PharmaSUG China 2019 - Paper DV-067

Graphical Presentation of Clinical Data in Oncology Studies Using R

Linqing Wang, Hutchison Medi Pharma

ABSTRACT

Graphical presentation in oncology studies is essential for comprehending data and overview of efficacy analysis. ODS Graphics and SGPLOT Procedure in SAS provide various and effective methods once you have learned well with SGPLOT and the grand ODS output. R basic functionality allow you to create plot in a simple way and multiple packages are developed for specific usage, it's free, easy to get start and can be well used by clinical studies. This paper will display and discuss the procedures to generate common plots in oncology studies in R and also demonstrate on the feature of R graphics system.

INTRODUCTION

Graphical presentation of clinical data in oncology can transport message quickly and directly. Commonly generated plots like Kaplan-Meier Curve visualize the survival curve by survival distribution; Waterfall Plot display the best tumor shrinkage of each subjects and colored by response or other features at the same time; Forest Plot further analysis the treatment effect by subgroup; Swimmer Plot illustrate each subjects tumor response over time.

Both SAS and R default are adequate for plot. But when we want a customized plot, all the defaults may be burdens to be modified. SAS enhance graphic through annotate facility. Whereas R built a plot from the ground up, thus are much more directly. In this paper, I will introduce some basic R packages and then introduce the procedures to draw graphics in oncology studies.

BASIC OF R GRAPHIC

R plot is a function activated and package based which allows user to define all plot settings from the ground up. The fundamental grDevices package provide basic plot functions such as color, font or device, it can be used by other functions directly. Graphics and grid are two plot systems based on grDevices. There are many advances functions in Graphics that to generate commonly used plot, such as points, lines, boxplot, etc. Other functions can be added, like legend or mtext. Lattice and ggplot2 are two packages developed based on grid and based on the cognition of majority, thus it can be used in more complicated plots.

DATA FLOW OF ONCOLOGY EFFICACY

Oncology efficacy data consist of:

- Tumor Assessment: Tumor Diameters are measured in target lesion and sum diameters are used in assessment. Non –target lesion status and new lesion status are followed. The percent change in sum diameters is a key indicator in tumor burden and usually displayed in plot.
- Tumor Response: Target, non-target and new lesion response will be assessed and overall response of each visit will be derived per RECISIT 1.1. Best overall response of each subject will also be derived from the overall response in each visit. Response result are important in illustrate patient's time-plot.
- Time to Event Analysis: the duration of time until event are calculated and survival function is estimate from data. The time to event plot will be drew and compared by treatment .

KAPLAN-MEIER CURVE

Kaplan-Meier estimate is one of the best methods to observe subject survival status over time. It measure the number of subjects survived or saved after treatment. In a specific time interval, a vertical decline means that at a specific time point, event occurs. A horizontal prolongation means no event. Thus a positive gap in vertical or horizontal represents superiority in efficacy.

KM curve should be plotted by treatment, with different color or symbol. Legend should be added to explain line or symbol.

Sometimes, statistics and test statistical result should be added in the figure as supportive evidence. Such as median, hazard ratio and p value, etc. In this situation, statistical text should be well added and risk table is also required. Two packages are recommended in drawing km plot:

Package “survival” is a powerful package which can provide core survival analysis such as graphics, matrix, model, stats, etc. plot.survfit {survival} function is designed to produce survival curve for each strata. The premise is a survival fit objective should be produced first. The plot procedure is quiet simple, but risk-table is not allowed.

Another Package “survminer” draw survival curves using ‘ggplot2’ and can add risk table.

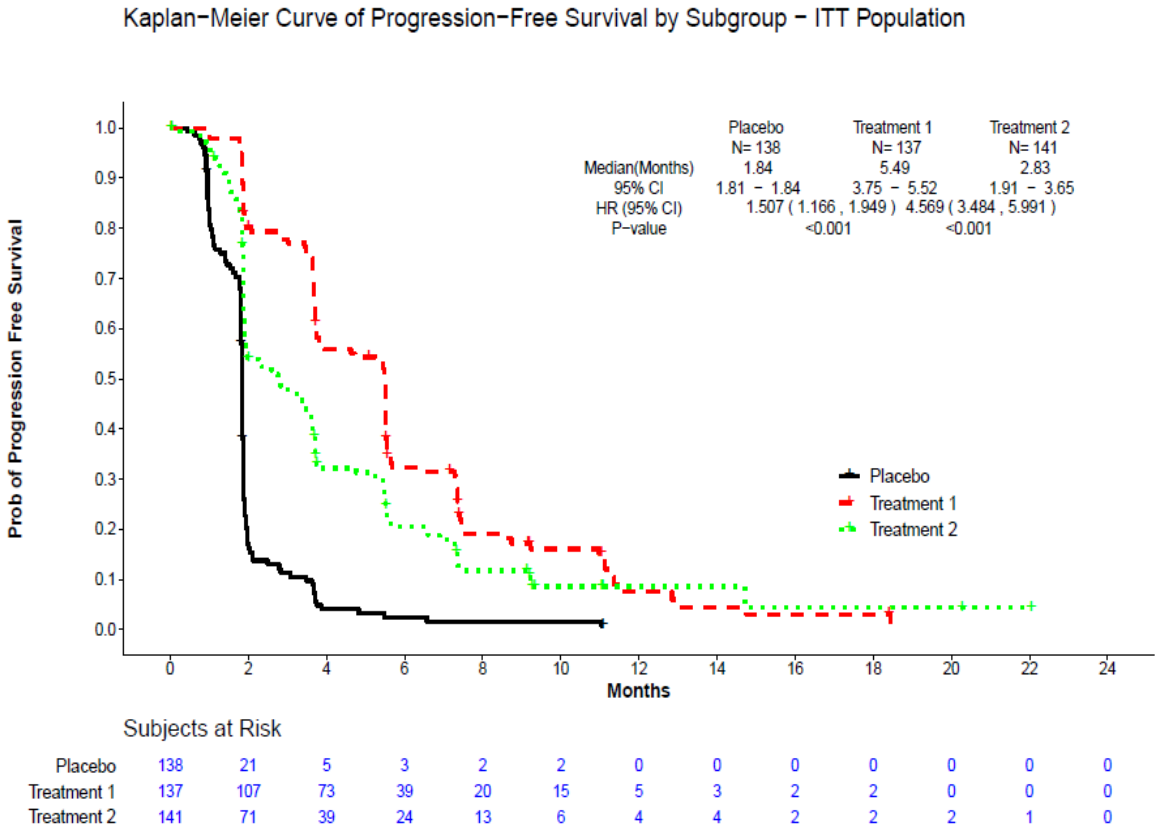

Figure 1. Kaplan-Meier Curve of Progression-Free Survival by Subgroup – ITT Population

Figure 1 can be divided into three parts: main plot, risk table and free-text. We can find the corresponding statements in the plot procedure.

1. Generate a survival fit objective

```
library("survival")
#fit curve
fit1<-survfit(Surv(AVAL,CNSR==0) ~ grpn,data=final1_1, type="kaplan-meier",conf.type = "log-log",
              conf.int=0.95)
sum<-summary(fit1)$table

## median & ci
medci<-as.data.frame(sum[,c(1,7,8,9)])
```

2. Cox-Proportion Hazard Model and Log-Rank Tests：

```
## HR
## set ref level
final1_1$grp <- factor(final1_1$grp, levels = c("Treatment 1","Treatment 2","Placebo"))
final1_1$grp = relevel(final1_1$grp, ref = "Placebo")

hr1<-coxph(Surv(AVAL,CNSR==0) ~ grp,data=final1_1,ties="efron",model=TRUE)
estimate<-as.data.frame(round(exp(hr1[["coefficients"]]),digits=3))
ci<-as.data.frame(round(exp(confint(hr1)),digits=3))
```

```
## P-value
fitp1<-survdiff(Surv(AVAL,CNSR==0) ~ grpn,data=final1_1,subset=final1_1$grpn!=3)
p.val1 <- ifelse(round(1 - pchisq(fitp1$chisq, length(fitp1$n) - 1),digit=5)>=0.001,
                 as.character(round(1 - pchisq(fitp1$chisq, length(fitp1$n) - 1),digit=5)),"<0.001"

fitp2<-survdiff(Surv(AVAL,CNSR==0) ~ grpn,data=final1_1,subset=final1_1$grpn!=2)
p.val2 <- ifelse(round(1 - pchisq(fitp2$chisq, length(fitp2$n) - 1),digit=5)>=0.001,
                 as.character(round(1 - pchisq(fitp2$chisq, length(fitp2$n) - 1),digit=5)),"<0.001"
```

3. Plot

```
## plot

par(mar=c(8,7,9,7)+0.1)
plot(fit1,mark.time=TRUE,col=c(1,2,3),lty=1:3,xaxt="n",yaxt="n",pch=1:3,lwd=2)

title(main="Kaplan-Meier Curve of Progression-Free Survival by Subgroup - ITT Population\n"
      col.main="Black",font.main=4,cex.main=1.2,
      xlab="Months",ylab="Prob of Progression Free Survival\n",
      col.lab="Black",cex.lab=1,font.lab=2, line=1)

axis(1,at=seq(0,26,by=2),las=0,cex.axis=0.8,padj=-1.5,tck=-0.01)
axis(2,at=seq(0,1,by=0.1),las=2,cex.axis=0.8,hadj=0.5,tck=-0.01)

legend(16,0.4,legend=c("Placebo","Treatment 1","Treatment 2"),lty=1:3,col=1:3,
       bty = "n",cex=0.8,y.intersp = 0.2,x.intersp=0.2,lwd=2)

text(14.5,1,"Placebo",cex=0.8,col="Black")
text(18.0,1,"Treatment 1",cex=0.8,col="Black")
text(21.5,1,"Treatment 2",cex=0.8,col="Black")

... other text statement

mtext(" \n",side=1,line=2,adj=0,cex=0.9,outer=FALSE)
mtext("Abbreviations:HR=Hazard Ratio",side=1,line=3,adj=0,cex=0.9,outer=FALSE)

... other mtext statement
```

i. mark.time control if curves are marked at each censoring time.

ii. color, linetype,linewidth and symbols are defined by col, lty, lwd and pch.

iii. x-axis and y-axis will not appear at first by setting xaxt & yaxt to “n” and later be added in axis function.

iv. legend is assigned with coordinate, label, linetype, color, interspt and box option.

v. text function can add text in plot. mtext function add text outside plot margin.

If use ‘survminer’ to add risk table:

risk.table set to “absolute” and set the other parameters.

```
## ggsurvplot

library(survminer)
ggsurv<-ggsurvplot(fit1, data = final1_1,
                   palette=c("black","red","green"), linetype=c(1,2,3),
                   tables.height = 0.15, tables.y.text = FALSE,
                   tables.theme = theme_cleantable(),

                   title="\n\nKaplan-Meier Curve of Progression-Free Survival by Subgroup - ITT Population\
                   font.title=c(17,"black"),
                   xlab="Months",font.x=c(13,"bold","black"),xlim=c(0,24),break.x.by=2,
                   ylab="Prob of Progression Free Survival",font.y=c(13,"bold","black"),
                   ylim=c(0,1),break.y.by=0.1,
                   font.tickslab=c(11,"black"),

                   risk.table = "absolute", risk.table.height=0.15,
                   risk.table.title="Subjects at Risk",
                   risk.table.fontsize=4, risk.table.col="blue",
                   risk.table.y.text=TRUE,risk.table.y.text.col=FALSE,

                   legend=c(0.75,0.3),legend.labs=c("Placebo","Treatment 1","Treatment 2"),
                   legend.title="",font.legend=12
)

ggsurv$plot <- ggsurv$plot +
  annotate("text", x = 15, y = 1, label = "Placebo", size = 4) +
  annotate("text", x = 18.5, y = 1, label = "Treatment 1", size = 4) +
  annotate("text", x = 22, y = 1, label = "Treatment 2", size = 4) +
  ... other annotate statement
```

FOREST PLOT

In oncology studies, Forest plot is a method of displaying the extent to which the estimated treatment effect differs across various subgroups of patients. It is quiet straight between subgroups and within subgroups.

Forest plot is composed by text display and main plot. Text part will list all the subgroups and other statistics will be list under treatment header. Besides that, main plot with axis and reference line is draw accordingly. Furthermore, gray strips across subgroups will help in review

There are various packages for forestplot. meta,metafor,metaviz are all support meta analysis, just like survival analysis in KM plot, it output a meta analysis objective and then generate forest plot using nested function. If our data for reporting is ready, it is more recommended to use another package which name is ‘forestplot’. The structure of desired input dataset is just the same as SAS

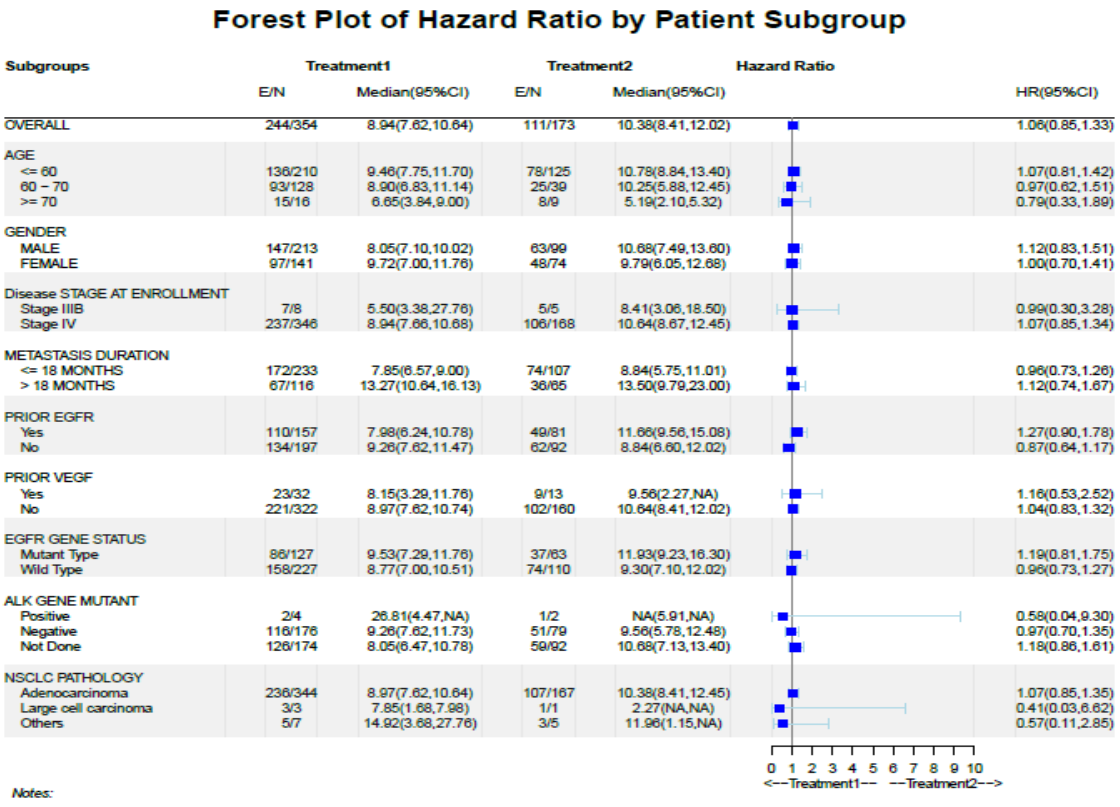

Figure 2. Kaplan-Meier Curve of Progression-Free Survival by Subgroup – ITT Population

The plot procedure in R is simple:

```
forestplot(as.matrix(report[,1:6]),
           report$HazardRatio,report$HRLowerCL,report$HRUpperCL,
           graph.pos=6, graphwidth=unit(40,"mm"),
           boxsize=0.5, lwd.ci=1, ci.vertices=TRUE, ci.vertices.height = 0.3,
           lineheight="auto", colgap=unit(8,"mm"),
           zero=1, xticks=c(0,1,2,3,4,5,6,7,8,9,10),
           title="Forest Plot of Hazard Ratio by Patient Subgroup\n",
           xlab="     <--Treatment1--   --Treatment2-->",
           col=fpColors(box="blue", lines="lightblue", zero = "gray50"),
           txt_gp = fpTxtGp(label = gpar(cex=10/14.4),
                            title = gpar(fontfamily = "", cex=10/7.2),
                            ticks = gpar(fontfamily = "", cex=10/14.4),
                            xlab  = gpar(fontfamily = "", cex = 10/14.4)
           ),
           hrzl_lines=list("1" = gpar(lwd=1, col="black"),
                           "5" = gpar(lwd=65, lineend="butt", columns=c(1:7), col="#99999922"),
                           "13" = gpar(lwd=52, lineend="butt", columns=c(1:7), col="#99999922" ),
                           "21" = gpar(lwd=52, lineend="butt", columns=c(1:7), col="#99999922" ),
                           "29" = gpar(lwd=52, lineend="butt", columns=c(1:7), col="#99999922" ),
                           "39" = gpar(lwd=65, lineend="butt", columns=c(1:7), col="#99999922" )
           )
)
grid.text("Notes:",unit(.15,'npc'),unit(.05,'npc'),gp=gpar(fontsize=8,font=3))
grid.text("Subgroups",unit(.16,'npc'),unit(.88,'npc'),gp=gpar(fontsize=9,font=2))
grid.text("Treatment1",unit(.36,'npc'),unit(.88,'npc'),gp=gpar(fontsize=9,font=2))
...other grid.text
```

i. designate labeltext: a matrix containing six column in data report
ii. designate plot column and plot in sixth column
iii. shape plot: boxsize, lwd.ci, ci.vertices…
iv. define gap between column and row: lineheight and colgap
v. col and txt_gp is to define color and text options.
vi. hrzl_lines is to add gray strips.

vii. grid.text() is to add header and footnote.

WATERFALL PLOT

In a oncology study, waterfall plot is used to present the each individual patient's response to a particular drug based on a parameter, such as tumor burden or tumor response. The x-axis stands for each patient, and y-axis measure the maximum percent change from baseline. The bar above horizontal zero means tumor growth, whereas below horizontal zero means tumor shrinkage. In general, waterfall plot goes from worst tumor growth to best tumor shrinkage and use different colors to indicate tumor response. Besides, two reference line are desired to be added to indicate partial response(PR) or progression disease(PD).

The powerful ggplot2 use ggplot+geom_bar statement, alone with theme aesthetic to help justification of plot. geom_hline and geom_text can add reference line and add text.

But it is not flexible in legend if you want to add more levels and hard to control the font in geom_text.

Whereas, barplot function of Graphics can well deal with above issues and with simpler code.

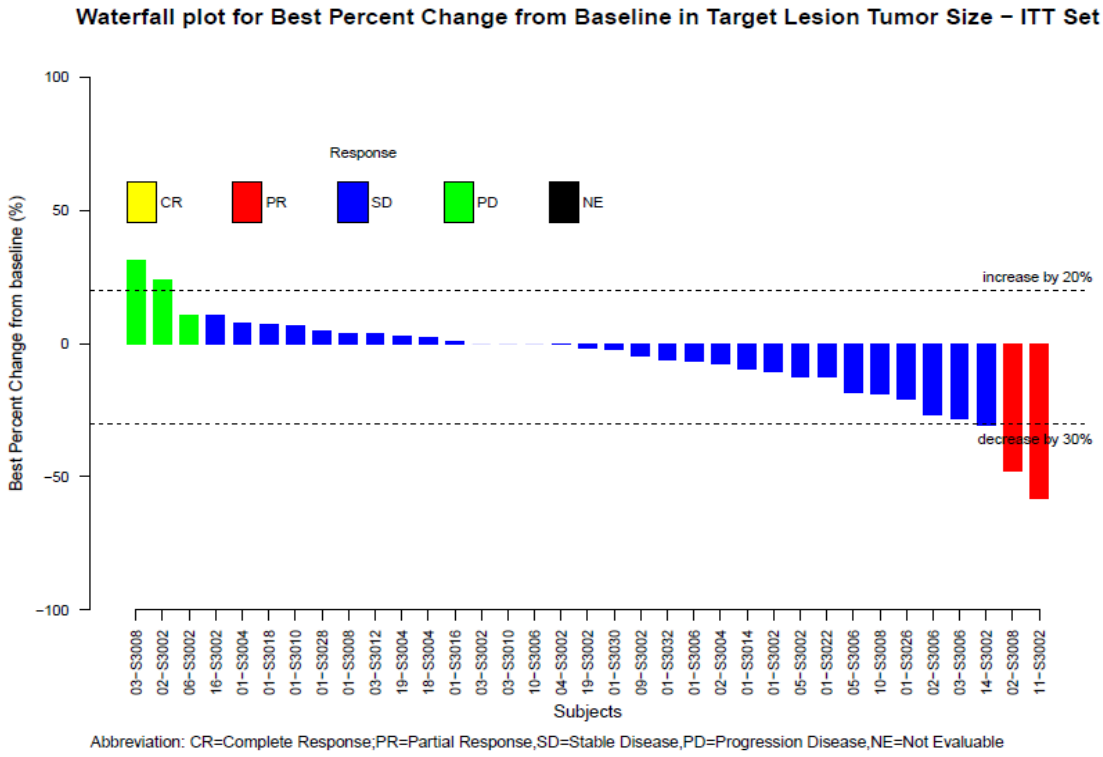

Figure 3. Waterfall plot for Best Percent Change from Baseline in Target Lesion Tumor Size – ITT Set

The plot procedure:

```
#plot
par(mar=c(9,6,6,6))  #bottom right top left
barplot(final$min,
        col=ifelse(final$BOR=="CR","yellow",
                   ifelse(final$BOR=="PR","red",
                          ifelse(final$BOR=="SD","blue",
                                 ifelse(final$BOR=="PD","green",
                                        ifelse(final$BOR=="NE","black",""))))),
        border=NA, space=0.5,
        main = "\n\nWaterfall plot for Best Percent Change from Baseline in Target Lesion Tumor Size - ITT Set\n\n"
        cex.main=1.3,cex.sub=1.3,

        ylim=c(-100,100),ylab="Best Percent Change from baseline (%)", cex.lab=1,

        names.arg=final$x,
        cex.names=0.8, cex.axis=0.8, axis.lty=1,las=2
        )
legend("topleft",horiz=TRUE,title="Response",
       legend=c("CR","PR","SD","PD","NE"),
       fill=c("Yellow","Red","Blue","Green","Black"),
       bty = "n",cex=0.8,y.intersp=0.7,x.intersp=0.1,text.width=2)
mtext(text="Subjects",side=1,outer=FALSE,line=4.5,cex=1)
abline(h=20,lty=2)
mtext(text="increase by 20%",side=1,at=55,line=-17,cex=0.8,col="black",adj=1)
abline(h=-30,lty=2)
mtext(text="decrease by 30%",side=1,at=55,line=-9,cex=0.8,col="black",adj=1)
mtext(text="Abbreviation: CR=Complete Response;PR=Partial Response,SD=Stable Disease,
           PD=Progression Disease,NE=Not Evaluable",side=1,line=6,adj=0,cex=0.9,outer=FALSE)
mtext(text="",side=1,line=7,adj=0,cex=0.9,outer=FALSE)
mtext(text="",side=1,line=8,adj=0,cex=0.9,outer=FALSE)
```

i. Use ifelse function to assign color
ii. Names.arg is to display x variable value
iii. Legend to populate legend
iv. Abline + mtext can draw reference line and add text.

SWIMMER PLOT

Swimmer Plot in oncology studies is desired to show multiple information of each subjects treated, such as response or treatment duration and subject status. X-axis is time, the length of bar stands for the treatment duration of each subject. Treatment duration is composed by response durations, which are colored by different color and arranged by time. After the bar ends, an extra symbol will be added to indicate subjects still on treatment and full text for all. As showed in Figure 4.

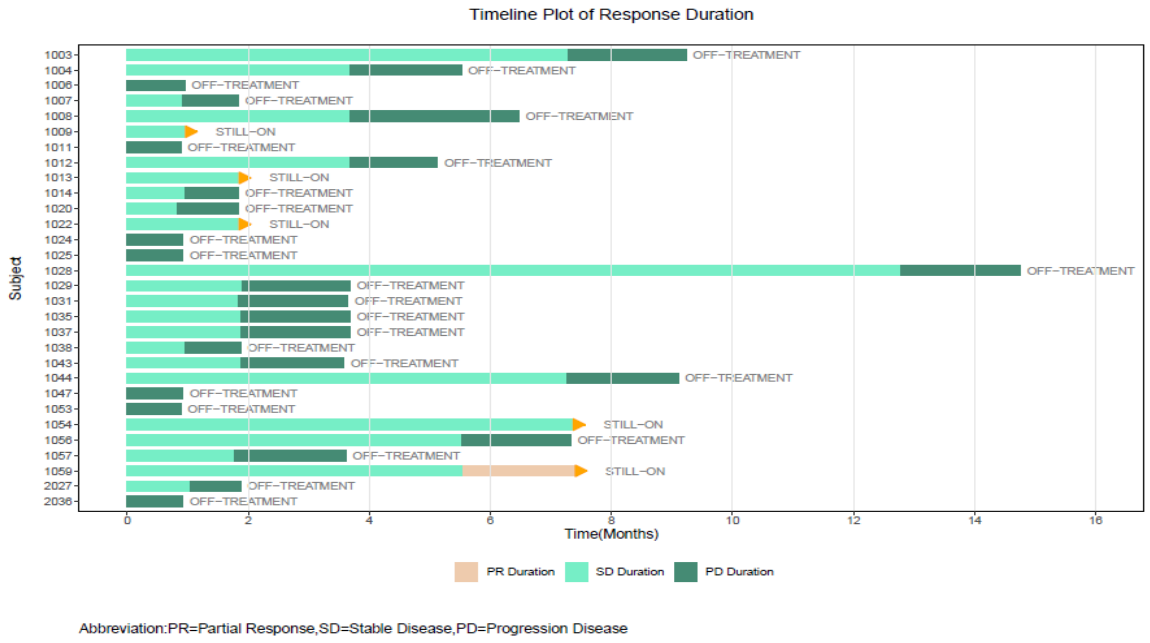

Figure 4. Timeline Plot of Response Duration

In contrast, Figure 5 illustrates response duration by marking symbols, and distinguish treatment by different bar color.

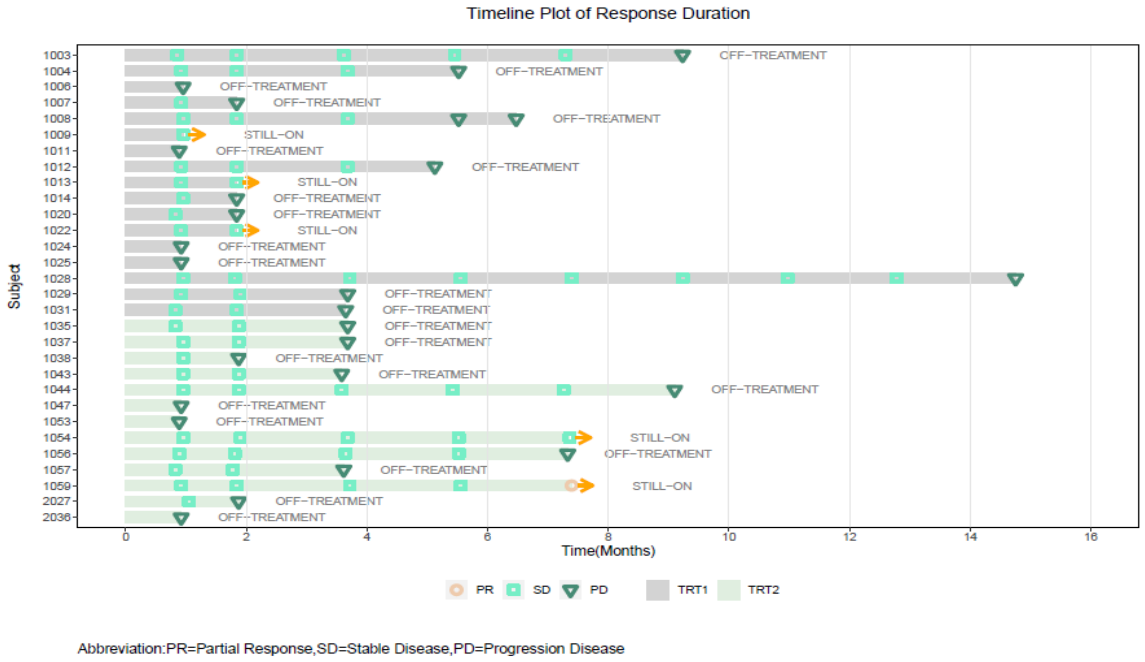

Figure 5. Timeline Plot of Response Duration

Key variables:

- SUBJECT: subject id
- DURATION: duration of each response
- AVALC: response
- yend: total duration of subject still in treatment
- textend: total duration of each subject

Procedure:

```
ggplot(time,aes(SUBJECT,DURATION))+
  labs(title = "\n\nTimeline Plot of Response Duration\n",
       caption ="\n\nAbbreviation:PR=Partial Response,SD=Stable Disease,PD=Progression Disease'
  xlab("Subject\n")+ylab("Time(Months)")+
  coord_flip()+
  scale_y_continuous(breaks=seq(0,16,by=2),limits=c(0,16))+
  geom_bar(aes(x=reorder(SUBJECT,-as.numeric(SUBJECT)),y=DURATION,fill=AVALC),
           stat="identity",width=0.7,position=position_stack(reverse=FALSE))+
  scale_fill_manual(name="",values=c("aquamarine4","peachpuff2","aquamarine2"),
                    labels=c("PR Duration","SD Duration","PD Duration"),
                    breaks=c("PR","SD","PD"))+
  geom_segment(aes(x=SUBJECT,xend=SUBJECT,y=yend+0.1,yend=yend+0.2),
               lineend="butt",size=0.7,color="orange",
               arrow=arrow(length=unit(0.13,"inches"),type="closed"))+
  geom_text(aes(label=STATUS,y=textend+1,x=SUBJECT),color="gray50",size=3)+
  geom_hline(yintercept=c(2,4,6,8,10,12,14,16),linetype=1,size=0.5,color="gray90")+
  theme(plot.margin=margin(.75,.5,.75,.5,"cm"),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        panel.border = element_rect(colour = "black", fill=NA, size=0.5),
        plot.title = element_text(size=13,hjust=0.5),
        plot.caption = element_text(size=11,hjust=0),
        legend.position="bottom")
```

i. geom_bar() is to add bar; x is subject in ascending order, y is duration and duration filled in different default color according to AVALC.

ii. Stat="identity" means bar height equal duration; position_stack is to stack next bar in the top of each bar, reverse=FALSE means do not reverse stack order.

iii. coord_flip is to reverse x and y coordinate

iv. scale_fill_manual is to change fill color manually and add legend.

v. geom_segment() is to add line segment; aes define coordinate position, arrow is to specify arrow heads, length in 0.13 inches and in closed trianle.

vi. geom_text() is to add gray text at arrow end

vii. geom_hline() is to add horizontal reference line

viii. theme() is to define margin, background, font, fontsize and all the other aesthetics of plot content.

Figure 5 fill duration according to treatment group and add symbol instead：

```
geom_bar(aes(x=reorder(SUBJECT,-as.numeric(SUBJECT)),y=DURATION,fill=TRTGRP),stat="identity"
         ,width=0.7,position=position_stack(reverse=TRUE))+
geom_point(data=time,aes(SUBJECT,TIME,shape=AVALC,color=AVALC),size=2,stroke=2)+
```

CONCLUSION

Graphical presentation in oncology study gives us a broad vision of data and is full of challenge. R can be an efficient and smart tool in help whether in generation or validation. Graphics and grid provides methods for statistical graphics and there're many other packages developed for oncology studies. The methods used in this paper can provide some thoughts when you get start.

REFERENCES

Emily C.Zabor. 2018. *"Survival Analysis in R".*

https://www.emilyzabor.com/tutorials/survival_analysis_in_r_tutorial.html

Jyothi. 2016. "*Forest Plot (With Horizontal Bands)".*

https://www.r-bloggers.com/forest-plot-with-horizontal-bands/

Max Gordon, Thomas Lurnley. 2019. *"Package 'forestplot'".*

https://cran.r-project.org/web/packages/forestplot/forestplot.pdf#page=3&zoom=100,0,89

Stacey Phillips. 2017. *"Creating a survival-swimmer plot in R".*

http://rstudio-pubs-static.s3.amazonaws.com/389060_a0ea812d8f8d490393bd1c65a6dcddef.html

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact the author at:

Linqing Wang
Hutchison Medi Pharma
13162570753
hailey0907@163.com