PharmaSUG China

Extend SAS by R

Derek Li, Bayer Healthcare Company Ltd., Beijing, China
Helen Wu, Bayer Healthcare Company Ltd., Beijing, China

ABSTRACT

Figures, simulations, and new statistical methodologies have become increasingly important and needed in pharmaceutical industry. Even though SAS is a powerful leading statistical analysis software in pharmaceutical industry, it is still not realistic to depend only on SAS to deal with all figures, simulations and new methodologies. Therefor extend SAS by some other tools are needed and there are several possible ways to extend SAS. One attractive option is to extend SAS by R, a free statistical environment which offers a wide of variety of statistical and computing techniques. This paper will share the knowledge about how to extend SAS by R and illustrate corresponding application using graph programming.

INTRODUCTION

It is well known that “a picture is worth a thousand of tables” and thus it is not surprising that figures have been used more and more frequently in reporting of clinical trials in pharmaceutical industry. In addition, complicated simulation and new statistical methods are becoming more and more needed in pharmaceutical industry. Even though SAS is a powerful statistical analysis software and a lot of great improvements have been made to match industry need, it is not realistic to depend only on SAS to solve all the challenges. In addition, SAS is in general not updated frequently in pharmaceutical companies probably due to preference of stability. Thus, some new features in SAS might not be available in our daily work. There is great need to look beyond SAS and investigate tools that could be used by SAS solve this problem.

R is a good choice to be used by SAS for several reasons. First, R is a language and free software environment for statistical computing and graphics. Second, R provides a wide of variety of statistical and graphical techniques, including some very powerful graphics packages such as grid, ggplot2. Furthermore, R is highly extensible, an important and attractive feature. More detailed information about this feature can be found from http://www.r-project.org.

HOW TO CREATE A FIGURE IN R

The first example is a classical lattice graph illustrated using lab data (Figure 1).

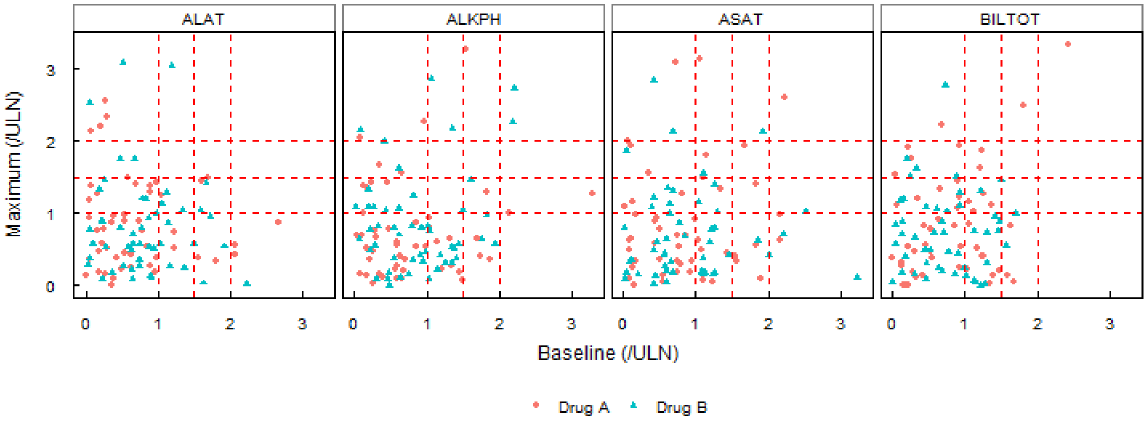

Figure 1 Lattice Graph

It is quite complicated to create this type of graph using traditional SAS SAS/graph model. However, it is pretty easy to do it in R. Below are the core R codes for Figure 1, which is in general much shorter and much more readable than SAS codes using annotation.

```
   library(ggplot2)
   library(foreign)

   lb<-read.xport("lb.xpt")
   p<-gplot(lb,aes(x=BASE,y=AVAL,group=TRTP))+geom_point(aes(shape=TRTP,colour=TRTP))+
   facet_grid(.~LBTEST)+coord_equal()+mytheme()+theme(strip.background=element_rect(fi
   ll='transparent'))

   p<-p+labs(x="Baseline (/ULN)",y="Maximum (/ULN)")
   line.data<-data.frame(x=seq(1,2,0.5),y=seq(1,2,0.5))

   #p<-p+geom_abline()
   p<-p+geom_vline(aes(xintercept = x), data = line.data,linetype=2,color='red')
   p<-p+geom_hline(aes(yintercept = y), data = line.data,linetype=2,color='red')
```

Please note this figure can be created easily by using GTL or PROC SGPANEL (only available in SAS 9.2 or above). Codes can be found in Clinical Graphs using ODS Graphics [Sanjay Matange]. However, if you do not have SAS 9.2 or above and there is unlikely to have SAS update in recent future, then you can create graph using this R code. Details of how to use this R code through SAS will be introduced in next section.

The second example is a line plot for PK data where in general data is measured more frequently at the early time points right after dosing and special characters might be needed in the graph to denote. To display PK data clearly, graph as shown in Figure 2 is commonly utilized.

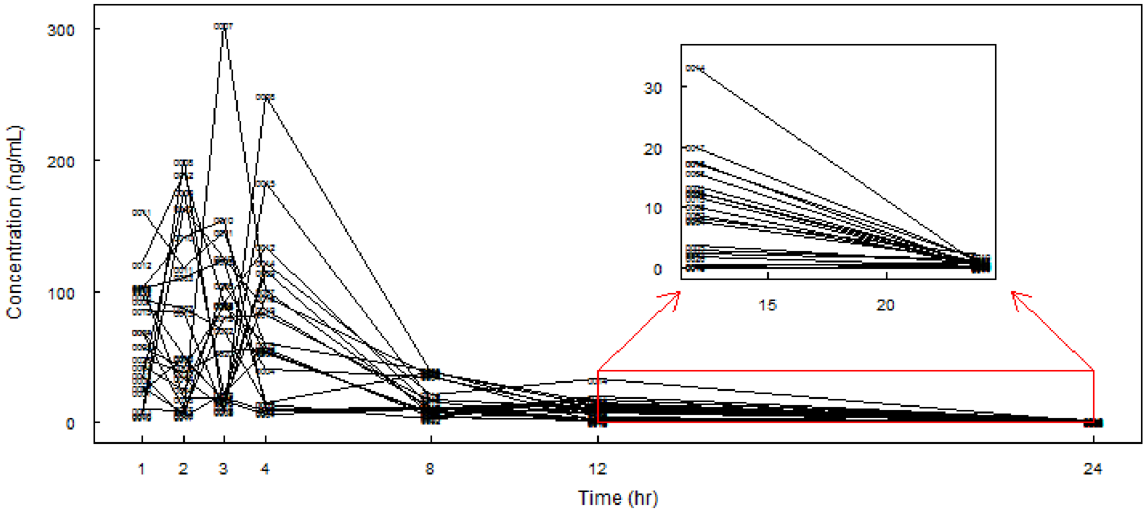

Figure 2 Line plot for PK data

For example 2, before SAS 9.3, GTL could not be used anymore due to the annotation capacity limitation. Instead, PROC GREPLAY or DSGI have to be used after two graphs are created by the traditional SAS graph model. But since a lot of annotations have to be used in the traditional SAS graph model for this type of graph, the program become too complicated. A much simpler way to create this graph is using R with following example codes:

```
pc<-read.xport("pc.xpt")
p1<-ggplot(pc,aes(x=TPT,y=AVAL,group=ID))+geom_line()+
     labs(x="Time (hr)",y="Concentration (ng/mL)")+
     scale_x_continuous(breaks=c(1,2,3,4,8,12,24))+mytheme()
p3<-p1+geom_text(aes(label=ID),size=2.0)

p4<-p3+scale_x_continuous(" ",limits=c(12, 24))+scale_y_continuous("",limits=c(0, 35))

g<-ggplotGrob(p4)

p5<-p3+annotation_custom(grob=g,xmin=12,xmax=22,ymin=50,ymax=300)
```

```
p6<-p5+geom_rect(xmin=12,xmax=24,ymin=0,ymax=40,color='red',fill='transparent')

p6<-p6+geom_segment(x=12,y=40,xend=14,yend=100,color='red',arrow = arrow(length =
unit(0.5,"cm")))

p6<-p6+geom_segment(x=24,y=40,xend=22,yend=100,color='red',arrow = arrow(length =
unit(0.5,"cm")))
```

RUN R THROUGH SAS

From above examples, we can see that R is a powerful and convenient tool to create graphs. However, as a SAS programmer, the main programming platform is SAS. It is inefficient to switch from SAS and R back and forth manually. Then, how to run R through SAS automatically? In the following, we will illustrate the details of this process in three steps.

1. Prepare the graph datasets in SAS

```
   data test;
   do i=1 to 1000;
       x=normal(0);
       y=x+normal(0);
       output;
    end;
  run;
```

2. Create the body part of R file in SAS data step

```
data r_body;
     length varl $1000;
     format varl $1000.;
     input varl;
cards4;
pc<-read.xport("pc.xpt")
p1<-ggplot(pc,aes(x=TPT,y=AVAL,group=ID))+geom_line()+
     labs(x="Time (hr)",y="Concentration (ng/mL)")+
scale_x_continuous(breaks=c(1,2,3,4,8,12,24))+mytheme()ggsave(p,file="t2.png")
....
;;;;
run;
```

3. Create whole R file through SAS and run it

```
   data r_header;
     length varl $1000;
     format varl $1000.;
     varl=’setwd(“&workspace”);
     output;
     varl=”library(ggplot2);
     output;
     varl=”library(foreign);
     output;
     test<-read.xport(“&workpath./test.xport”);
     output;
   run;

   data r_foot;
     length varl $1000;
     format varl $1000.;
     *do the data.frame to a file for further use in SAS as needed;
     varl=’Write.csv(test,file=’||”&workpath./test.csv”||’)’;
```

```
  output;
  varl=’q()’;
  output;
run;

data _null_;
  length varl $1000;
  file “&workpath./r_code.r”;
  set r_header r_body r_foot;
  put varl;
run;


 *run R:

 *&PATH=the physical install path of R;

 Systask command “&RPATH CMD BATCH –vanilla –quiet r_code.r r_log.txt” wait;
```

CALL R DIRECTLY IN SAS

From above, we can see that R is a powerful tool, and it can be used by SAS programmer to create graph. However, the switch between R and SAS could be troublesome sometimes. Is there any way to call R in SAS like call JAVA Object? The exciting news is that R is combined into IML model in SAS and it can be called in SAS/IML directly. Please see below for the example codes:

```
proc IML;
  run ExportDataSetToR(“work.pc”,”dataframe”);
  submit /R
  p1<-ggplot(pc,aes(x=TPT,y=AVAL,group=ID))+geom_line()+
  labs(x="Time (hr)",y="Concentration (ng/mL)")+
  scale_x_continuous(breaks=c(1,2,3,4,8,12,24))+mytheme()
endsubmit;
quit;
```

This is a convenient and efficient way of working with SAS and R together. But it should be noted that the matching version between SAS and R is required. Please see below for the list of recent SAS releases and the corresponding R versions that SAS supports.

SAS Version	PROC IML	R Versions
9.2	NA	2.6.1 - 2.11.1
9.22	9.22	2.9.1 - 2.11.1
9.3	9.3	2.9.1 - 2.15.3
9.3m2	12.1	2.9.1 - 2.15.3
9.4	12.3	2.13.0 - 3.0.1
9.4m1	13.1	2.13.0 - present

Besides the matching version, R_HOME environment variable must be set accordingly and the RLANG option must also be included in the SAS invocation. For further information, you can visit http://blogs.sas.com/content/iml/2013/09/16/what-versions-of-r-are-supported-by-sas

DISCUSSION

SAS is a powerful and also dominant statistical software in pharmaceutical industries. However, it is not realistic to depend only on SAS to solve all the needs such as figures, simulations, new statistical methodologies. In addition, even though there is continuous improvement in SAS, the most updated SAS version might not be available in daily work, considering that we in general depends on company to update SAS and this update is in general not frequent probably due to preference of stability in pharmaceutical industry. Looking beyond SAS, it seems that R is a good choice that could be used to extend SAS. R is free, highly extensible, easy to update with a lot of powerful packages.

The properties of SAS are stability, backwards compatibility and professional supports. When we have any question, we can easily find detail help information from the shipped SAS help documents or we can call SAS for help, but the update have to be depend on the SAS company. On the other hand, as a free software, the documentation of R is often kind of terrible and opaque, but anybody can create new features through create a package. In pharmaceutical industry SAS is the dominant statistical software, to determine when need to extend SAS by R is really depend on what kind of question we are facing. If the question can be done by SAS not very hard, our suggestion is do it by SAS, no matter it can be solve by R or not. If the question is produce some figures or something, which current SAS version we have is not easy to handle, we can think use R and even in this case, we still prefer create the graphing ready datasets by SAS.

REFERENCES

1. Wickham, H. ggplot2: Elegant Graphics for Data Analysis. (Springer, Berlin, 2009).

2. Chang, W. R Graphics Cookbook. (O’Reilly, Sebastopol, CA, 2013).

3. Sanjay Matange. Clinical Graphs using ODS Graphics, available at
http://support.sas.com/resources/papers/wusspaper.pdf

CONTACT INFORMATION

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

Name: Derek Li
Enterprise: Bayer Healthcare Company Ltd.
Address: 19F, Tower B, Bayer Center, No.27 Dong San Huan North Road, Chaoyang District
City, State ZIP: Beijing, China, 100020
Work Phone:
Fax:
E-mail: derek.li1@bayer.com
Web:

Name: Helen Wu
Enterprise: Bayer Healthcare Company Ltd.
Address: 19F, Tower B, Bayer Center, No.27 Dong San Huan North Road, Chaoyang District
City, State ZIP: Beijing, China, 100020
Work Phone:
Fax:
E-mail: Helen.wu@bayer.com
Web: