

Grouped Jittered Box Plots in SAS 9.2 and SAS 9.3

Volker Harm, Bayer Pharma AG, Berlin, Germany
Catalina Mejía Herrera, Bayer Pharma AG, Berlin, Germany

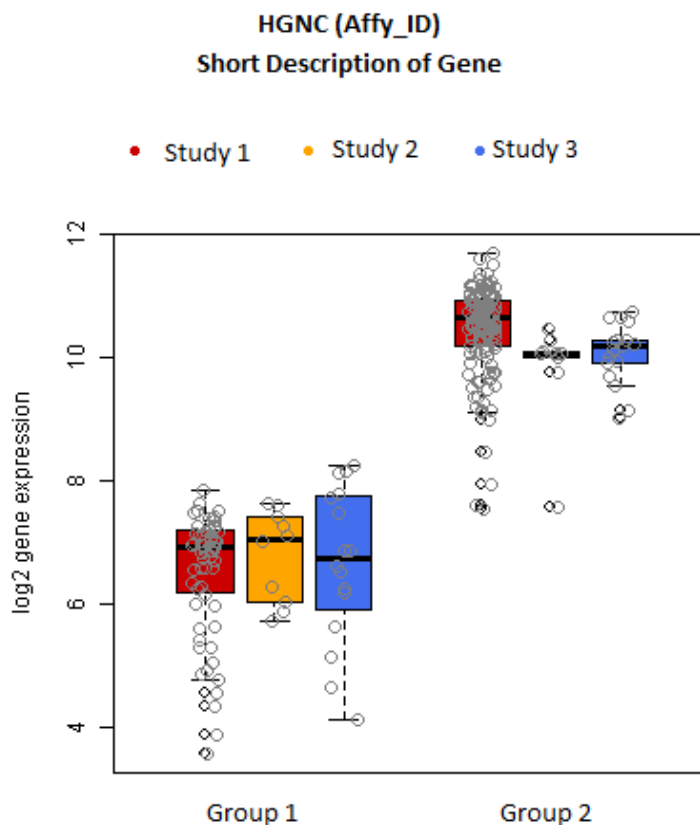
ABSTRACT

In gene expression analysis often box plots are required with the individual jittered data overlaid. Two solutions are presented with SAS Statistical Graphics. In SAS 9.2 the box plots have to be constructed with vector statements, in SAS 9.3 a more natural solution is possible using a categorical and a linear x-axis.

INTRODUCTION

When presenting results of gene expression analyses often results of two factors for different studies are compared. To guide the audience in interpreting the box plots the individual data are overlaid spread for a certain random amount within the area of the box plot.

In our department the main tool for gene expression analyses is the Bioconductor software that uses the R statistical programming language. With R functions `boxplot()` and `stripchart()` hands-on solutions were created leading to plots which typically looked like the following:



As we are a SAS shop and the graphics capabilities of SAS 9.2 promised a lot, we got the task to develop a SAS macro to be used routinely for producing this kind of displays to be used in presentations resembling as much as possible the appearance of the graphs above.

THE DATA

MEASUREMENTS

The measurements are given in a long data set with columns containing the Affymetrix gene identification, the log2-transformed gene expression measurements and columns denoting the group (numeric and textual) and the study (numeric and textual) the measurement belongs to. That means a data set is given with the following structure:

Variable name	Description	Type
Affy_ID	Affymetrix gene identification	Character
GroupNo	Group number	Numeric
GroupName	Group name	Character
StudyNo	Study number	Numeric
StudyName	StudyName	Character
Value	Log2 gene expression	Numeric

JITTERING

Jittering will be achieved by introducing a new variable GroupJittered, which will be the group number with some random noise added:

$$\text{StudyNoJittered} = \text{StudyNo} + 0.3 * (\text{uniform}(1) - 0.5).$$

This is a quite simple form of jittering; more general approaches can be found in Friendly (2011) and Wicklin (2011).

GENE INFORMATION

A second data set contains data about the genes. For each Affymetrix gene identification it has a column for the unique gene name (HGNC) and a short description. That means a data set is given with the following structure:

Variable name	Description	Type
Affy_ID	Affymetrix gene identification	Character
HGNC	Unique gene name	Character
Short_Description	Short description	Character

Source code for creating example data sets is given in the appendix.

THE TASK

Given the set of data for each gene in the measurements table and the metadata in the gene information table create a graph displaying the log2 gene expressions for each group by a box plot for each study overlaid with the individual measurements jittered within the box plot area. The box plot style should be schematic and not displaying the mean. The studies should be indicated by colours. The titles of the graph should contain the gene information, the studies displayed should be shown in a legend above the graph, the groups should be indicated below the x-axis.

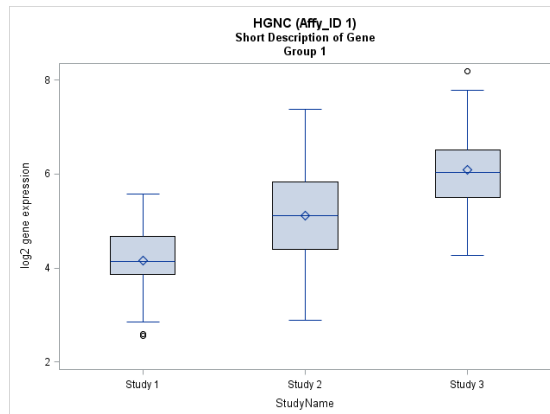
All this should be realized by exploiting as much as possible the new statistical graphics capabilities of SAS. Where possible the Statistical Graphics Procedures proc sgplot for one group and proc sgpanel for multiple groups should be used. Otherwise we looked for a solution with the Graph Template Language (GTL), which allows more options to modify the appearance of the graph.

WHAT DOES NOT WORK

Box and scatter plots are easily created in SAS 9.2 and SAS 9.3 with proc sgplot.

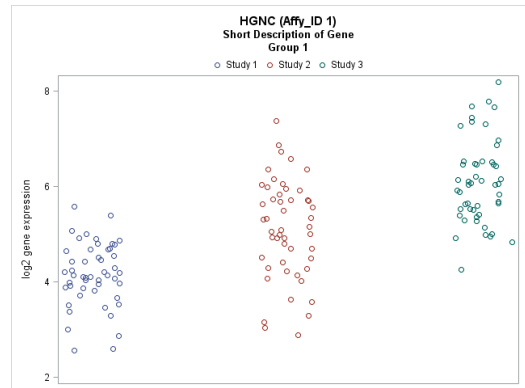
A box plot is created with the vbox statement. The following code produces box plots for group 1 of our example data as shown.

```
proc sgplot data = Expressions;
  title "&HGNC. (%trim (&Affy_ID.))";
  title2 "&Short_Description.";
  title3 "Group 1";
  vbox Value/ category = StudyName;
  where GroupName = 'Group 1';
run;
```



As easy as this graph is produced there are some flaws regarding to our requirements: The x-axis has tick marks for the studies and a label, which are not needed, but there is no legend. The box plot is schematic, but displays the mean and the outliers which both are not needed. The jittered measurements can be displayed in a graph by using the scatter statement. The following code produces scatter plots for group 1 of our example data as shown.

```
proc sgplot data = GraphData;
  title "&HGNC. (%trim (&Affy_ID.))";
  title2 "&Short_Description.";
  title3 "Group 1";
  xaxis display = none;
  scatter x = StudyNoJittered y = Value/
    group = StudyName;
  keylegend/ position = top noborder;
  where GroupName = 'Group 1';
run;
```



Here we have the studies coloured and a legend for the studies below the title using the xaxis and keylegend statement.

This looks promising: If we could overlay the graphs, we could combine features, and then use proc sgpanel to add a further group level. Unfortunately this does not work. The following code

```
proc sgplot data = GraphData;
  title "&HGNC. (%trim (&Affy_ID.))";
  title2 "&Short_Description.";
  title3 "Group 1";
  xaxis label = " ";
  vbox Value/ category = StudyName;
  scatter x = StudyNoJittered y = Value/ group = StudyName;
  keylegend/ position = top noborder;
  where GroupName = 'Group 1';
run;
```

results in the following message:

```
ERROR: Attempting to overlay incompatible plot or chart types
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE SGPLOT used (Total process time):
      real time          0.01 seconds
      cpu time           0.00 seconds
```

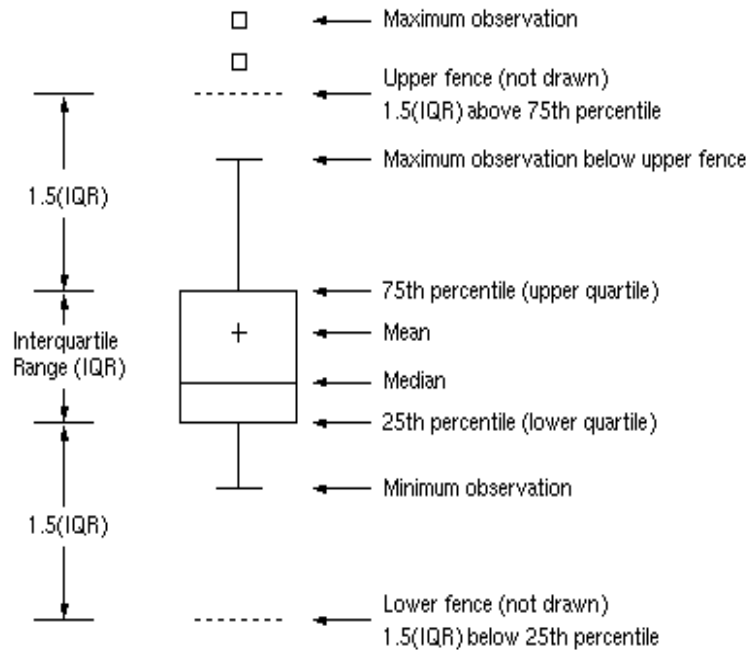
SAS is very strict in defining plot types for proc sgplot; the scatter plot is typed as basic, the box plot as distribution. By definition they are not compatible, and this is true for SAS 9.2 and SAS 9.3 (see SAS Institute Inc. 2010 and SAS Institute Inc. 2012). Furthermore, box plots cannot be combined with any other plot types.

BUILDING OUR OWN BOX PLOTS

The requirements on the box plots in the graph are quite simplistic. Therefore building the box plots by drawing them instead of using the vbox statement could be an alternative. In Harris. 2010 we found the idea to use the vector statement in proc sgplot to do this. The vector statement is available for proc sgplot and proc sgpanel with SAS 9.2 Phase 2 and later. It draws arrows from a point of origin to data points. These arrows may also be lines using the option noarrowheads. The main advantage we gain using this statement is that it a basic plot, which can be overlaid by a scatter plot.

To create the box plots in this way we have to calculate the box plots statistics and the coordinates for the box plot beforehand.

In the following figure we see a typical description of a box plot, from which we can easily identify the needed statistics:



The boxes are defined by the lower quartile (q1) and the upper quartile (q3) and the line goes through the median. The so-called whiskers below and above the box are defined by the interquartile range (qrange) and minimum (min) and maximum (max) of the observations. These statistics can easily be calculated by proc means and merged to our data with the following code:

```
proc sort data = GraphData;
    by GroupName StudyNo;
run;
proc means data = GraphData noprint;
    by GroupName StudyNo;
    var Value;
    output out = BoxPlotStats mean = mean median = median q1 = q1 q3 = q3
        qrange = qrange min = min max = max;
run;
data GraphData;
    merge GraphData BoxPlotStats;
    by GroupName StudyNo;
run;
```

Now as we have the statistics we can calculate the coordinates of the box plots according to the figure above:

```
data GraphData;
    set GraphData;
    XBoxLeft      = StudyNo - 0.25;
    XBoxRight     = StudyNo + 0.25;
    XWhiskerLeft  = StudyNo - 0.05;
    XWhiskerRight = StudyNo + 0.05;
    YWhiskerTop   = (1.5*qrange) + q3;
    YWhiskerBottom = q1 - (1.5*qrange);

    if max le YWhiskerTop then
        YWhiskerTop = max;

    if min ge YWhiskerBottom then
        YWhiskerBottom = min;
run;
```

The constant 0.25 was arbitrarily chosen and determines the box width. This data provided and again looking at the figure above defining a box plot we can formulate the SAS code for the box plots. As we want to use it within proc sgplot as well as in proc sgpanel we put it in a macro:

PhUSE 2012

```
%macro DrawBoxPlots;
  * draw median line;
  vector x = XBoxRight y = median/
    noarrowheads xorigin = XBoxLeft yorigin = median;
  * draw quantile box;
  vector x = XBoxRight y = q1/
    noarrowheads xorigin = XBoxLeft yorigin = q1;
  vector x = XBoxRight y = q3/
    noarrowheads xorigin = XBoxLeft yorigin = q3;
  vector x = XBoxLeft y = q3/
    noarrowheads xorigin = XBoxLeft yorigin = q1;
  vector x = XBoxRight y = q3/
    noarrowheads xorigin = XBoxRight yorigin = q1;
  * draw whiskers;
  vector x = StudyNo y = YWhiskerTop/
    noarrowheads xorigin = StudyNo yorigin = q3;
  vector x = StudyNo y = YWhiskerBottom/
    noarrowheads xorigin = StudyNo yorigin = q1;
  vector x = XWhiskerRight y = YWhiskerTop/
    noarrowheads xorigin = XWhiskerLeft yorigin = YWhiskerTop;
  vector x = XWhiskerRight y = YWhiskerBottom/
    noarrowheads xorigin = XWhiskerLeft yorigin = YWhiskerBottom;
%mend;
```

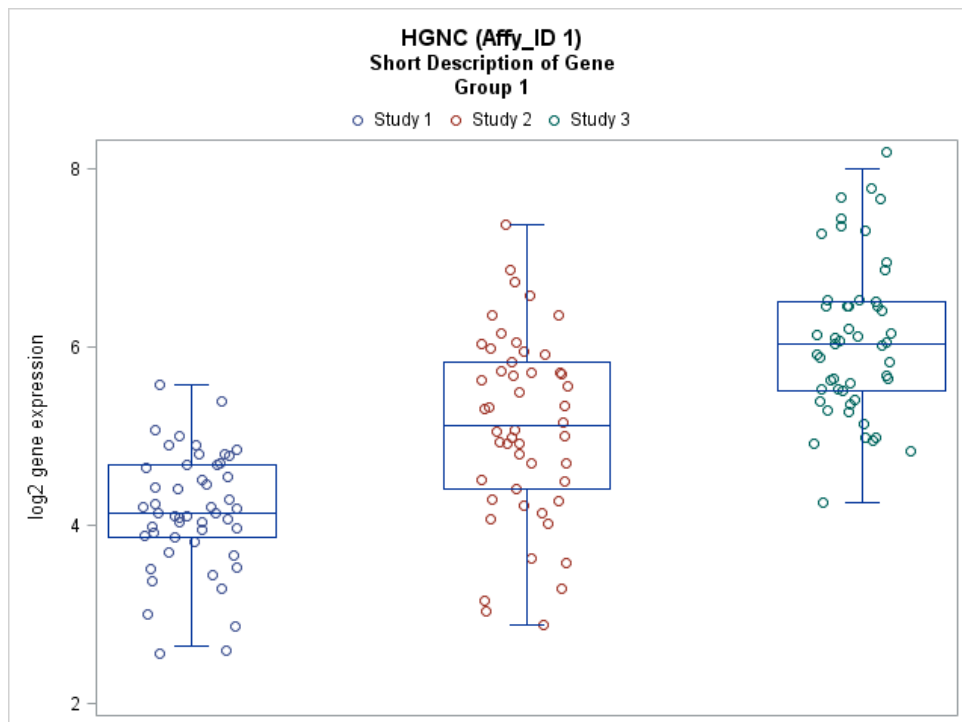
Putting all this and the scatter plot discussed above into the following proc sgplot statement

```
proc sgplot data = GraphData (where = (GroupName = 'Group 1'));
  keylegend/ position = top noborder;
  axis display = none;

  * draw jittered values;
  scatter x = StudyNoJittered y = Value/ group = StudyName;
  keylegend/ position = top noborder;

  %DrawBoxPlots;
run;
```

results in the following graph (Titles are set outside of the sgplot statement.):



After all this preparations to get the jittered measurements overlaid the extension of this solution for multiple groups is straightforward using proc sgpanel:

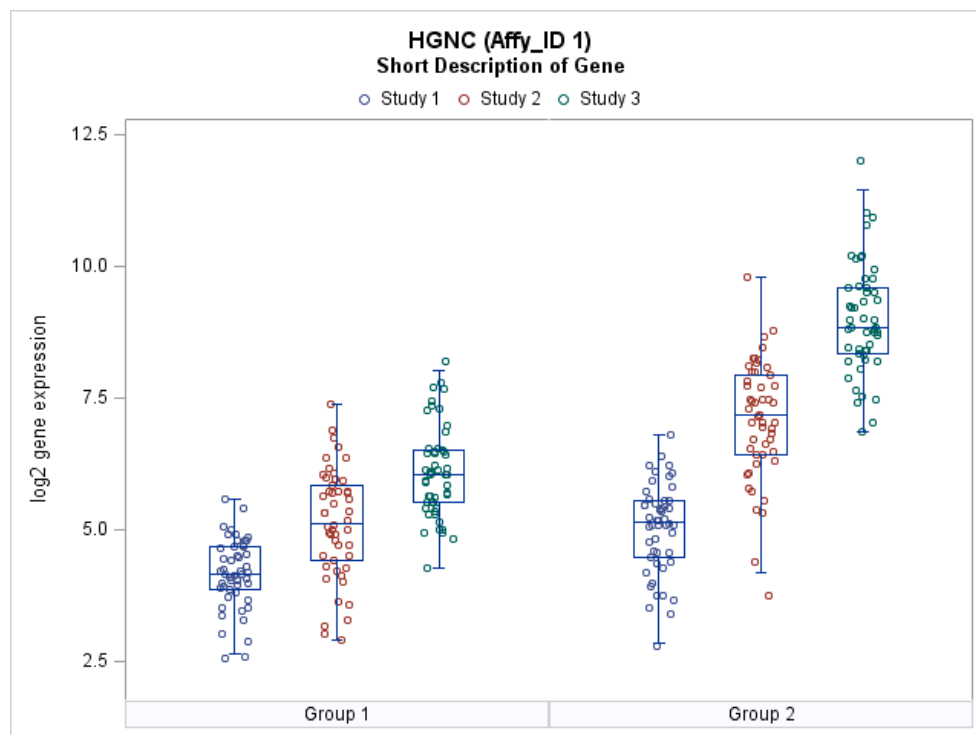
```
proc sgpanel data = GraphData;
  * define the structure of the panel;
  panelby GroupName /
    novarname layout = columnlattice colheaderpos = bottom noborder;

  * leave some room between the columns of the panel;
  colaxis display = none offsetmax = 0.2 offsetmin = 0.2;

  * draw jittered values;
  scatter x = StudyNoJittered y = Value / group = StudyName;
  keylegend / position = top noborder;

  %DrawBoxPlots;
run;
```

The panelby statement defines that the groups should be displayed side by side indicated with group indicators at the bottom, the colaxis statement cares for a good visual appeal of the box plots within the graph. The result is as following:



This is an acceptable solution for our task formulated above: The box plots are overlaid with jittered data, studies are indicated by colours, grouping for two levels is possible, and titles and axis labels are handled correctly. Furthermore this solution is easily integrated into a general macro. Room for improvement is in coloring the boxes and simplifying the group indicators. Also keep in mind that the box plots only fulfill minimal requirements.

EXPLOITING SAS 9.3

In SAS 9.3 there are two new features in the Graph Template Language (GTL), which are quite usable for our task:

Box plots now support an independent, numeric axis, so that box and scatter plots can be overlaid and there is a new group option for box and scatter plots, which gives a convenient way to fill the boxes with colors according to the studies.

We will start with one group level.

Box plots inherently have a discrete axis with equal spacing between the values. This is the reason, why in SAS 9.2 they are not compatible with linear axes needed by scatter plots. In SAS 9.3 this has changed and, if in the layout overlay statement the x-axis is explicitly defined as linear (type = linear), the box plot can use this axis. The equivalents to the vbox and scatter statements in GTL are the BoxPlot and ScatterPlot statements. In the BoxPlot statement we use the new option Group that colors the boxes for each study. In the Display option we

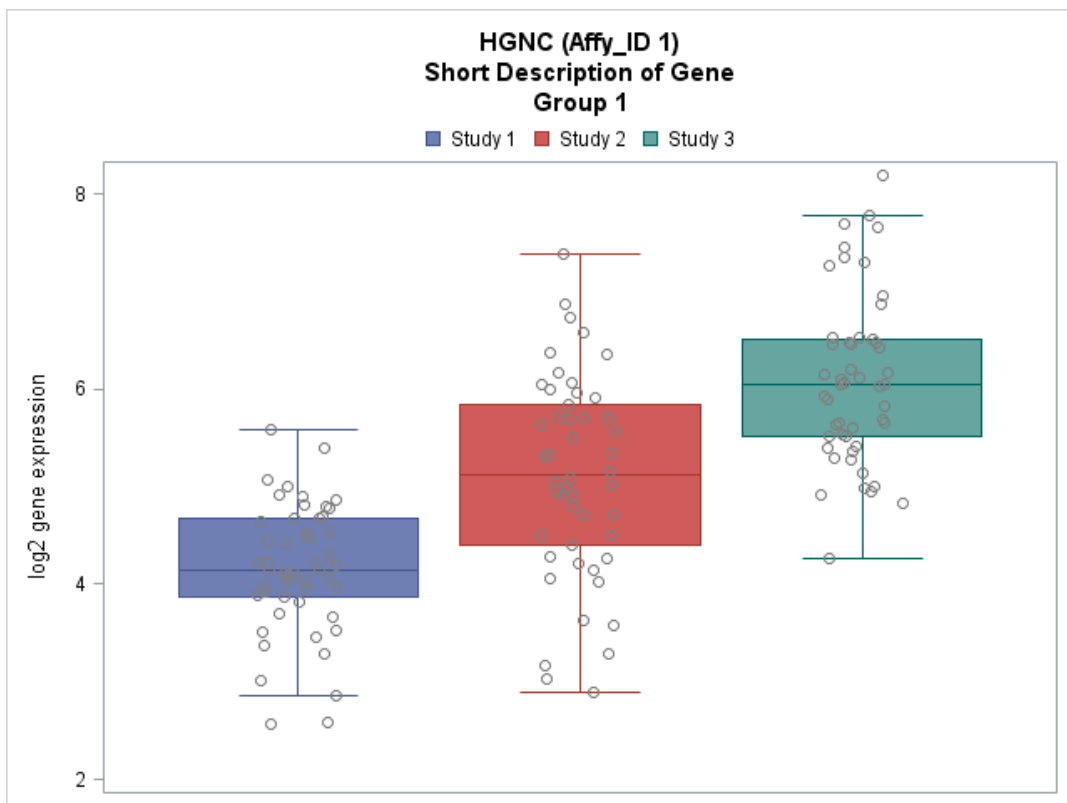
PhUSE 2012

have to exclude the outliers as we overlay all the individual values. In the ScatterPlot statement we use the markerattrs option to gray the overlaid individual data. Setting the other values analog to the sgplot solution we get the following for onid group level:

```
proc template;
  define statgraph Overlaid;
    begingraph /;
      EntryTitle "&HGNC. (%trim (&Affy_ID.))";
      EntryTitle "&Short_Description.";
      EntryTitle "Group 1";
      layout overlay/ xaxisopts = (display = none type = linear);
        BoxPlot x = StudyNo y = Value/
          Group = StudyName
          Display = (caps median fill)
          LegendLabel = "log2 gene expression"
          Name = 'BOX';
        ScatterPlot x = StudyNoJittered y = Value/
          markerattrs = (color = gray);
        DiscreteLegend "BOX"/ Valign = top Border = false;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data = GraphData (where = (GroupName = 'Group 1'))
  template = Overlaid;
run;
```

This results in the following graph:



In a last step we want to extend our results to the second group level. The analogon to proc sgpanel in the Graph Template Language is layout datalattice. Unfortunately the boxplot statement cannot be used in layout datalattice. Instead the boxplotparm statement must be used, which causes some preprocessing of the data. SAS supports this by providing a macro %boxcompute to calculate the statistics needed and deliver them in a format that is understood by the boxplotparm statement. The results have to be added to the graph data set. The final input data set has the following structure:

PhUSE 2012

Variable name	Description	Type
STAT	Name of the box plot statistic	Character
BoxValue	Value of the box plot statistic	Numeric
Affy_ID	Affymetrix gene identification	Character
GroupNo	Group number	Numeric
GroupName	Group name	Character
StudyNo	Study number	Numeric
StudyName	Study name	Character
StudyNoJittered	Jittered study number	Numeric
Value	Log2 gene expression	Numeric

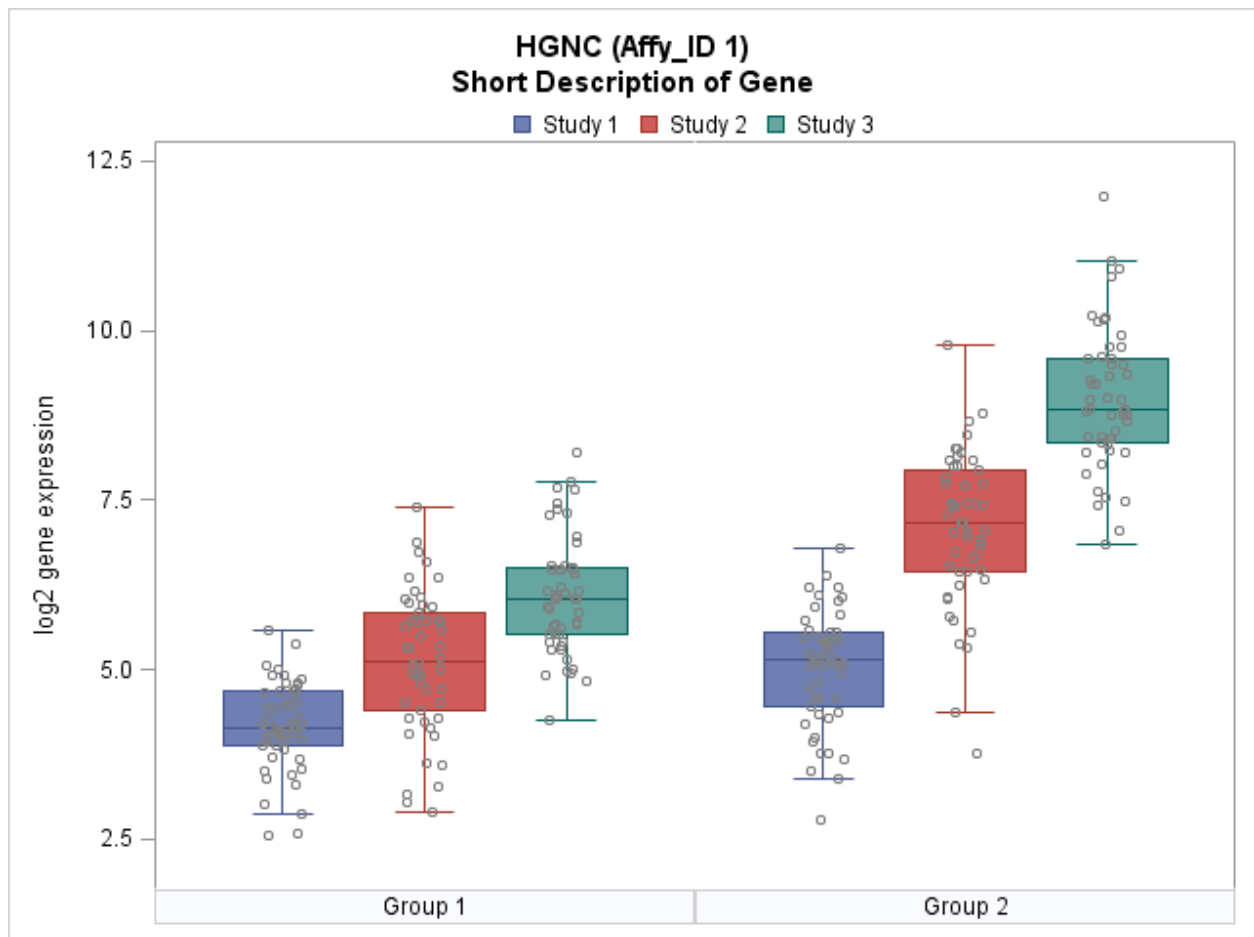
A macro to create this input data set is given in the appendix.

Given this graph input data set we can use the following code, which resembles the grouping features of the sgpanel solution to produce a suitable template and render it:

```
proc template;
  define statgraph JitteredGroupedBoxPlots;
    begingraph;
      EntryTitle "&HGNC. (%trim (&Affy_ID.))";
      EntryTitle "&Short_Description.";
      layout datalattice columnvar = GroupName/
        ColumnHeaders = bottom
        Border = false
        ColumnAxisOpts = (display = none offsetmax = 0.2
                          offsetmin = 0.2)
        HeaderLabelDisplay = value
        ;
      layout prototype/ walldisplay = none;
        BoxPlotParm x = StudyNo Y = BoxValue stat = STAT/
          DisplayGroup = (caps fill median)
          Group = StudyName name = 'Box';
        ScatterPlot x = StudyNoJittered y = Value/
          Primary = true Markerattrs = (color = gray);
      endlayout;
      sidebar/ align=top;
        DiscreteLegend "Box" /border = false;
      endsidebar;
    endlayout;
  endgraph;
end;
run;

proc sgrender data = GraphData template = JitteredGroupedBoxPlots;
run;
```

This finally produces our final result on grouped jittered box plots:



CONCLUSION

Although it is not straightforward to produce such grouped jittered box plots we get quite appealing results, which are easy to maintain and quite flexible in handling groups in two levels. And: SAS is learning, SAS 9.3 gives major improvements to SAS 9.2. We are looking forward to further versions.

REFERENCES

- Friendly, Michael. 2006. The jitter macro. <http://www.datavis.ca/sas/mac/jitter.html>
- Harris, Kriss. 2010. Jitter Scatter Plot with Boxplot overlaid SAS. Innovation blog. <http://krissharris.co.uk/wp>
- SAS Institute Inc. 2010. SAS/GRAPH® 9.2: Statistical Graphics Procedures Guide, Second Edition. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2012. SAS® 9.3 ODS Graphics: Procedures Guide, Third Edition. Cary, NC: SAS Institute Inc.
- Wicklin, R.. 2011. To jitter or not to jitter: That is the question. The DO Loop blog. <http://blogs.sas.com/content/iml/>
- Wicklin, R.. 2011. Jittering to prevent overplotting in statistical graphics. The DO Loop blog. <http://blogs.sas.com/content/iml/>
- Wicklin, R.. 2011. How to use transparency to overcome overplotting. The DO Loop blog. <http://blogs.sas.com/content/iml/>
- Wicklin, R.. 2012. What is the difference between categories and groups in PROC SGPLOT? . The DO Loop blog. <http://blogs.sas.com/content/iml/>

RECOMMENDED READING

- Feliu, Anthony L. 2011. Scatter Charts of Serial Observations with PROC SGPLOT and Graphics Template Language. PharmaSUG 2011, Paper TT01
- Kuhfeld, Warren F. 2010. Statistical Graphics in SAS®: An introduction to the Graph Template Language and the Statistical Graphics Procedures. Cary, NC: SAS Institute Inc.
- Matange, Sanjay, Heath, Dan. 2011. Statistical Graphics Procedures by Example: Effective Graphs Using SAS®. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:
Volker Harm

PhUSE 2012

Bayer Pharma AG
Müllerstraße 178
13342 Berlin
Work Phone: +49-30-468-11208
Email: volker.harm@bayer.com
Web: <http://www.bayerhealthcare.com>

Catalina Mejía Herrera
Bayer Pharma AG
Müllerstraße 178
13342 Berlin
Work Phone: +49-30-468-14212
Email: catalina.mejiaherrera@bayer.com
Web: <http://www.bayerhealthcare.com>

Brand and product names are trademarks of their respective companies.

APPENDIX

SOURCE CODE FOR THE EXAMPLE DATA SETS

```
%macro CreateAnnotationsDS;

    data Annotations;
        length Affy_ID $20;
        Affy_ID = 'Affy_ID 1';
        HGNC = 'HGNC';
        Short_Description = 'Short Description of Gene';
    run;

%mend;

%CreateAnnotationsDS;

%macro SetGeneInfo;

    data _null_;
        set Annotations;
        call symput ('Affy_ID', Affy_ID);
        call symput ('HGNC', HGNC);
        call symput ('Short_Description', Short_Description);
    run;

%mend;

* gene info macro variables;
%global Affy_ID HGNC Short_Description;

%SetGeneInfo;

%macro CreateExpressionsDS;

    data Expressions (drop = i);
        call streaminit(123);
        length Affy_ID $20;
        label Value = "log2 gene expression";

        %do i = 1 %to &NoOfGenes.;
            %do j = 1 %to &NoOfGroups.;
                %do k = 1 %to &NoOfStudies.;
                    do i = 1 to 50;
                        Affy_ID = "Affy_ID &i.";
                        GroupNo = &j.;
                        GroupName = "Group &j.";
                    end;
                end;
            end;
        end;
```

PhUSE 2012

```
StudyNo = &k.;
StudyName = "Study &k.";
Value = rand('NORMAL', 10 + &j.*&k.*10, 10)/10 + 2;
output;
end;
%end;
%end;
%end;
run;

%mend;

* dimensions of expressions data set;
%let NoOfGenes = 1;
%let NoOfGroups = 2;
%let NoOfStudies = 3;

%CreateExpressionsDS;

* jitter study numbers;
%macro JitterStudyNos;

data GraphData;
set Expressions;;
StudyNoJittered = StudyNo + 0.3*(uniform (2) - 0.5);
run;

%mend;

%JitterStudyNos;
```

MACRO SOURCE CODE TO CREATE THE INPUT DATA SET FOR GROUPED JITTERED BOX PLOTS IN SAS 9.3

```
%macro CreateGraphDataBoxPlotParm;
* compute the box plot statistics for each group;
%do i = 1 %to &NoOfGroups.;

data Group&i.(where = (GroupNo = &i.));
set Expressions;
run;

* keep group and study names;
data _null_;
set Group&i. end = last;
by StudyNo;
call symput ("GroupName&i.", GroupName);

if first.StudyNo then do;
XStudyNo = put (StudyNo, best.);
call symput (compress ("StudyName"||XStudyNo), StudyName);
NoOfStudies + 1;
end;

if last then do;
call symput ('NoOfStudies', NoOfStudies);
end;
run;

* compute the box plot statistics for one group (see SAS documentation);
%boxcompute (
indsn = Group&i.,
x = StudyNo, y = Value, outdsn = boxdata&i.,
datalabel =, qntldef = 5, table = no);
```

PhUSE 2012

```
* get group and study names;
data boxdata&i.;
  set boxdata&i.(rename = (x = StudyNo));
  GroupName = "&&GroupName&i.";

  %do j = 1 %to &NoOfStudies.;
    if StudyNo = &j. then
      StudyName = "&&StudyName&j.";
  %end;
run;

%end;

* combine the box plot statistics;
data boxdata;
  set boxdata1-boxdata&NoOfGroups.;
run;

* declare the individual data as outliers;
data GraphData;
  set GraphData;
  STAT = 'OUTLIER';
  BoxValue = Value;
run;

* create the final graph data set;
data GraphData;
  set Boxdata (rename = (Value = BoxValue)) Graphdata;
run;

%mend;
```