PhUSE 2014

Paper SP08

Forest Plot: W3 + H2
WHAT is it, WHY use it, WHEN to use it
+ HOW to generate and HOW to interpret

Petra van Wetten, Novartis Pharma AG,
Basel, Switzerland

ABSTRACT

Contextualization, visualization and interpretation of complex biomedical data continue to be a challenge for many organizations and none more so than within the pharmaceutical industry. In February 2014, the European Medicines Agency (EMA) issued a draft guideline [1] advising on using forest plots as the graphical display for subgroup analysis. The visual representation provides a concise summary of results for easy interpretation.

Forest plots can be utilized to display the results of (but not limited to):

- meta-analysis
- subgroup analysis
- sensitivity analysis

This paper will present examples of forest plots focusing on meta-analysis and subgroup analysis. It will cover the following key objectives:

- Why use it
- When to use it
- What is it
- How to interpret
- How to generate

INTRODUCTION

In the author's experience, generating graphical outputs is often a daunting prospect for many statistical programmers. One useful step to counteract this is by understanding the 'bigger picture'. The subsequent sections will cover key areas in helping a programmer to understand and interpret forest plots.

WHAT, WHY and WHEN

A forest plot is a convenient and intuitively easily understood graphical display of a number of analyses of statistical parameters (e.g. means) and their confidence intervals. Forest plots came to be used increasingly frequently with the growth of meta-analysis associated with systematic review. In this context, typically, forest plots show the treatment effect size of each study and the results of the meta-analysis.

More generally, a forest plot presents the results of each analysis using a symbol (e.g. circle or square) to represent the point estimate and error bars representing the confidence interval. [2]

As stated in the abstract, the EMA recently drafted a guideline to present subgroup analyses by means of forest plots. On top of this, instead of potentially producing many more pages with summary tables for a subgroup analysis a forest plot gives a quick overview. It acts as a useful catalyst for timely and informed decision making on potential further analysis. Forest plots are typically used for meta-analysis, subgroup analysis and sensitivity analysis.

.

The questions asked in the above three types of analyses are different. Meta-analyses are used to combine the results of separate analyses (e.g. from different clinical trials). A subgroup analysis is used to either assess the homogeneity of a result across different patient subgroups to demonstrate that the result applies equally to all patient subgroups or to identify subgroups of patients in which the treatment effect differs from that of the main analysis (e.g. to define subgroups of patients who are particularly responsive to treatment). In a subgroup analysis, patients are typically

grouped according to common baseline characteristics (e.g. young patients vs old patients). A sensitivity analysis is done to check the robustness of the primary analysis to changes in the dataset, e.g. the impact of the removal of outliers, the use of a different analysis population or the influence of an alternative rule for missing data imputation.

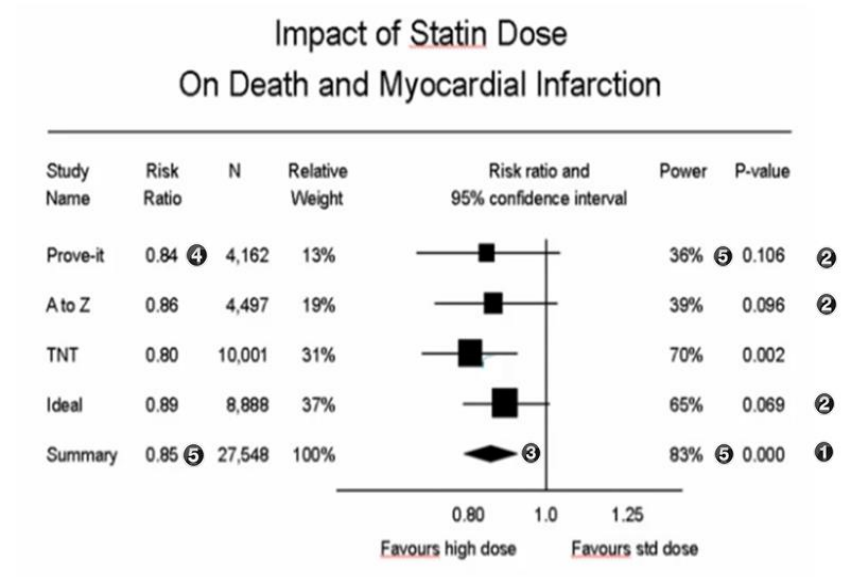
HOW to interpret

When it comes to interpretation, for programmers it may sometimes be hard to 'see the wood from the trees in the forest'. For the interpretation, this paper will only focus on the meta-analysis and the subgroup analysis. The results easily extend to sensitivity analyses.

Meta-Analysis

A forest plot used for a Meta-Analysis provides context across several studies and the overall result. One can see quickly in the graph, for example, whether the treatment effect is consistent across all studies. This is the case if the treatment effect across studies line up vertically. Even with a statistically significant p-value presented in a summary table one would not know whether this effect is consistent across the single studies supporting the overall analysis. Weight is assigned according to the precision of the estimate, based on the sample size, number of events, etc.
The box sizes offer clues about the level of precision of the results and with the reference line one can see at first sight whether study treatment or control is favored for each individual study and overall.

Figure 1 [3]

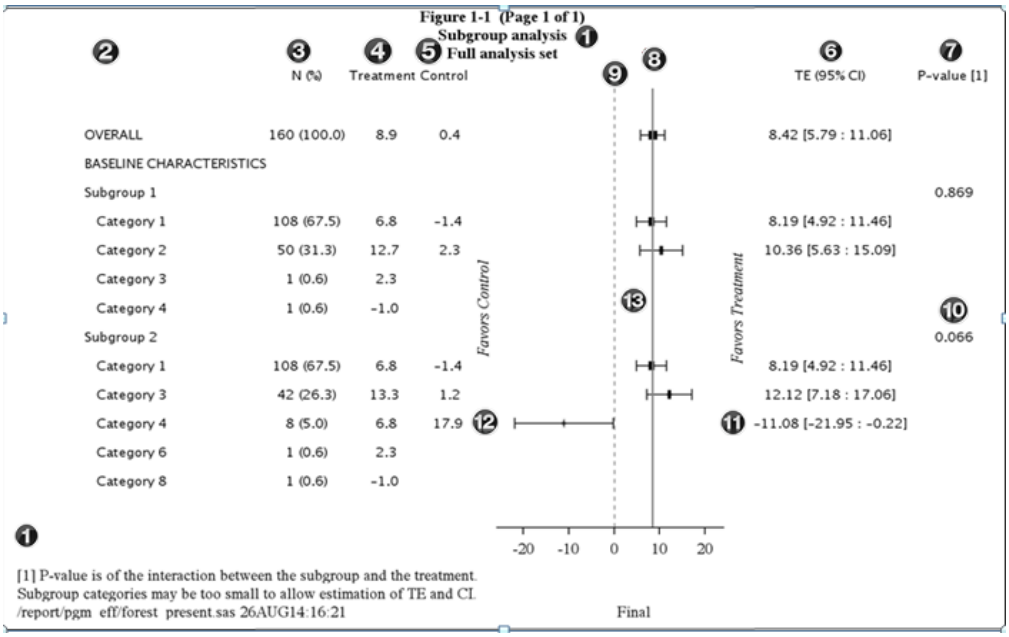


In the example of

Figure 1, the overall p-value ❶ for the combined results from all trials is statistically significant at the 5% level although three out of the four studies are not statistically significant ❷ at the 5% level. Since the studies are not wide spread across the graph and seem to line up the conclusion can be drawn that the single studies are supporting the overall effect ❸ and that for the three statistically insignificant studies the sample size might be too low (formally this corresponds to a non-significant p-value for the treatment-by-study interaction which can be interpreted as a test of heterogeneity). This is supported by the power where one can see that the power of all four studies combined is much higher than in each single study. E.g. the risk ratio ❹ for study Prove-it is 0.84 and for the summary it is 0.85. The difference in risk ratio is only 0.01 but when looking into the power ❺ the difference is quite impressive. For Prove-it the power is 36% where for the summery it is 83%, more than the double. So looking into effect sizes provides much more information than looking into p-values only. [3]

Subgroup Analysis

Figure 2



When doing a subgroup analysis, according to ICH E3, the subgroups need to be defined before treatment unblinding, and are usually based on baseline characteristics.
In the example of Figure 2, based on fictitious data, subgroups 1 and 2 represent different baseline characteristics whereas the categories are the values of the subgroup variable.

The total number of patients in each subgroup category ❸, the least square means of the primary variable for Treatment ❹ and Control ❺ arms respectively, the Treatment Effect with the 95% Confidence Interval ❻ and the p-value of the interaction between treatment and the baseline characteristic ❼ are presented on the graphical display ⓭ for additional information. There are two reference lines presented, one at 8.42, presenting the overall treatment effect ❽, and one at 0, indicating the "no difference"-line ❾ between treatment and control.
The p-values, for the interaction between treatment and the subgroup factor are to assess consistency of the treatment effect across the subgroup categories. A higher p-value and a confidence interval containing the overall treatment effect estimate is associated with greater consistency. The p-value for 'Subgroup 2' ❿ is close to the 5% level, usually used as a threshold, is caused by the result in category 4 ⓫ which is quite different from that in categories 1 and 3. On further investigation, the least square means in the treatment arm seem to be consistent with other subgroups whereas there seems to be a big difference in the control group for patients within that subgroup ⓬. A result like that might require further investigation for that specific subgroup.

In the example of figure 2 only two subgroups were displayed but this example already resulted in ten pages of summary tables. One can imagine the increased benefit in visual approach as more subgroups and categories are added.

HOW to generate

For the generation of the output we will focus on the subgroup analysis example in Figure 2 using two different approaches, namely SAS® Graphic Template Language (GTL) and SAS® ODS Graphic Designer (GD).

Graph Template Language (GTL): PROC TEMPLATE [4]

For Figure 2 above, PROC TEMPLATE has been used within a macro to define a template for the output. An input dataset needs to be prepared containing all required variables to be presented within the plot.
The reference numbers in the dataset header and in the code below are referring to the respective panels in the forest

plot in Figure 2 above.

Figure 3

VIEWTABLE: Rchwork.Result

	studyvalue	constant	cat	npct	nper	ntrt	trtv	nctr	ctrv	HRCL	diff	PVAL	pvalue	Estimate	overest	Lower	Upper	x1	x2
1	13	1	OVERALL	N (%)	160 (100.0)	Treat	8.9	Cntrl	0.4	TE (95% CI)	8.42 [5.79 : 11.06]	P-value		8.4233	8.4233	5.7911	11.0554	7.5072519	9.4510701897
2	12	1	BASELINE CHARACTERISTICS	N (%)		Treat		Cntrl		TE (95% CI)		P-value		.	8.4233	.	.	.	.
3	11	1	Subgroup 1	N (%)		Treat		Cntrl		TE (95% CI)		P-value	0.869	.	8.4233	.	.	.	.
4	10	1	Category 1	N (%)	108 (67.5)	Treat	6.8	Cntrl	-1.4	TE (95% CI)	8.19 [4.92 : 11.46]	P-value		8.1905	8.4233	4.9214	11.4596	7.578116438	8.8524049821
5	9	1	Category 2	N (%)	50 (31.3)	Treat	12.7	Cntrl	2.3	TE (95% CI)	10.36 [5.63 : 15.09]	P-value		10.3616	8.4233	5.6344	15.0888	9.995403654	10.741139014
6	8	1	Category 3	N (%)	1 (0.6)	Treat	2.3	Cntrl		TE (95% CI)		P-value		.	8.4233	.	.	0	0
7	7	1	Category 4	N (%)	1 (0.6)	Treat	-1.0	Cntrl		TE (95% CI)		P-value		.	8.4233	.	.	0	0
8	6	1	Subgroup 2	N (%)		Treat		Cntrl		TE (95% CI)		P-value	0.066	.	8.4233	.	.	.	.
9	5	1	Category 1	N (%)	108 (67.5)	Treat	6.8	Cntrl	-1.4	TE (95% CI)	8.19 [4.92 : 11.46]	P-value		8.1905	8.4233	4.9214	11.4596	7.578116438	8.8524049821
10	4	1	Category 3	N (%)	42 (26.3)	Treat	13.3	Cntrl	1.2	TE (95% CI)	12.12 [7.18 : 17.06]	P-value		12.1190	8.4233	7.1822	17.0558	11.75820092	12.490817897
11	3	1	Category 4	N (%)	8 (5.0)	Treat	6.8	Cntrl	17.9	TE (95% CI)	-11.08 [-21.95 : -0.22]	P-value		-11.0833	8.4233	-21.9510	-0.2157	-11.0197158	-11.14731811
12	2	1	Category 6	N (%)	1 (0.6)	Treat	2.3	Cntrl		TE (95% CI)		P-value		.	8.4233	.	.	0	0
13	1	1	Category 8	N (%)	1 (0.6)	Treat	-1.0	Cntrl		TE (95% CI)		P-value		.	8.4233	.	.	0	0

The forest plot, like shown in Figure 2, is based on the input dataset shown in Figure 3 above. In order to create the plot following code in GTL has been developed.

```
proc template;
   define statgraph simpleforest;
 ❶ dynamic _title0 _title1 _title2 _title3 _title4 _title5 _title6 _title7 _title8 _title9
         _title10
         _footnote1 _footnote2 _footnote3 _footnote4 _footnote5 _footnote6 _footnote7
         _footnote8 _footnote9 _footnote10;

   begingraph / designheight=610px designwidth=900px;

    entrytitle halign=left textattrs=(size=10pt) _title0;
    entrytitle halign=left textattrs=(size=10pt) _title1;
    entrytitle halign=center textattrs=(size=10pt) _title2;
    entrytitle halign=center textattrs=(size=10pt) _title3;
    entrytitle halign=center textattrs=(size=10pt) _title4;
    entrytitle halign=center textattrs=(size=10pt) _title5;
    entrytitle halign=center textattrs=(size=10pt) _title6;
    entrytitle halign=center textattrs=(size=10pt) _title7;
    entrytitle halign=center textattrs=(size=10pt) _title8;
    entrytitle halign=center textattrs=(size=10pt) _title9;
    entrytitle halign=center textattrs=(size=10pt) _title10;

    entryfootnote halign=left textattrs=(size=10pt) _footnote1;
    entryfootnote halign=left textattrs=(size=10pt) _footnote2;
    entryfootnote halign=left textattrs=(size=10pt) _footnote3;
    entryfootnote halign=left textattrs=(size=10pt) _footnote4;
    entryfootnote halign=left textattrs=(size=10pt) _footnote5;
    entryfootnote halign=left textattrs=(size=10pt) _footnote6;
    entryfootnote halign=left textattrs=(size=10pt) _footnote7;
    entryfootnote halign=left textattrs=(size=10pt) _footnote8;
    entryfootnote halign=left textattrs=(size=10pt) _footnote9;
    entryfootnote halign=left textattrs=(size=10pt) _footnote10;

    layout lattice / columns=7 columngutter=0 columnweights=(.26 .08 .07 .06 .28  .17 .08 );

      ❷ * Labels to be displayed in the left panel of the output
      ********************************************************************************;
      layout overlay / walldisplay=none
                       xaxisopts=(display=none offsetmin=0 offsetmax=0)
                       y2axisopts=(display=(tickvalues) offsetmin=0.03 offsetmax=0.03
                                   tickvalueattrs=(size=8pt Family="Courier New")
                                   linearopts=(tickvaluesequence=(start=1 end=13 increment=1)
                                   viewmin=0 viewmax=14));
                       entry halign=left "" / location=outside valign=top
                             textattrs=(size=8pt Family="Courier New" ) ;
                       scatterplot y=studyvalue x=constant / yaxis=y2 markerattrs=(size=0);
      endlayout;
```

```
      ❸ * N (%) / Treatment estimate / Control estimate / TE (95%CI) / P-value
      ********************************************************************************;
      layout overlay / walldisplay=none border=false
                       xaxisopts=(display=none offsetmin=0 offsetmax=0)
                       yaxisopts=(display=none linearopts=(tickvaluesequence=(start=1
                                  end=13 increment=1) viewmin=0 viewmax=14));
                       entry "N (%)" / location=outside valign=top textattrs=(size=8pt
                                       Family="Courier New" );
                       scatterplot y=studyvalue x=npct / markercharacter=nper
                                   markercharacterattrs=(size=8pt Family="Courier New");
       endlayout;

       ⓭ * Forest Plot
       ********************************************************************************;
       layout overlay / walldisplay=none
                        xaxisopts=(display=(line ticks tickvalues) /*linearopts=(viewmin=-40
                        viewmax=30 )*/)
                        x2axisopts=(display=none)
                        yaxisopts=(display=(label) label="Favors Control"
                                   labelattrs=(size=GraphValueText:FontSize style=italic)
                                   linearopts=(tickvaluesequence=(start=1 end=13 increment=1)
                                  viewmin=0 viewmax=14))
                        y2axisopts=(display=(label) label="Favors Treatment"
                                    labelattrs=(size=GraphValueText:FontSize style=italic)
                        linearopts=(tickvaluesequence=(start=1 end=13 increment=1)
                                    viewmin=0 viewmax=14));
                        scatterplot x=estimate y=studyvalue / xerrorlower=lower
    xerrorupper=upper
                                    markerattrs=(symbol=plus);
                        scatterplot x=estimate y=studyvalue / markerattrs=(symbol=plus size=0)
                                    xaxis=x2 yaxis=y2;;
                        vectorplot x=x2 y=studyvalue xorigin=x1 yorigin=studyvalue /
                                   lineattrs=graphdata1(color=black thickness=8)
                        arrowheads=false;
                        referenceline x=0 /  lineattrs=(pattern=shortdash) ;   ❾
                        referenceline x=overest /  lineattrs=(pattern=solid);  ❽
         endlayout;
       endlayout;
     endgraph;
   end;
run;
```

Seven panels within the object were defined, one panel for the subgroup labels, five different panels presenting numbers and statistics and one panel for the forest plot itself.

❶ titles and footnotes are defined as dynamic variables to have the titles and footnotes passed dynamically at run-time. The entrytitle and entryfootnote statements are used in order to embed the titles and footnotes within the plot area.

❷ definition of subgroup labels and indentation .
A format is assigned to variable studyvalue, which is passes as the y2 axis variable, in order to indent the labels in the dataset.

❸ display of numbers and statistics
Repeat this step for the panel ❹, ❺, ❻ and ❼ and replace the panel headers, variable x and the Markercharacter

Panel header: “Treatment”/ x=ntrt / markercharacter=trtv ❹
Panel header: “Control” / x=nctr / markercharacter=ctrv ❺
Panel header: “Treatment estimate (95% CI)” / x=hrcl / markercharacter=diff ❻
Panel header: “p-Value” / x=pval / markercharacter=pvalue ❼

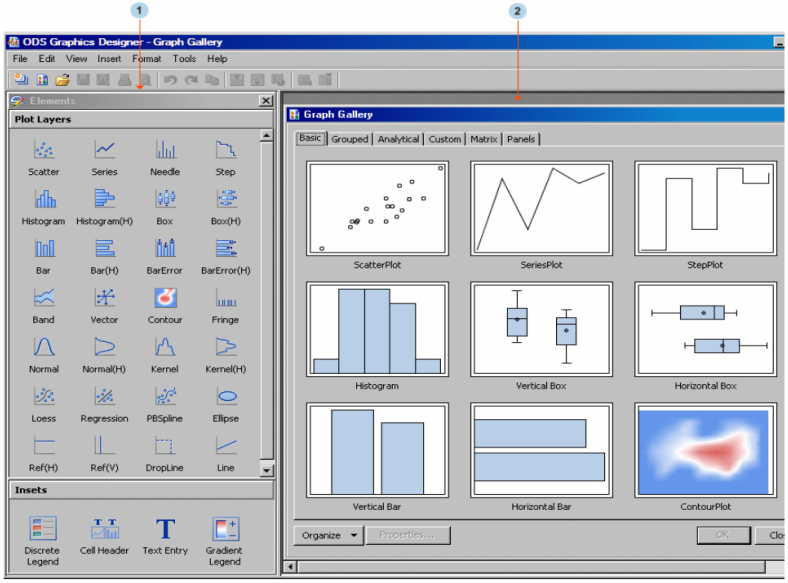
The variables passed as the x-axis variable contain a constant value for all observations for each of the variables (e.g. npct will contain value N(%) for all observations). The actual values to be displayed on the plot are included in the variables assigned to the markercharacter option (e.g. nper).

⓭ display of the graph: 2 scatterplots and one vector plot
a box is drawn by defining a thick line (thickness=8) between variable x1 and x2 within the vector plot reflecting the number of patients within a specific subgroup category

ODS Graphics Designer (GD) [5]

For programmers unfamiliar with GTL and PROC TEMPLATE the ODS, Graphics Designer is a useful tool to set up a

simple forest plots and to modify the code retrieved afterwards.



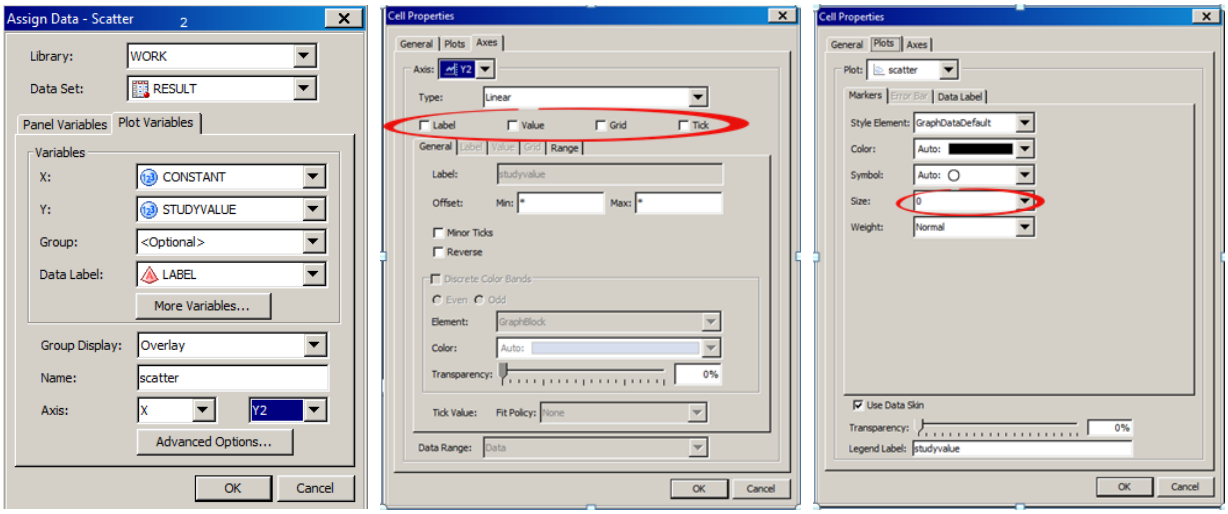
[5]

1 **Elements pane**
Contains plots, lines, and insets that you can insert into a graph. To insert an element, click and drag the element to the graph.

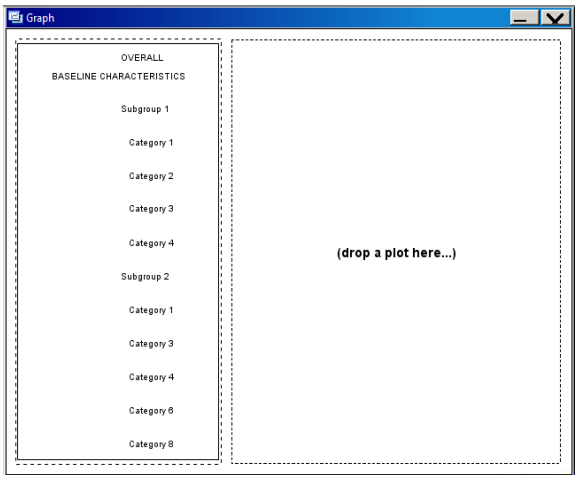
The elements on this pane are enabled only when a graph is open.

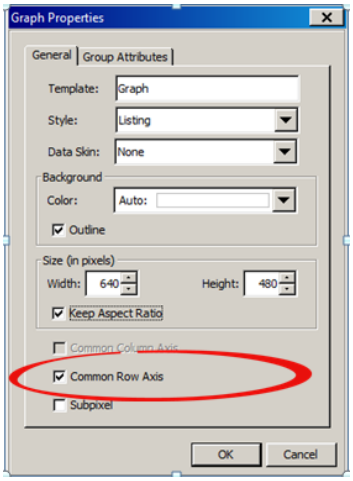
2 **Work area**
Contains one or more graphs that you can design. In addition to the graphs, you can display the Graph Gallery, a collection of predefined graphs that are supplied with the software. You can design a graph from the Graph Gallery by double-clicking one of the graph icons and then making your changes. (Alternatively, you can design your graph from a blank graph without using the Graph Gallery.)

To create a simple forest plot with only two panels, select a new graph (file → new → blank graph) and click on SCATTER in the elements pane, and include the scatterplot by drag and drop into the blank working area. Select the variables for x and y axis, and select the y2 axis in order to keep the indenting of the subgroup and category labels and modify the plot properties. Plot properties can either be accessed by the main menu (→ Format → Plot properties …) or by right clicking in the graph. Deselect Label, Value, Grid and Tick for both the x- and y2-axis.

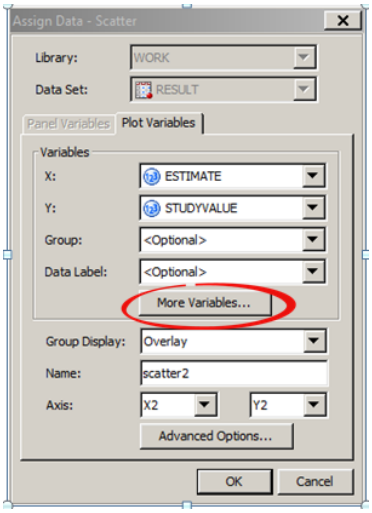


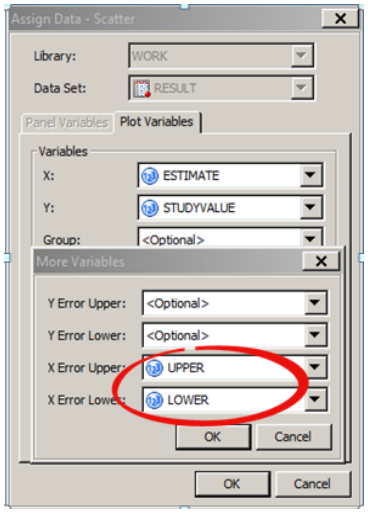
Insert a column either by the main menu (Insert → column) or by right clicking in the graph and choosing ‘add column’ from the drop down menu. Once the new column is added, select Graph Properties either from the main menu (Format → Graph Properties …) or by right clicking in the graph and select ‘common row axis’.

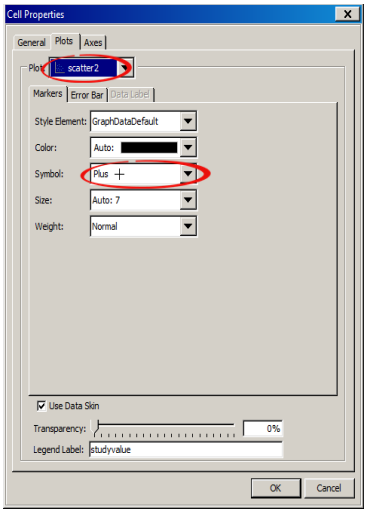




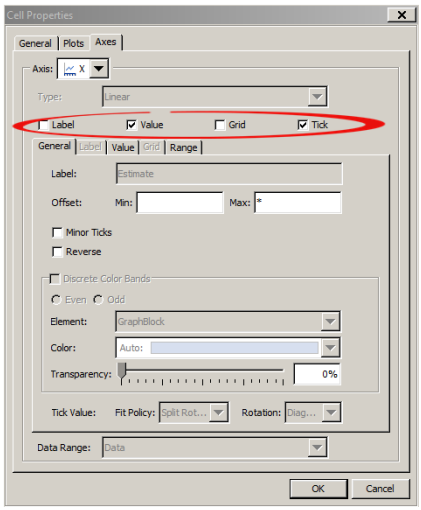
The forest plot itself is created by adding a scatter plot and a vector plot in the new working area.
Drag the scatter plot from the element pane and drop it into the new column working area. In order to create the confidence intervals, more variables need to be selected and defined as the x Error Upper and Lower. In this case variable UPPER and LOWER are selected. Change the marker (right click →Plot Properties … → select scatter 2 → change symbol to plus).

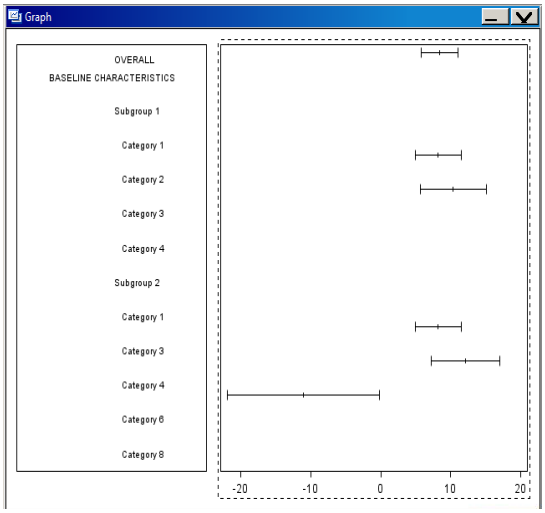




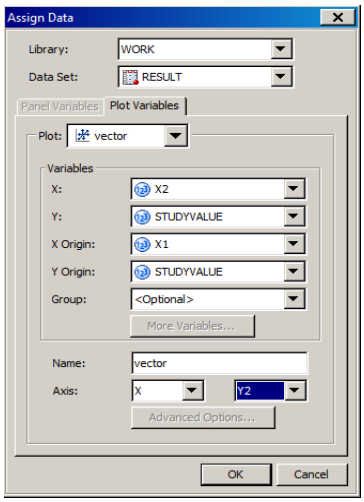


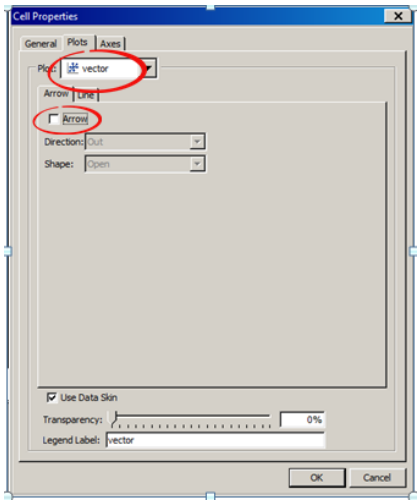
When defining the axis properties deselect Label, Value, Grid and Tick for the y-axis, but this time the 'tick and value' need to be selected for the x-axis.

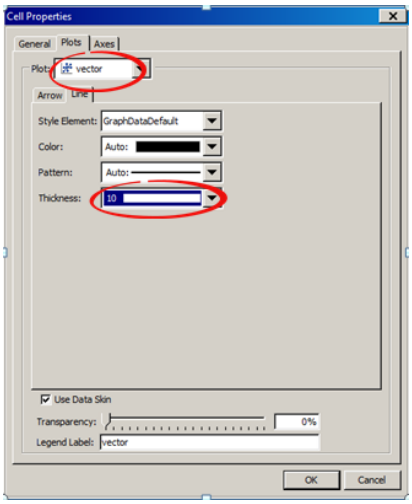




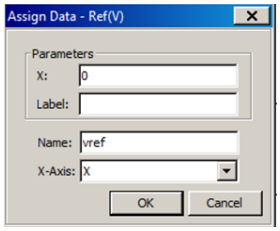
By dragging the vector plot from the element pane into the new column area the box is drawn. The size of the box is defined by the variables x1 and x2 and the line thickness. X1 and x2 are assigned to x and x origin, y2-axis needs to be selected in order to assign different labels later on. Modify the plot properties by un-ticking Arrow and by choosing a thick line. Whenever multiple plots are included to create a graph, carefully select the plot where you want to modify the properties.

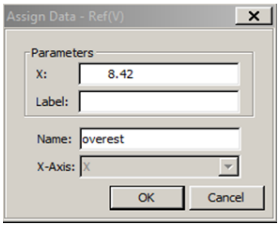


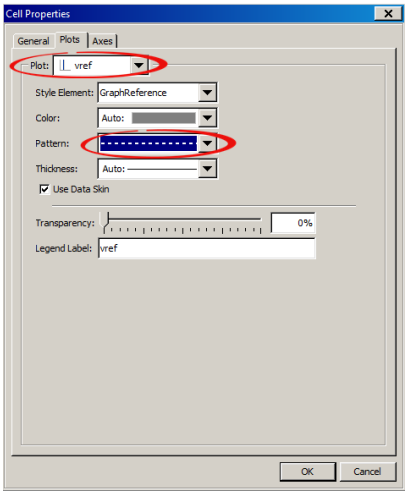


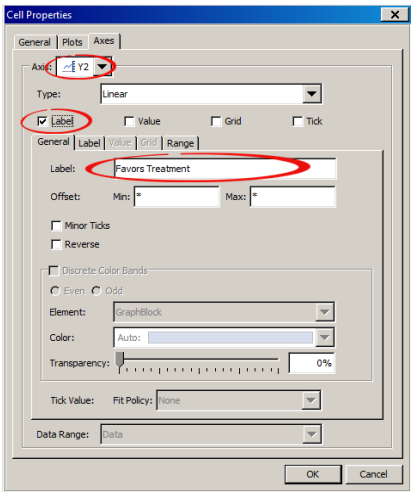


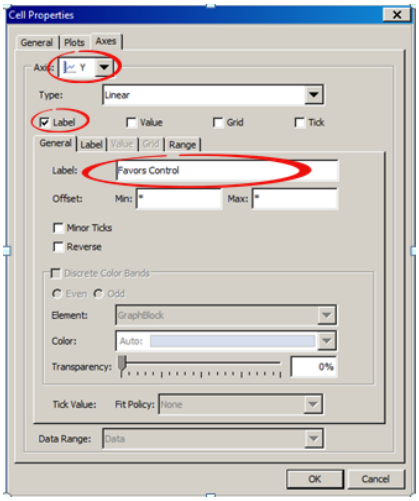
Two reference lines will be drawn by dragging and dropping Ref(v) from the elements panel into the working area. For the first reference line the value x is set to 0 and for the second x is 8.42 and name is set to overest. The pattern for the reference line can be modified within the plot properties as well. Finally a label for y and y2 is assigned indicating which side favors Control and which side favors Treatment.



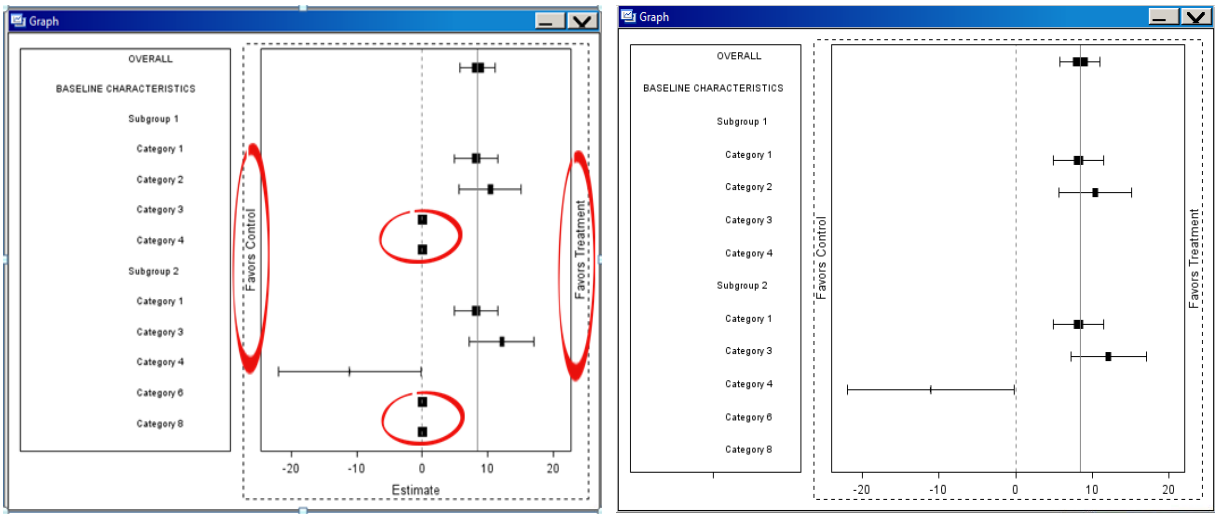








In the graph below on the left the box size is not reflecting the actual number of patients in some of the categories where the confidence interval is missing. In the example of figure 2 this is the case if n=1. By looking into that graph one gets the impression that the population in the respective category is actually quite big and the confidence interval could be very narrow, what is a completely wrong presentation. In order to display the size boxes only when the confidence interval is not missing, variables x1 and x2 need to be set to missing. This example is showing that it is still very important to prepare your dataset thoroughly even when creating a plot with drag & drop.



The code of the final graph can be retrieved and modified afterwards by going to the main menu (view → Code).
For the subgroup example above the panels containing numbers and statistics would need to be included in the retrieved code.

Code - Graph

```
proc template;
define statgraph Graph;
dynamic _CONSTANT _STUDYVALUE _LABEL _ESTIMATE _STUDYVALUE2 _LOWER _UPPER _X2A _STUDYVALUE3 _X1A
_STUDYVALUE4;
begingraph;
  layout lattice / rowdatarange=data row2datarange=data columndatarange=data columns=2 rowgutter=10 columngutter=10
columnweights=(0.6611842105263158 1.34);
    layout overlay / xaxisopts=( display=(LINE )) y2axisopts=( display=(LINE ) griddisplay=off);
      scatterplot x=_CONSTANT y=_STUDYVALUE / datalabel=_LABEL name='scatter' yaxis=Y2 markerattrs=(size=0 );
    endlayout;
    layout overlay / xaxisopts=( display=(TICKS TICKVALUES LINE )) yaxisopts=( display=(LINE LABEL ) label=('Favors Control'))
y2axisopts=( display=(LINE LABEL ) label=('Favors Treatment'));
      scatterplot x=_ESTIMATE y=_STUDYVALUE2 / xerrorupper=_LOWER xerrorlower=_UPPER name='scatter2' clusterwidth=0.5
markerattrs=(symbol=PLUS );
      vectorplot x=_X2A y=_STUDYVALUE3 xorigin=_X1A yorigin=_STUDYVALUE4 / name='vector' yaxis=Y2 arrowheads=false
lineattrs=(thickness=10 );
      referenceline x=0.0 / name='vref' xaxis=X curvelabelposition=max lineattrs=(pattern=SHORTDASH );
      referenceline x=8.42 / name='overest' xaxis=X;
    endlayout;
  endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.RESULT template=Graph;
dynamic _CONSTANT="CONSTANT" _STUDYVALUE="STUDYVALUE" _LABEL="LABEL" _ESTIMATE="ESTIMATE"
_STUDYVALUE2="STUDYVALUE" _LOWER="LOWER" _UPPER="UPPER" _X2A="X2" _STUDYVALUE3="STUDYVALUE"
_X1A="X1" _STUDYVALUE4="STUDYVALUE";
run;
```

CONCLUSION

A forest plot is a very flexible graph. It's very efficient and displays all important information usually displayed in a meta-analysis or a subgroup/sensitivity analysis. A key benefit with a forest plot is that it facilitates an easier review as multiple analyses are displayed visually compared with an equivalent tabular format over many more pages. The old adage 'a picture paints a thousand words' holds true in this regard.

There are different methods to create a forest plot and programmers should explore the method they are most comfortable with. It is important for programmers to understand the bigger picture in terms of the what, why, when and how before undertaking the generation of an output.

REFERENCES

[1] http://www.ema.europa.eu/docs/en_GB/document_library/Scientific_guideline/2014/02/WC500160523.pdf

[2] https://www.statstodo.com/ForestPlot_Exp.php

[3] Meta-Analysis of cardiovascular outcomes trials, Comparing intensive versus moderate statin therapy
Christopher P. Cannon, Benjamin A. Steinberg, Sabina A. Murphy, Jessica L. Mega, Eugene Braunwald
Michael Borenstein, www.Meta-Analysis.com,
lecturing session on YouTube https://www.youtube.com/watch?v=xWiXeKR3dB4)

[4] SAS Global Forum 2010 (support.sas.com/resources/papers/proceedings10/195-2010.pdf)
Creating Forest Plots from Pre-computed Data using PROC SGPLOT and Graph Template Language

[5] SAS/GRAPH(R) 9.2: ODS Graphics Designer Help:
support.sas.com/documentation/cdl/en/grstatdesign/61690/HTML/default/viewer.htm#p0swzt49p7k30rn1jnztg97nuicp.htm)

ACKNOWLEDGMENTS

I'd like to thank all my colleagues, who reviewed, for their most valuable comments.
Special thanks to Jimmy Fernandes, who developed the forest plot macro shown in this paper, for his support throughout, to Harry Staines, for sharing his knowledge on how to interpret forest plots and to James Gallagher, for his help, support and encouragement.

CONTACT INFORMATION

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

Petra van Wetten
WSJ 210.7.13
Novartis Pharma AG
Postfach
CH-4002 Basel
SWITZERLAND
Email: petra.van_wetten@novartis.com

Brand and product names are trademarks of their respective companies.