

RECISTance is Useless

PHUSE EU Connect: November 2025



The Title

A Novel Visualisation...

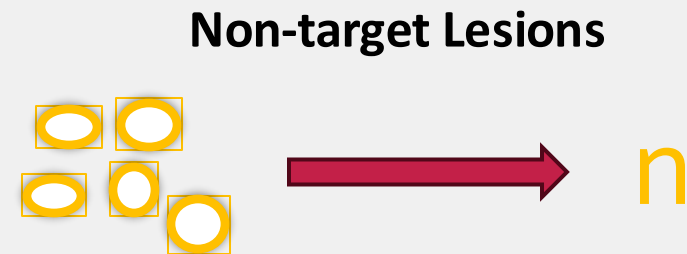
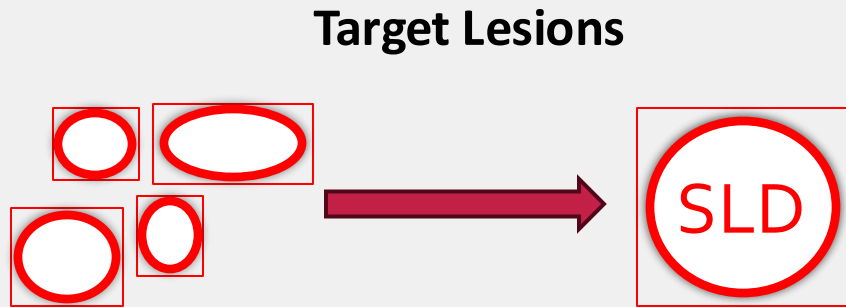


The Hitchiker's Guide to the Galaxy (Adams, 1978)¹ has some sound advice on the front cover, simply the words:

DON'T PANIC!

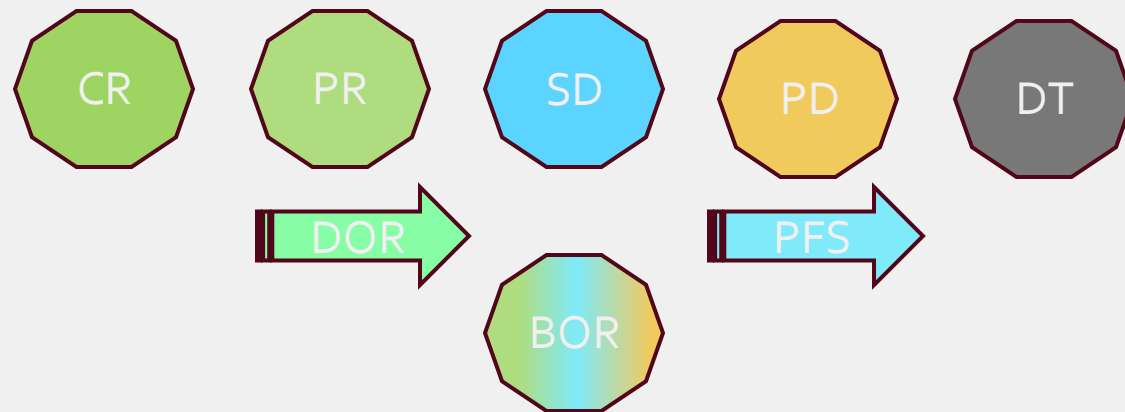
The concept of RECIST can seem a daunting one, especially for programmers that are tackling their first cancer studies. This concise tour and summary should be an encouragement to any that might be floundering.

RECIST: in 1 minute



SLD¹ (mm) & lesion(n) contribute to the categorization of overall disease status

Measures of clinical status



Abr.	Description	Fulfillment
CR	Complete Response	Disappearance of all target and non-target lesions (¹ SLD=0mm)
PR	Partial Response	Significant reduction in tumour size (SLD reduction >=30%)
SD	Stable Disease	No significant change in SLD or lesion counts (not CR, PR, PD)
PD	Progressive Disease	Increase in SLD (>=20% + absolute increase >=5mm from ² nadir) or any additional lesions
DT	Death	x—————NA—————x

Abr.	Description	Fulfillment
DOR	Duration of Response	Time since PR or CR, without PD
PFS	Progression Free Survival	Time since start of study without PD
DSD	Duration of Stable Disease	Time from first result of SD, PR or CR without PD
BOR	Best Overall Response	The best response during the study (Hierarchy: CR > PR > SD > PD)

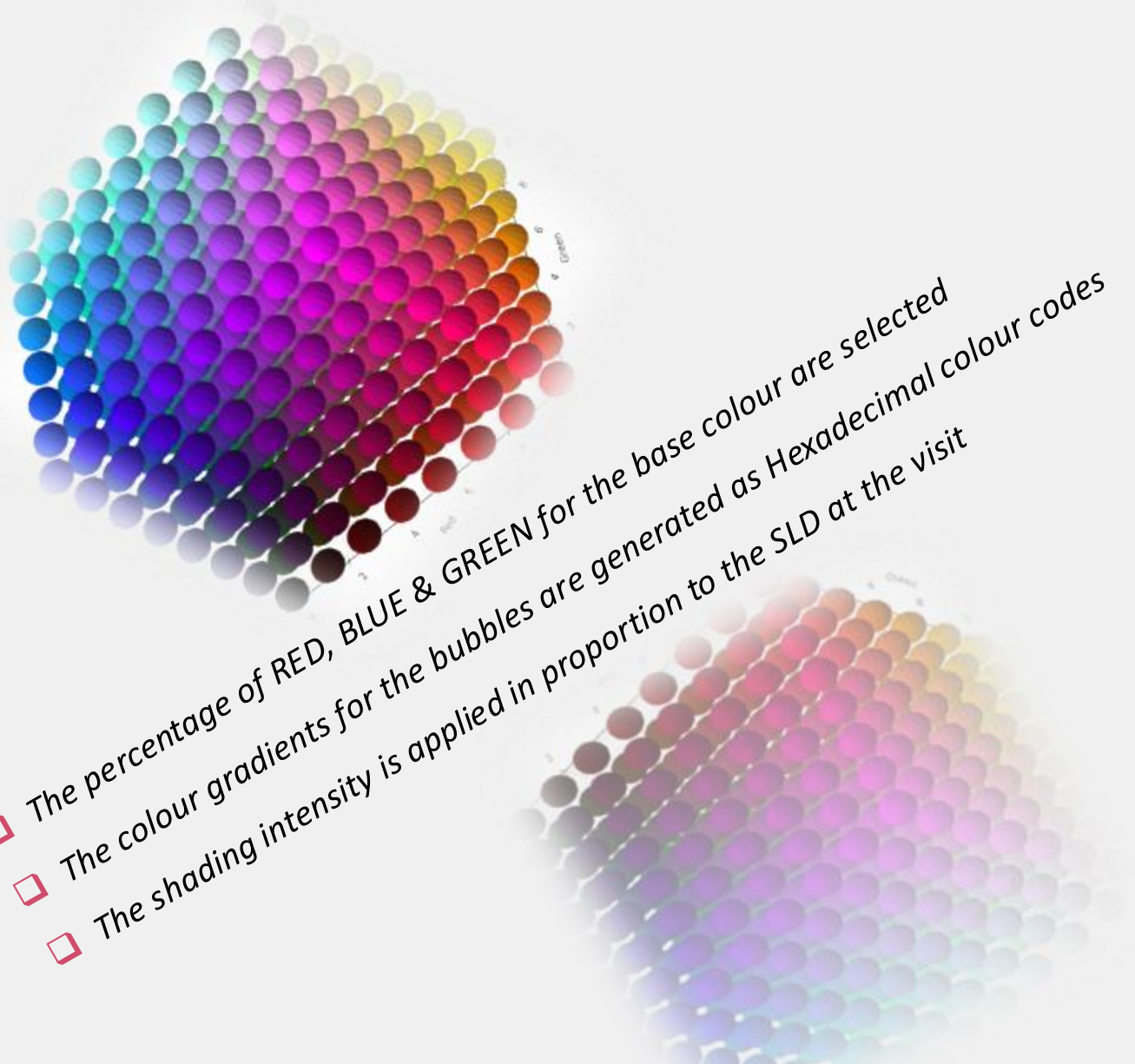
Seed Ideas

- ☐ Present by patient plots with disease status at each visit
- ☐ Use circles to represent SLD (target lesions) at each patient/visit
- ☐ Capture overall patient status for the study (BOR, DOR, PFS, DSD)
- ☐ Use colour shading to represent SLD at each patient/visit
- ☐ Include counts of target and non-target lesions at each visit
- ☐ Present the overall study results on a single page – Future addition

Putting into Practice

- ☐ Generate realistic test data, covering many patient statuses/outcomes
- ☐ A bubble plot as a sensible starting point for the SLD data
- ☐ Plot Patient versus Visit with SLD controlling the bubble diameters
- ☐ Overlay a scatterplot that shows the disease status at each visit
- ☐ Include control over the colour scheme : colour proportions and shading
- ☐ Include a summary section within the plot area, for each patient
- ☐ Utilise macro language for flexibility and ease of use

Colour Control: RGB elements

- 
- ❑ The percentage of RED, BLUE & GREEN for the base colour are selected
 - ❑ The colour gradients for the bubbles are generated as Hexadecimal colour codes
 - ❑ The shading intensity is applied in proportion to the SLD at the visit

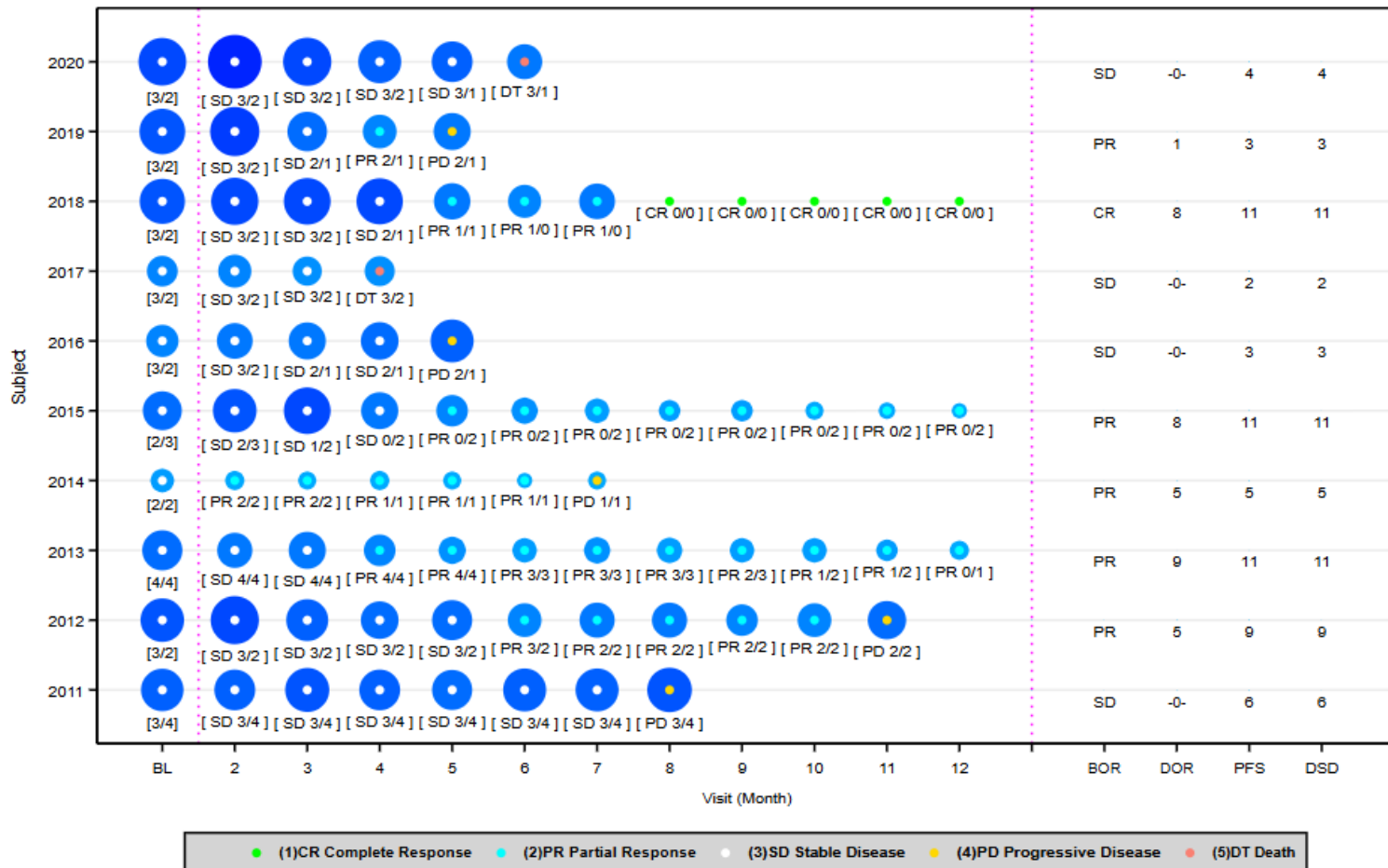
RECISTance is Useless

8

Elements of the Graph

Tumour Burden Evaluation by patient : Using RECIST 1.1

Treatment Group : 2 mg Active-drug



Bubble sub-text : [Disease Status at visit & n (target) / n (non-target)]

BOR -> Best Overall Response DOR -> Duration of Response PFS -> Progression Free Survival DSD -> Duration of SD

disease status (colour)

diameter proportional to SLD



disease status (text)

non-target lesion(n)

target lesion(n)

Test Data Preparation

Total rows: 510 Total columns: 11

	treatm	subject	visit	target	ntarget	stat	sld_p	status
1	1	1001	BL	2	2		138	[2/2]
2	1	1001	2	1	2	SD	159	[SD 1/2]
3	1	1001	3	1	2	PD	201	[PD 1/2]
4	1	1001	4	1	1			[1/1]
5	1	1001	5	1	1			[1/1]
6	1	1001	6	1	1			[1/1]
7	1	1001	7	1	1			[1/1]
8	1	1001	8	1	1			[1/1]
9	1	1001	9	1	1			[1/1]
10	1	1001	10	1	1			[1/1]
11	1	1001	11	1	1			[1/1]
12	1	1001	12	1	1			[1/1]
13	1	1001					0.01	
14	1	1001	BOR				0.01	SD
15	1	1001	DOR				0.01	-0-
16	1	1001	PFS				0.01	1
17	1	1001	DSD				0.01	1
18	1	1002	BL	4	3		116	[4/3]
19	1	1002	2	4	3	SD	116	[SD 4/3]
20	1	1002	3	3	2	SD	85	[SD 3/2]
21	1	1002	4	3	2	DT	74	[DT 3/2]
22	1	1002	5					
23	1	1002	6					
24	1	1002	7					
25	1	1002	8					
26	1	1002	9					
27	1	1002	10					
28	1	1002	11					
29	1	1002	12					
30	1	1002					0.01	
31	1	1002	BOR				0.01	SD
32	1	1002	DOR				0.01	-0-
33	1	1002	PFS				0.01	2
34	1	1002	DSD				0.01	2

Visits for x-axis

Nominal small SLD values to maintain the plotting points for disease status

Adjusted SLD determines bubble diameter (non-summary data)

Insert a gap to the X axis, between visit and summary data.

Bubble labels: summary info

Disease Status (per visit)

Code Snippets: Scaling SLD values

```
* identify the min/max SLD values from all subjects - assign to macro variables;
proc sql noprint;
  select min(sld), max(sld) into: minsld, :maxsld
  from recist;
quit;

* factor to scale SLD bubble sizes - the smaller the supplied number
- the greater the value of plotted SLD2;
%let scale=500; * Can be tweaked by the user dependant on data values;

data recist3(drop=rsld updown);
  set recist2;
  * Scale the SLD for plotting purposes - based on the range of SLD values;
  rsld=&maxsld - &minsld;
  factor=rsld/&scale; * scaled SLD;
  sld2=round(sld/factor, 1);
run;
```

Code Snippets: Generating colour hexadecimal codes and derive block size (range of SLD/15)

```
* macro to convert proportions of red, green and blue (out of 255 max)
* to a hexadecimal SAS colour code;
* this uses the SAS supplied HEX2 format;
* RR: proportion of RED in the mixture (x/255);
* GG: proportion of GREEN in the mixture (x/255);
* BB: proportion of BLUE in the mixture (x/255);
* Concatenate all 3 components and create a "CX----" colour code;
%macro rgbhex(rr, gg, bb);
  %sysfunc(compress(cx%sysfunc(putn(&rr, hex2.))
    %sysfunc(putn(&gg, hex2.))
    %sysfunc(putn(&bb, hex2.))));
%mend rgbhex;
```

```
* macro to generate colour gradient for the SLD / bubble sizes;
* BASECOL: base colour;
* SECOND: second colour to mix with base;
* VARCOL: colour to be used as a mixer in order to vary the tone/shade;
* TREATM: treatment group to include;
%macro colgen(basecol=, second=, varcol=, treatm=);
  * identify the "block size" for the entire SLD range - if divided into 15 parts;
  data _null_;
    range2=(&maxsld2 - &minsld2)/15;
    call symput('block', compress(put(range2,8.)));
  run;
  %put Block Size=&block;
```

Code Snippets: Assigning colour mixtures and densities to macro symbols (15 shades of the mixed colour)

```
* initialize colours each to zero;
```

```
%let pos1=0;
```

```
%let pos2=0;
```

```
%let pos3=0;
```

```
* assign values for proportions of colours;
```

```
%if      &basecol=red      %then %let pos1=230;
```

```
%if      &basecol=green    %then %let pos2=230;
```

```
%if      &basecol=blue     %then %let pos3=220;
```

Degree of saturation (255 max)

```
%if      &second=red       %then %let pos1=230;
```

```
%if      &second=green     %then %let pos2=230;
```

```
%if      &second=blue      %then %let pos3=230;
```

Base colours (&basecol and &second) are given a fixed saturation and the variable colour has a saturation that is based on the loop iteration number

```
%if      &varcol=red       %then %let pos1=%nrstr(&x*(12));
```

```
%else %if &varcol=green    %then %let pos2=%nrstr(&x*(12));
```

```
%else %if &varcol=blue     %then %let pos3=%nrstr(&x*(12));
```

```
* generate X colour shades - based on number of colour group values specified;
```

```
data _null_;
```

```
  %do x = 1 %to 15;
```

```
    * use the macro to mix the colours in correct proportions;
```

```
    tvar="%rgbhex(&pos1, &pos2, &pos3)";
```

```
    s1="&pos1"; s2="&pos2"; s3="%unquote(&pos3)";
```

```
    call symput("newcolr"!!compress(put(&x,8.)), compress(tvar));
```

```
    output;
```

```
  %end;
```

```
run;
```

Code Snippets: Using Graphics Template Language (GTL) - build an attribute map using the previously generated hexadecimal codes. This associates 15 ranges of SLD with 15 shades of the mixed colour.

```
proc template;
  define statgraph recistplot;
    begingraph / border=false;
      * associate colours with SLD ranges - for bubbles;
      rangeattrmap name="recmap" ;
        range missing / rangealtcolor=white rangecolor=white;
        range 0 - %eval((1*&block)-1) / rangecolor=&newcolr15;
        range %eval(1*&block) - %eval((2*&block)-1) / rangecolor=&newcolr14;
        range %eval(2*&block) - %eval((3*&block)-1) / rangecolor=&newcolr13;
        range %eval(3*&block) - %eval((4*&block)-1) / rangecolor=&newcolr12;
        range %eval(4*&block) - %eval((5*&block)-1) / rangecolor=&newcolr11;
        range %eval(5*&block) - %eval((6*&block)-1) / rangecolor=&newcolr10;
        range %eval(6*&block) - %eval((7*&block)-1) / rangecolor=&newcolr9;
        range %eval(7*&block) - %eval((8*&block)-1) / rangecolor=&newcolr8;
        range %eval(8*&block) - %eval((9*&block)-1) / rangecolor=&newcolr7;
        range %eval(9*&block) - %eval((10*&block)-1) / rangecolor=&newcolr6;
        range %eval(10*&block) - %eval((11*&block)-1) / rangecolor=&newcolr5;
        range %eval(11*&block) - %eval((12*&block)-1) / rangecolor=&newcolr4;
        range %eval(12*&block) - %eval((13*&block)-1) / rangecolor=&newcolr3;
        range %eval(13*&block) - %eval((14*&block)-1) / rangecolor=&newcolr2;
        range %eval(14*&block) - %eval(15*&block) / rangecolor=&newcolr1;
      endrangeattrmap ;

      * associate the colour map with the SLD_P variable;
      rangeattrvar attrvar=rangevar var=sld_p attrmap="recmap";
```

Code Snippets: Build the attribute map for the disease statuses – static colour for each possible disease status.

```
* assign colours for the disease status scatterplot (overlaid on bubbles);
discreteattrmap name="dstatus";
  value " " / markerattrs = (color = white transparency=1);
  value "(4) Progressive Disease" / markerattrs = (color = gold);
  value "(2) Partial Response" / markerattrs = (color = cyan);
  value "(3) Stable Disease" / markerattrs = (color = white);
  value "(1) Complete Response" / markerattrs = (color = lime);
  value "(5) Death" / markerattrs = (color = salmon);
enddiscreteattrmap;
```

```
* associate the scatterplot colour map with the STAT variable;
discreteattrvar attrvar=rv var=stat attrmap="dstatus" ;

.....:
entrytitle      halign=left textattrs=(family='light' size=5) 'Tumour Burden Evaluation by patient :.....:
Using RECIST 1.1';
entrytitle      halign=left textattrs=(family='light' size=5) "Treatment Group : &treatm. mg Active-drug";
entryfootnote   halign=left textattrs=(family='light' size=5) "Bubble sub-text : [ Disease Status at visit
& n (target) / n (non-target) ]";
entryfootnote   halign=left textattrs=(family='light' size=5) "BOR -> Best Overall Response
DOR -> Duration of Response
PFS -> Progression Free Survival
DSD -> Duration of SD";
```

Code Snippets: Overlay the plots (Bubble Plot and Scatter Plot)

```
layout overlay/  
  /* define the X axis*/  
  xaxisopts=(label="Visit (Month)" labelattrs=(weight=normal size=5pt family='light')  
  linearopts=(viewmin=0.1 viewmax=18 tickvaluesequences=(start=0 end=18 increment=1)) offsetmin=0 offsetmax=0  
  tickvalueattrs=(family="light" weight=normal size=5pt))
```

```
  /* define the Y axis*/  
  yaxisopts=(label="Subject" labelattrs=(weight=normal size=6pt family='light')  
  linearopts=(viewmin=0 viewmax=14 tickvaluesequences=(start=0 end=8 increment=1))  
  tickvalueattrs=(weight=normal size=5pt family="light")  
  type=discrete griddisplay=on offsetmin=0.075 offsetmax=0.075 display=all);
```

```
  /* add 2 reference lines to separate baseline and summary sections*/  
  referenceline x=1.5 / lineattrs=(color=magenta pattern=dot thickness=1.5);  
  referenceline x=13 / lineattrs=(color=magenta pattern=dot thickness=1.5);
```

```
  /* define the bubbleplot - based on SLD2 - linked to colour map via "rangevar"*/  
  bubbleplot x=visit y=subject size=sld_p / datalabel=status datalabelposition=bottom  
    datalabelattrs=(size=5 color=black family="light")  
    colorresponse=rangevar relativescale=false relativescaletype=proportional display=(fill)  
    bubbleradiusmin=0 bubbleradiusmax=12 includemissinggroup=true;
```

```
  /* define the scatterplot based on STAT(linked via the "rv" reference*/  
  scatterplot x=visit y=subject / markerattrs=(size=5 symbol=circlefilled) group=rv  
  groupdisplay=overlay name='rv2';
```

```
  discretelegend 'rv2' / sortorder=ascendingformatted backgroundcolor=verylightgrey across=6  
    valueattrs=(weight=bold size=5pt family='light');
```

```
endlayout;  
endgraph;
```

```
end;
```

```
run;
```

Code Snippets: Call the macro to generate figures – define base, secondary and variable colours, along with a subset.

```
* render the graph from the template;  
proc sgrender data=plotrdy template=recistplot;  
  where treatm=&treatm;  
  format stat $legend.;  
run;
```

```
%mend colgen;
```

```
* call macro for each treatment group;  
%colgen(basecol=red,    second=red,    varcol=blue,  treatm=1)  
%colgen(basecol=red,    second=red,    varcol=green, treatm=2)  
%colgen(basecol=red,    second=green,   varcol=blue,  treatm=3)
```

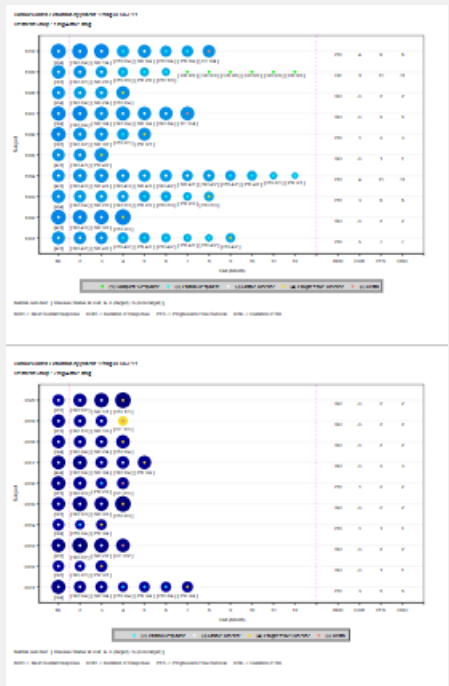
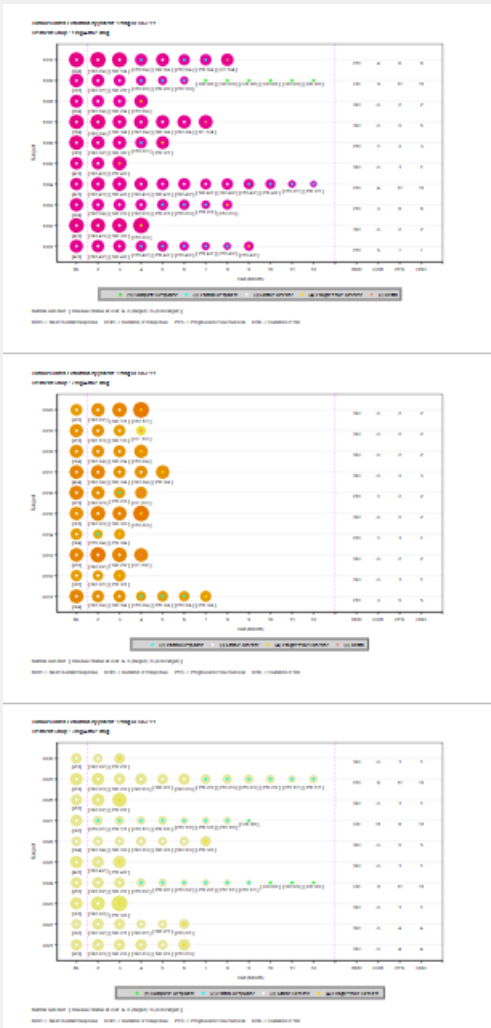
```
%colgen(basecol=green,  second=green,  varcol=blue,  treatm=1)  
%colgen(basecol=green,  second=green,  varcol=red,   treatm=2)
```

```
%colgen(basecol=blue,   second=blue,   varcol=red,   treatm=3)  
%colgen(basecol=blue,   second=blue,   varcol=green, treatm=1)  
%colgen(basecol=blue,   second=blue,   varcol=blue,  treatm=2)
```

RECISTance is Useless

Output plots showing the various colour combinations that were defined.

1
7



THE END

[Full SAS Code Download Link](#)

Daniel.Lindenbaum@phastar.com

