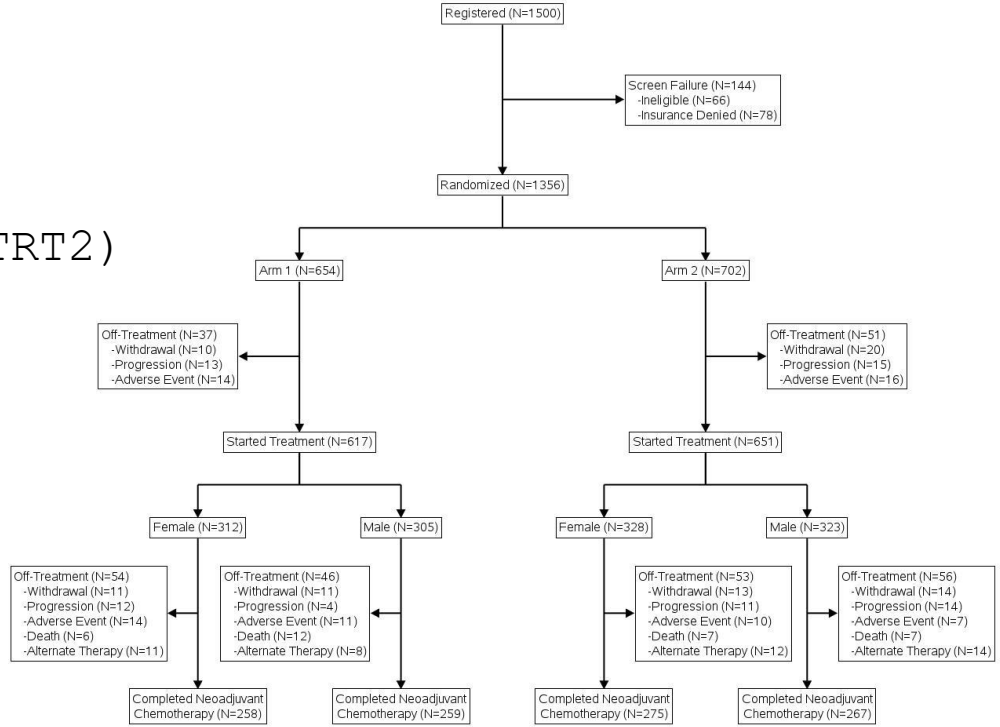
```
%CONSORT(…,
   SPLIT=|ARM,
OFFREASON=OFFTRT|OFFTRT2)
```



%CONSORT Introduction

Two SPLIT Example

```
%CONSORT(…,
   SPLIT=|ARM|SEX,
OFFREASON=OFFTRT|OFFTRT2)
```



%CONSORT Introduction

Live Examples

- SAS Studio demo

Graphics Setup

Graphics Setup

Creating a Textbox

- SGPLOT procedure's TEXT statement works well
  - BACKFILL and OUTLINE options create an opaque bordered box
  - SPLITJUSTIFY, SPLITCHAR, SPLITPOLICY=ALWAYS allow appropriately aligned text with controllable line breaks
  - Textbox is automatically sized by SGPLOT to match the text
  - POSITION allows user specified anchor point for the box at the x/y-coordinates
  - Requires variables X, Y and TEXT

Graphics Setup

Creating the Connecting Lines

- SGPLOT procedure's SERIES statement works well
  - Two different series plots: one with and one without arrowheads
  - Unique ID variable for each line segment is used with the GROUP option
  - Requires X and Y variables for coordinates

Graphics Setup

Graph Space

- X and Y axes are set from 0-100
- Columns and rows are allocated a percentage of the space
- All graph borders, axes, and labels are turned off

Automation Methods

Automation Methods

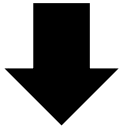
What is Needed

- There are four primary data items to be automated
  - Determine the unique patient paths through the trial
  - Derive the text and counts for each textbox
  - Determine the parent-child links between textboxes
  - Determine the x/y coordinates of each textbox

Automation Methods

Patient Paths

ID	Arm	Reg	Rand	Treated	Neo	Offrsn
19	Arm 2	Registered	Randomized	Started Treatment		Withdrawal



ID	Node	Phase	Order	Label	Split1	Offrsn	Off_order	Off_trt
19	1.01	1	1	Registered				0
19	2.01	1	1	Randomized				0
19	3.02	2	2	Started Treatment	Arm 2			0
19	4.02	2	2	Arm 2	Arm 2			0
19	5.02	2	2	Off-Treatment	Arm 2	Withdrawal	1	1

Automation Methods

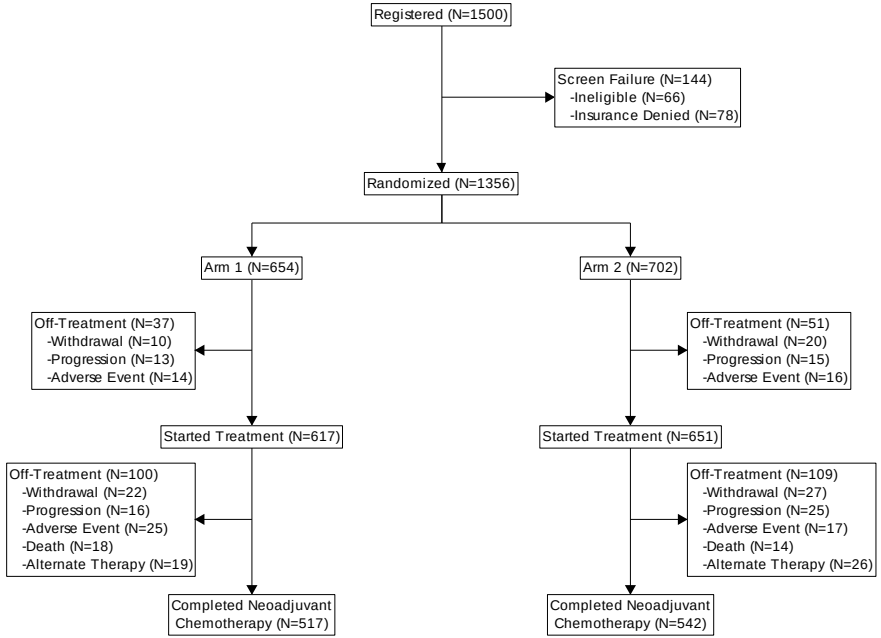
Patient Paths

Unique paths through entire study

Path	Step1	Step2	Step3	Step4	Step5	Last_step
1	1.01	3.01	4.02	6.02	8.02	8.02
2	1.01	3.01	4.03	6.03	8.03	8.03

Unique paths through going off-treatment

Step1	Step2	Step3	Step4	Step5	Last_step	Connect_backwards
1.01	2.01				2.01	1.01
1.01	3.01	4.02	5.02		5.02	4.02
1.01	3.01	4.03	5.03		5.03	4.03
1.01	3.01	4.02	6.02	7.02	7.02	6.02
1.01	3.01	4.03	6.03	7.03	7.03	6.03



Automation Methods

Calculating Patient Counts

```
select phase, node, label, 'BOTTOM' as position,
    count(distinct id) as n,
        case(missing(label))
            when 0 then strip(label)||' (N='||
                        strip(put(calculated n,12.0))||')'
        else '' end as text length=1000
        from _temp4 where off_trt<1
        group by phase,node,label,position
```

Automation Methods

Calculating Patient Counts

Phase	Node	Label	Position	N	Text	Off_trt	Off_order	Offrsn
1	1.01	Registered	BOTTOM	1500	Registered (N=1500)			
1	2.01	Screen Failure	RIGHT	144	Screen Failure (N=144)	1		
1	2.01		RIGHT	66	-Ineligible (N=66)	1	1	Ineligible
1	2.01		RIGHT	78	-Insurance Denied (N=78)	1	2	Insurance Denied
1	3.01	Randomized	BOTTOM	1356	Randomized (N=1356)			
2	4.02	Arm 1	BOTTOM	654	Arm 1 (N=654)			
2	4.03	Arm 2	BOTTOM	702	Arm 2 (N=702)			
2	5.02	Off-Treatment	LEFT	37	Off-Treatment (N=37)	1		
2	5.02		LEFT	10	-Withdrawal (N=10)	1	1	Withdrawal
2	5.02		LEFT	13	-Progression (N=13)	1	2	Progression
2	5.02		LEFT	14	-Adverse Event (N=14)	1	3	Adverse Event
2	5.03	Off-Treatment	RIGHT	51	Off-Treatment (N=51)	1		
2	5.03		RIGHT	20	-Withdrawal (N=20)	1	1	Withdrawal
2	5.03		RIGHT	15	-Progression (N=15)	1	2	Progression
2	5.03		RIGHT	16	-Adverse Event (N=16)	1	3	Adverse Event
2	6.02	Started Treatment	BOTTOM	617	Started Treatment (N=617)			
2	6.03	Started Treatment	BOTTOM	651	Started Treatment (N=651)			
2	7.02	Off-Treatment	LEFT	100	Off-Treatment (N=100)	1		
2	7.02		LEFT	22	-Withdrawal (N=22)	1	1	Withdrawal
2	7.02		LEFT	16	-Progression (N=16)	1	2	Progression
2	7.02		LEFT	25	-Adverse Event (N=25)	1	3	Adverse Event
2	7.02		LEFT	18	-Death (N=18)	1	4	Death
2	7.02		LEFT	19	-Alternate Therapy (N=19)	1	5	Alternate Therapy
2	7.03	Off-Treatment	RIGHT	109	Off-Treatment (N=109)	1		
2	7.03		RIGHT	27	-Withdrawal (N=27)	1	1	Withdrawal
2	7.03		RIGHT	25	-Progression (N=25)	1	2	Progression
2	7.03		RIGHT	17	-Adverse Event (N=17)	1	3	Adverse Event
2	7.03		RIGHT	14	-Death (N=14)	1	4	Death
2	7.03		RIGHT	26	-Alternate Therapy (N=26)	1	5	Alternate Therapy
2	8.02	Completed Neoadjuvant~Chemotherapy	BOTTOM	517	Completed Neoadjuvant~Chemotherapy (N=517)			
2	8.03	Completed Neoadjuvant~Chemotherapy	BOTTOM	542	Completed Neoadjuvant~Chemotherapy (N=542)			

Automation Methods

Parent-Child Links Between Textboxes

Path	Phase	Node	Row	Row_link	Connect_forward	Connect_backward
1	1	1.01	1	1	3.01	1.01
1	1	3.01	3	1	4.02	1.01
1	2	4.02	4	3	6.02	3.01
1	2	6.02	6	4	8.02	4.02
1	2	8.02	8	6		8.02
2	1	1.01	1	1	3.01	1.01
2	1	3.01	3	1	4.03	1.01
2	2	4.03	4	3	6.03	3.01
2	2	6.03	6	4	8.03	4.03
2	2	8.03	8	6		6.03

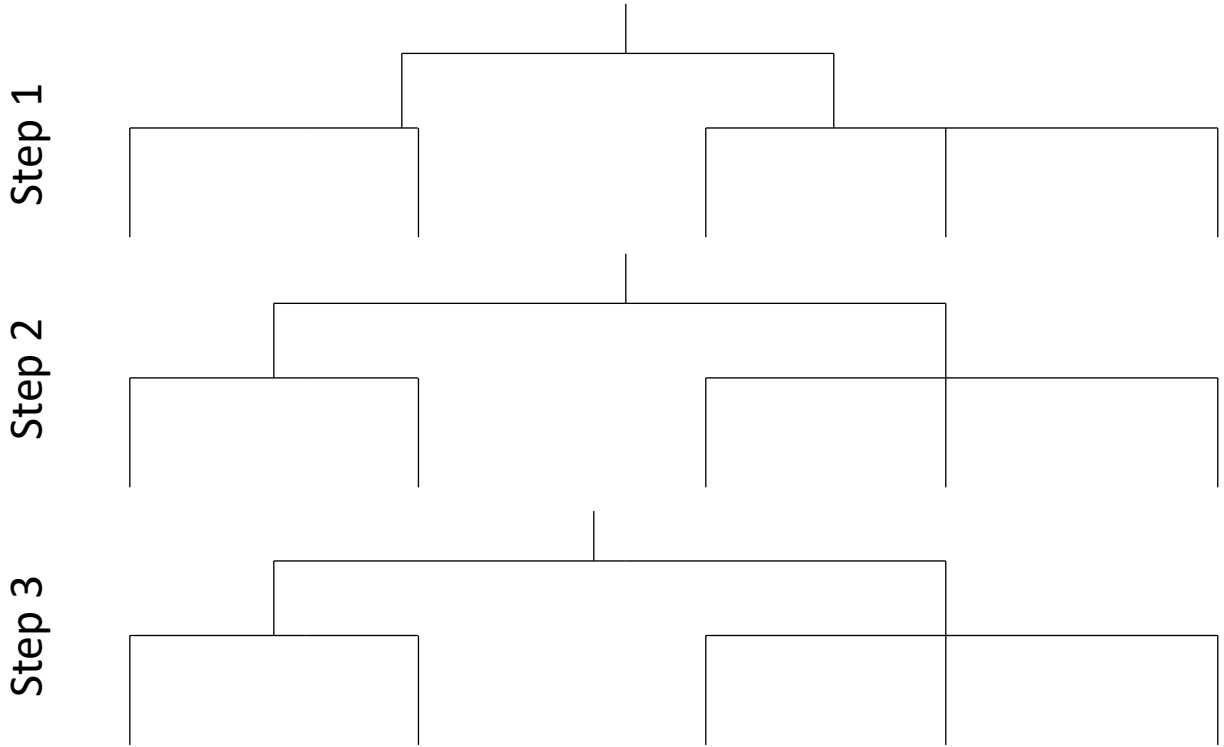
Automation Methods

Determine the X/Y Coordinates of Each Textbox

- X-coordinates are the difficult coordinate
  - Properly spacing and aligning different branching paths
- Y-coordinates are mostly straightforward with two different methods:
  - Allocate equal space to each row of the CONSORT and center the textboxes
  - Give each row of the CONSORT a proportionate space depending on how many lines of text exist and center the textboxes

Automation Methods

Determine the X-Coordinates



Final Steps

Final Steps

What is Left

- Collapse text values into one row
  - Concatenate with the split character to act as the delimiter
  - The TEXT plot will not combine text items in the same coordinates automatically, so must be done prior
- Output the x/y values for the connecting lines

Final Steps

Collapsing Text

```
Data _temp9;
  set _temp8;
  by y x;
     where ^missing(x) and ^missing(y);
     length _temp_text $10000.;
     if first.x then call missing(_temp_text);
     if ^(first.x and last.x) then do;
        _temp_text=catx('~',_temp_text,text);
        if last.x then do;
           text=_temp_text;
           output;
        end;
     end;
     else output;
     retain _temp_text;
Run;
```

phase	node	position	text	off_trt	n_off	off_order	connect_backward	connect_forward	x	y	column
1	1.01	BOTTOM	Registered (N=1500)				1.01	3.01	50.0	1	3
1	2.01	RIGHT	Screen Failure (N=144)~ -Ineligible (N=66)~ -Insurance Denied (N=78)	1	1	2	1.01	3.01	65.0	2	4
1	3.01	BOTTOM	Randomized (N=1356)				1.01	4.03	50.0	3	3
2	4.02	BOTTOM	Arm 1 (N=654)				3.01	6.02	25.0	4	2
2	4.03	BOTTOM	Arm 2 (N=702)				3.01	6.03	75.0	4	5
2	5.02	LEFT	Off-Treatment (N=37)~ -Withdrawal (N=10)~ -Progression (N=13)~ -Adverse Event (N=14)	1	1	3	4.02	6.02	17.5	5	1
2	5.03	RIGHT	Off-Treatment (N=51)~ -Withdrawal (N=20)~ -Progression (N=15)~ -Adverse Event (N=16)	1	1	3	4.03	6.03	82.5	5	6
2	6.02	BOTTOM	Started Treatment (N=617)				4.02	8.02	25.0	6	2
2	6.03	BOTTOM	Started Treatment (N=651)				4.03	8.03	75.0	6	5
2	7.02	LEFT	Off-Treatment (N=100)~ -Withdrawal (N=22)~ -Progression (N=16)~ -Adverse Event (N=25)~ -Death (N=18)~ -Alternate Therapy (N=19)	1	1	5	6.02	8.02	17.5	7	1
2	7.03	RIGHT	Off-Treatment (N=109)~ -Withdrawal (N=27)~ -Progression (N=25)~ -Adverse Event (N=17)~ -Death (N=14)~ -Alternate Therapy (N=26)	1	1	5	6.03	8.03	82.5	7	6
2	8.02	BOTTOM	Completed Neoadjuvant~Chemotherapy (N=517)				6.02		25.0	8	2
2	8.03	BOTTOM	Completed Neoadjuvant~Chemotherapy (N=542)				6.03		75.0	8	5

Final Steps

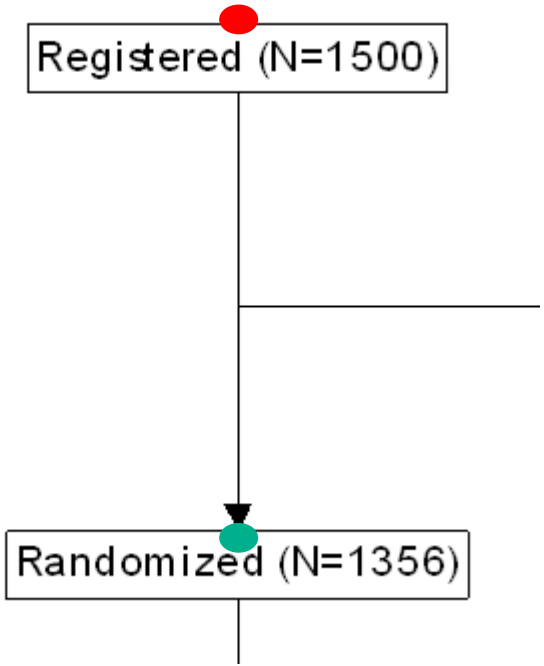
Calculating x/y values for connecting lines

- A one row data set containing all x/y coordinates and positions for each textbox is saved out
- This data set is then merged to every row of the data set on the previous slide
- This gives each row access to the attributes of each textbox for use in array functions
  - Using the row/column indexes previously calculated
  - The row/column indexes for forward/backward connecting nodes are also merged into the data set

Final Steps

Linking textboxes in the same column

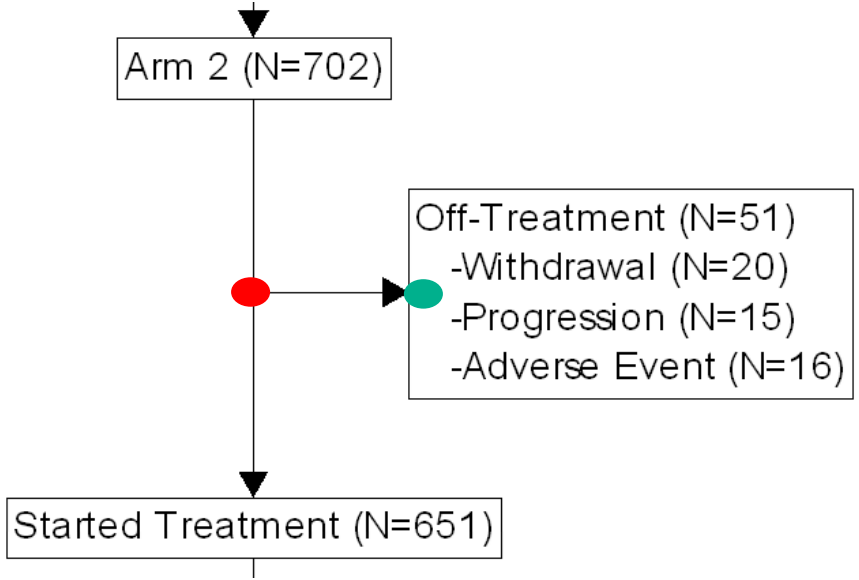
- Most straightforward
  - Origin: x/y of previous textbox
  - Finish: x/y of current textbox



Final Steps

Linking Off-Treatment Textboxes

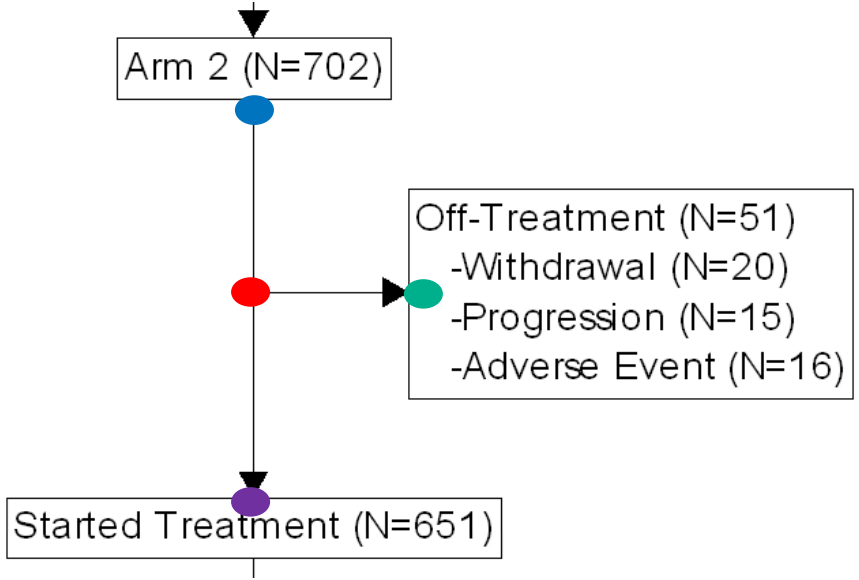
- Y-coordinates are tricky
  - Origin:
    - X-coordinate of previous textbox
    - Y-coordinate is midpoint of previous textbox and next textbox
  - Finish:
    - X-coordinate of current textbox
    - Y-coordinate is midpoint of previous textbox and next textbox



Final Steps

Linking Off-Treatment Textboxes

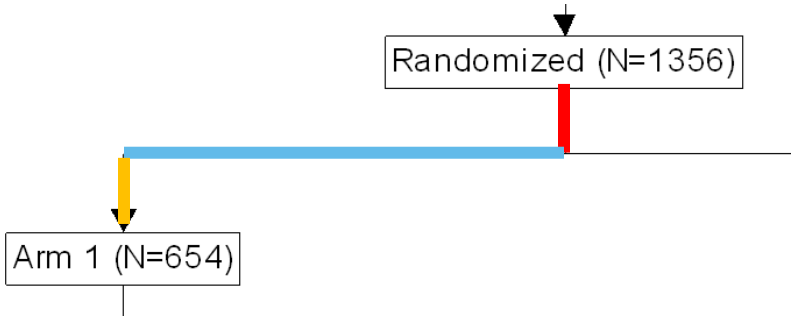
- Midpoint is calculated from *bottom* of previous textbox and the top of current
  - Top of the next is known
  - Bottom of previous must be approximated
    - Macro has an adjustable parameter (&MULTILINE_ADJUST) to give space for each line of text



Final Steps

Linking Textboxes not in the Same Column

- Three separate lines to calculate
  - Line coming down from previous textbox
  - Horizontal line between textboxes
  - Line connecting to current textbox



Final Steps

Snapshot of Line Coordinates

Phase	Node	Position	Label	X_b2	X_b	Y	X_r	X_l	X_line	Y_line	Id	X_line2	Y_line2
1	1.01	BOTTOM	Registered (N=1500)		50	6.2500							
1	2.01	RIGHT	Screen Failure (N=144)~ -Ineligible (N=66)~ -Insurance Denied (N=78)			19.7682	65.0						
1	2.01	RIGHT							50.0	19.7682	1002.0		
1	2.01	RIGHT							65.0	19.7682	1002.0		
1	3.01	BOTTOM	Randomized (N=1356)		50	31.2500							
1	3.01	BOTTOM							50.0	6.2500	1003.0		
1	3.01	BOTTOM							50.0	31.0500	1003.0		
2	4.03	BOTTOM	Arm 2 (N=702)		75	43.7500							
2	4.03	BOTTOM							75.0	38.5182	1004.0		
2	4.03	BOTTOM							75.0	43.5500	1004.0		
2	4.03	BOTTOM									1004.1	50	38.5182
2	4.03	BOTTOM									1004.1	75	38.5182
2	4.03	BOTTOM									1004.2	50	38.5182
2	4.03	BOTTOM									1004.2	50	31.2500

Final Steps

Superscript

- Superscripts only work in TEXT plots when they are applied as a numeric format to a variable
- CONSORT macro creates a format where each observation (START=_N_) is matched with the textbox label value
  - The variable START variable is plotted with the format attached

```
proc format cntlin=_consort (keep=start label fmtname
                                  where=(^missing(start)));
run;
```

Conclusion

- These methods can be used outside of a macro to programmatically create a CONSORT
  - There are many different preferences and styles of CONSORT depending on journal and employer
- The CONSORT macro is available to be downloaded and modified

Thank you!

Contact Information
Meyers.Jeffrey@mayo.edu
Jpmeyers.spa@gmail.com