

Paper DV05

# Generating Statistical Figures for Regulatory Submission: Comparison of SAS vs. R

Girija Manohar Javvaji and Sivasankar Konda  
Labcorp Drug Development

## Abstract

Statistical graphs are the essential component of the clinical trials that are included in the submission package to the regulatory agencies. As we all know, R language is being adopted in pharmaceutical companies along with the traditional SAS (which is still in use). In this paper, we would like to discuss some of the prominent clinical graphs such as Forest plot, Kaplan-Meier plot, Waterfall plot, Swimmer Lane plot and Multipanel plot using both SAS and R. A side-by-side comparison of the above graphs in SAS and R using their respective syntax, key differences and features that are available in both languages are highlighted in this paper.

## Introduction

When we have the knowledge and software available for both SAS and R, we have the luxury to choose them based on the graphs that we need to plot. With the upgraded versions of Graphic Template Language (GTL), SAS is getting user friendly to accomplish the desired high-quality clinical graphs, while R is open source and easy to adapt. The ggplot2 package from R is a robust library that has the features to plot most of the clinical graphs. Additionally, there are many dedicated R packages to create specific graphs such as 'survival' for Kaplan-Meier and 'forestplot' package for Forest plots, and there is scope to create more advanced customized plots in the future. In this paper we will take a closer look at the following types of graphs.

- Forest plot
- Kaplan-Meier
- Waterfall/Swimmer Lane plot
- Multipanel plot

We are using GTL for SAS, which might be embedded in macros. For R, we are covering all the graphs using mostly the ggplot2 package. We will distinguish outputs using the two factors of Time and Quality.

```

data final;
length subgroup $55 nleft nright $8 diff $30 diffrem $30;
infile datalines dlm=', ' ;
input subgroup & $ nleft nright diff & $ diffrem & $;
cards;
Baseline Headache Status: Presence
Drug B vs Placebo , 8 , 12, 27.0% (-70.1%, 30.2%)
Drug C vs Placebo , 9 , 12, -15.9% (-60.7%, 32.0%)
Drug A + Drug B vs Placebo , 18 , 12, -31.9% (-66.7%, 10.0%)
Drug A + Drug C vs Placebo , 17 , 12, -20.2% (-58.5%, 1.1%)
Drug B + Drug C vs Placebo , 21 , 12, -17.0% (-55.6%, 25.7%)
Drug A + Drug B vs Drug B , 20 , 12, -11.9% (-42.5%, 24.9%)
Drug B + Drug C vs Drug B , 22 , 12, 2.0% (-25.8%, 35.8%)
Drug A + Drug C vs Drug C , 19 , 13, -15.5% (-48.4%, 17.4%)
Drug B + Drug C vs Drug C , 20 , 13, -3.3% (-37.7%, 33.1%)
Baseline Headache Status: Absence
Drug B vs Placebo , 5 , 9, -53.9% (-113%, 27.8%)
Drug C vs Placebo , 6 , 9, -45.9% (-97.1%, 6.4%)
Drug A + Drug B vs Placebo , 11 , 9, -21.6% (-71.4%, 28.1%)
Drug A + Drug C vs Placebo , 10 , 9, -42.9% (-86.0%, 0.3%)
Drug B + Drug C vs Placebo , 11 , 9, -24.7% (-71.7%, 22.3%)
Drug A + Drug B vs Drug B , 9 , 5, 32.2% (-42.6%, 67.0%)
Drug B + Drug C vs Drug B , 11 , 5, 18.2% (-34.1%, 70.5%)
Drug A + Drug C vs Drug C , 8 , 6, 0.0% (-38.0%, 38.0%)
Drug B + Drug C vs Drug C , 11 , 6, 18.2% (-24.1%, 60.4%)
run;

```

```

entrytitle halign=center 'Figure 1. Forest Plot with Advanced Options' / textattrs = (weight=normal size=8);
discreteattrmap name="id" / ignorecase = true;
  value '1' / textattrs = (color=black size=10 weight=bold);
  value '2' / textattrs = (color=black size=7 weight=normal);
enddiscreteattrmap ;

discreteattrvar attrvar=sid var=id attrmap="id" ;

layout overlay / xaxisopts=(linearopts=(tickvaluesequence=(start=&lv end=&uv increment=&inc)
  viewmin=&lv viewmax=&uv) display=(ticks line tickvalues) lineextent=data
  %if &xlabelb = Y %then %do; label="&&xlabbb" %end;))
  yaxisopts=(reverse=true display=none linearopts=(viewmin=1
    tickvaluesequence=(start=1 end=56 increment=1)))
  x2axisopts=(display=(line) %if &xlabelt = Y %then %do; label="&xlabt" %end;));

referenceline y=ref / lineattrs=(thickness=12 color=cxf0f0f0);

  highlowplot y=&yidv low=&vlcl high=&vucl /highcap=highcap lowcap=lowcap lineattrs=(pattern=1 thickness=0.5);

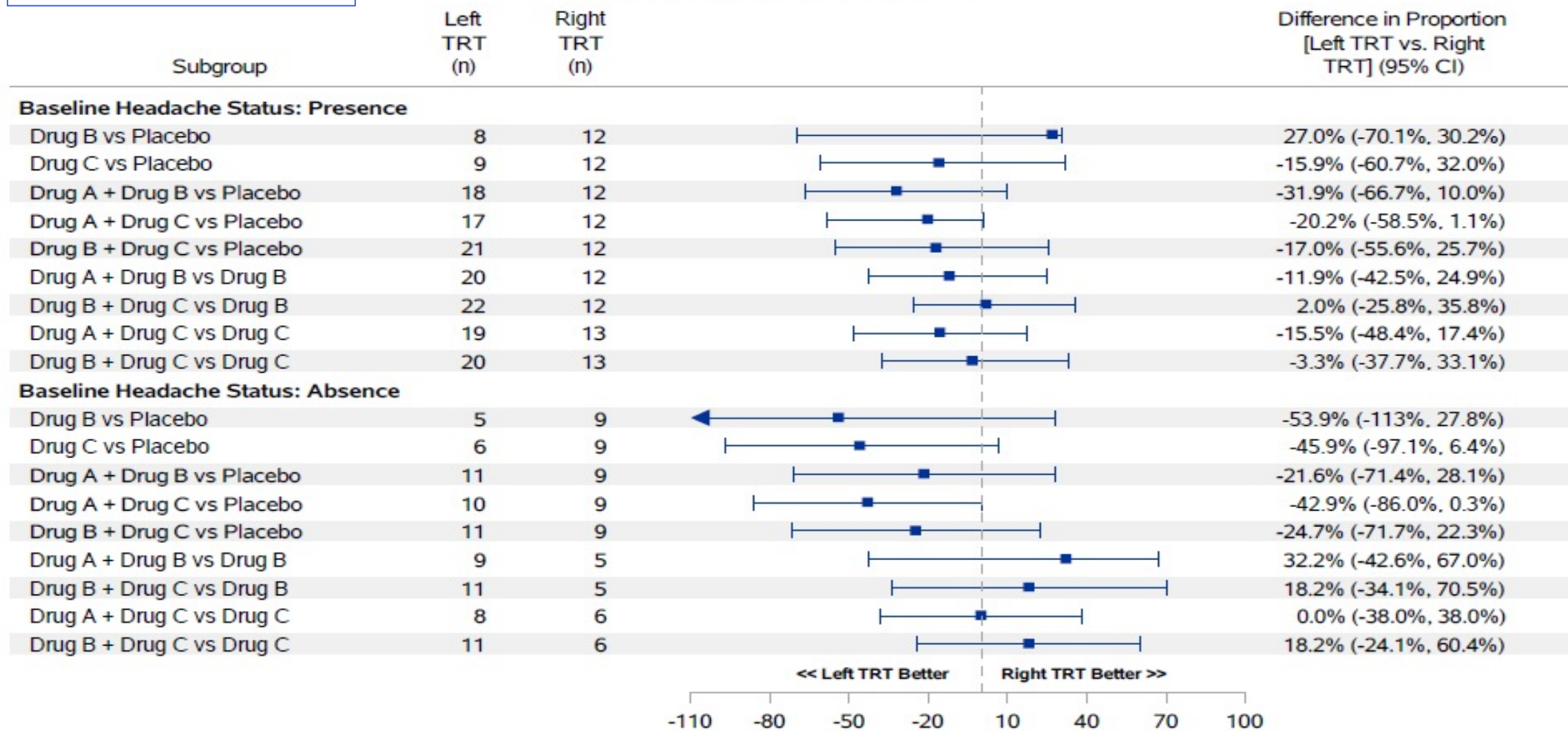
scatterplot y=&yidv x=&mv / markerattrs=(symbol=&msymbol) ;
scatterplot y=&yidv x=&mv / markerattrs=(size=0) xaxis=x2;
referenceline x=0 / lineattrs=(pattern=&reflinep);

%if &textyn eq Y %then %do;
textplot x=x1 y=&yidv text=text / position=center contributeoffsets=none textattrs=(weight=bold);
%end;

%do j=1 %to &cnt;
  %if &&rl&j = left %then %do;
    innermargin / align=&&rl&j ;
    %if &&var&j ^=empty %then %do;
      axistable value=&&var&j y=&yidv/ textgroup=sid labelattrs=(size=8)
        valuejustify=&&jrl&j labeljustify=&&ljl&j valueattrs=(size=8);
    %end;
    %else %if &&var&j = empty %then %do;
      axistable value=&&var&j y=&yidv/ labelattrs=(size=2 color=CXFFFFFF)
        valuejustify=&&jrl&j labeljustify=&&ljl&j valueattrs=(size=2);
    %end;
  endinnermargin;

```

Figure 1. Forest Plot with Advanced Options



```

library(ggplot2)
library(tidyr)
library(grid)
library(gridExtra)
library(forcats)

fp <- fp[order(fp$obs),]

p <- ggplot(fp, aes(x = ldiff, y = obs)) +
  geom_hline(aes(yintercept = gray_id),colour = "gray95",size = 7,show.legend=F) +
  geom_errorbarh(aes(xmax = ucl, xmin = lcl, height = .2),color = "blue") +
  geom_errorbarh(data = subset(fp, fp$lcl <= -110), aes(xmax = ucl, xmin = -110, height = .2),color = "blue") +
  geom_point(shape = 22, fill = "blue") + geom_hline(yintercept = 0.6, linetype = 1) +
  geom_point(data = subset(fp, fp$lcl <= -110), aes(x = -109, y = obs), shape = 18, size=3, color = "blue") +
  geom_segment(x= 1,y = -1,xend = 1,yend = -50, linetype = 2) +
  geom_text(aes(y = 21,x = -40,label = "<< Left TRT Better", fontface = "plain"),size=3) +
  geom_text(aes(y = 21,x = 40,label = "Right TRT Better >>", fontface = "plain"),size=3) +
  xlab(" ") + ylab(" ") + theme_classic() + scale_y_continuous(trans = "reverse") +
  scale_x_continuous(limits = c(-110, 100), breaks = seq(-110,100,30)) +
  theme(axis.text.y = element_blank(), axis.title.y = element_blank(), axis.ticks.y = element_blank(),
        axis.line.y = element_blank()) + theme(plot.margin = margin(2.5, -1, -7.5, -9))

data_left <- ggplot(fp, aes(y = obs)) +
  geom_hline(aes(yintercept = gray_id),colour="gray95",size=7,show.legend=F) +
  geom_text(data=subset(fp, obs %in% c(1,11) ),aes(x = 0, label = subgroup, fontface="bold"), hjust = 0) +
  geom_text(data=subset(fp,! (obs %in% c(1,11)) ),aes(x = 0, label = subgroup), hjust = 0) +
  geom_text(aes(x = 30, label = nleft)) + geom_text(aes(x = 45, label = nright), hjust = 1) +
  geom_hline(yintercept = 0.6, linetype = 1) + scale_colour_identity() +
  scale_y_continuous(trans="reverse") + theme_void() + theme(plot.margin = margin(5, -1, 35, 1))

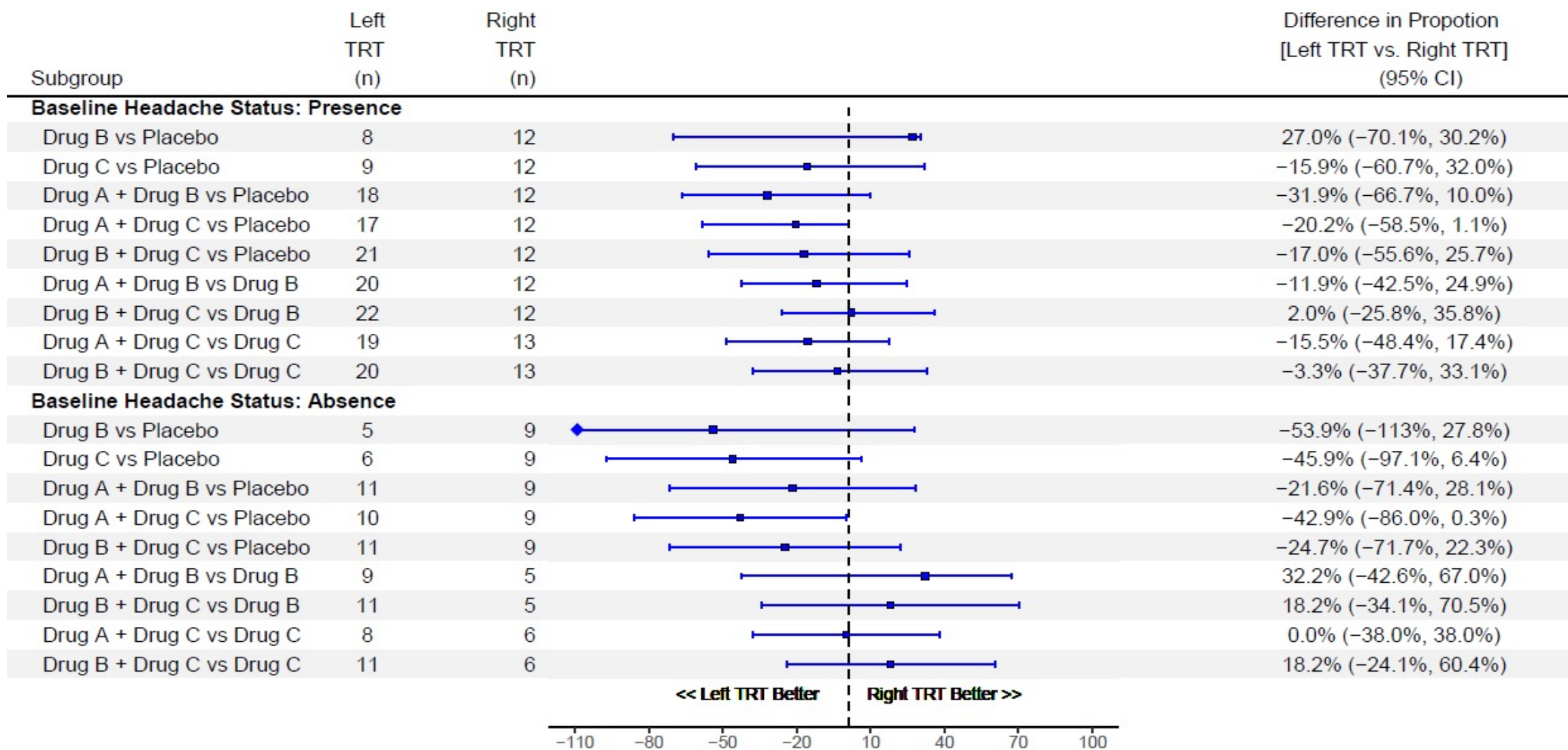
data_right <- ggplot(fp, aes(y = obs)) +
  geom_hline(aes(yintercept = gray_id),colour="gray95",size=7,show.legend=F) +
  geom_text(aes(x = 0, label = diff), hjust=0.5) +
  scale_colour_identity() + scale_y_continuous(trans="reverse") +
  geom_hline(yintercept = 0.6, linetype = 1) + theme_void() +
  theme(plot.margin = margin(5, 0, 35, 0))

fpfinal <- grid.arrange(data_left,p,data_right,top = "Figure 1. Forest Plot with Advanced Options", ncol = 3)

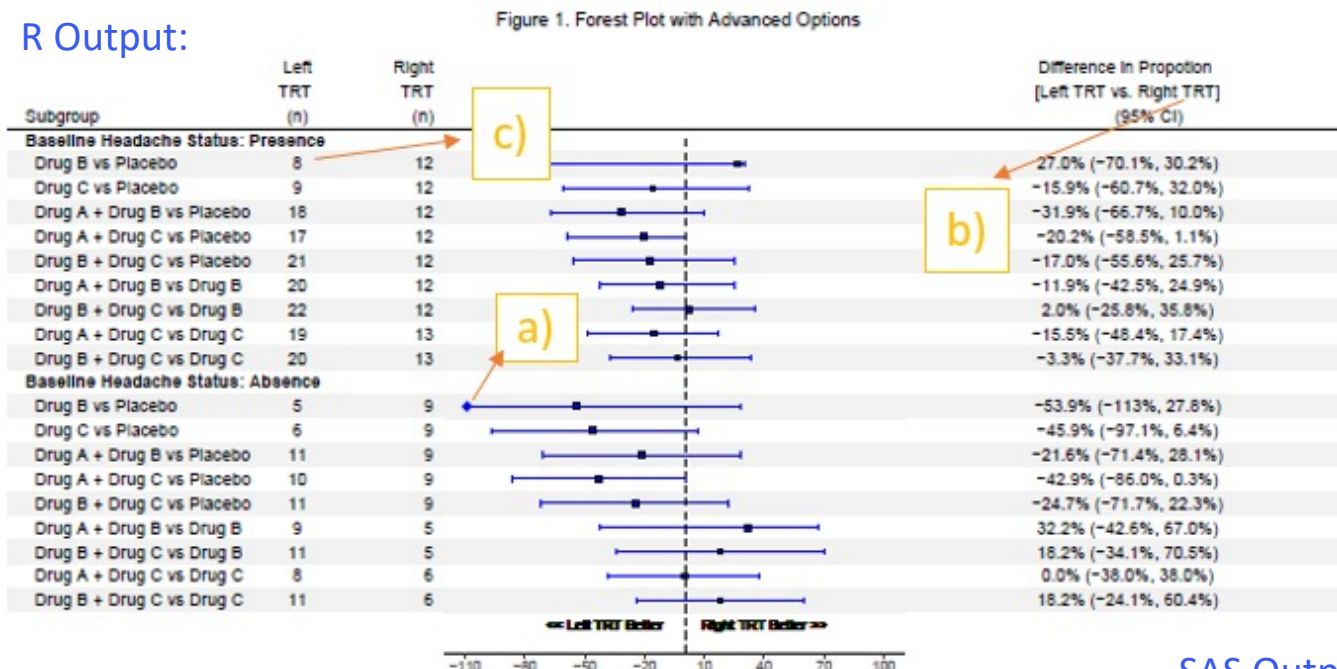
ggsave(fpfinal, file="//forestplot.pdf", width = 12, height=6, dpi=300)

```

Figure 1. Forest Plot with Advanced Options



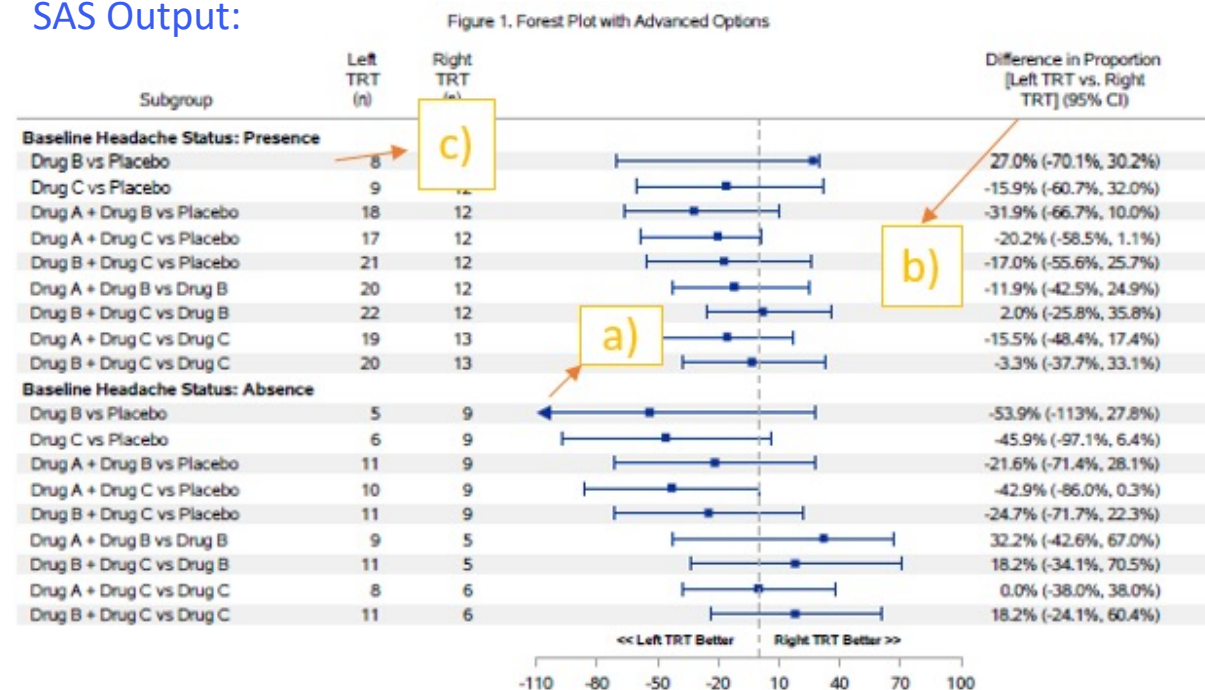
## R Output:



**Differences:** Some of the observed limitations are indicated below.

- a) SAS is using the left filled triangle, where R doesn't have that symbol so we have to go with a diamond symbol in the plot (for values that are going beyond the minimum value, i.e., -110).
- b) SAS is wrapping the labels of the last column, which is not the problem with R, as we are passing the values as row values rather than labels of x-axis.
- c) In SAS, every column is allotted some space so the second column will start only after the first column allotted space. This is not the case with R, which can help us save some space.

## SAS Output:



**Time:** To create, SAS took less time when compared to R.

And also when SAS code is embedded in sophisticated macros can save more time.

**Quality:** Almost similar but SAS has an edge over R, as we are creating different parts of the output. Every time and in the end we are attaching all of them to get the desired output, which might reduce the quality.

## Data for Kaplan Meier and Procedure Used

```
proc format;
  invalue trtfrm
    'AML-High Risk' =1
    'AML-Low Risk' =2
    'ALL'          =3;
run;

data final;
  set sashelp.bmt;
  subjid=_n_;  aval=(t/365.25);
  avalM=(t/365.25/12);  censor=status;
  trtn=input(group,trtfrm.);
run;

ods output ProductLimitEstimates=dsset;
proc lifetest data=final;
  time aval*censor(0);
  strata trtn;
run;
```

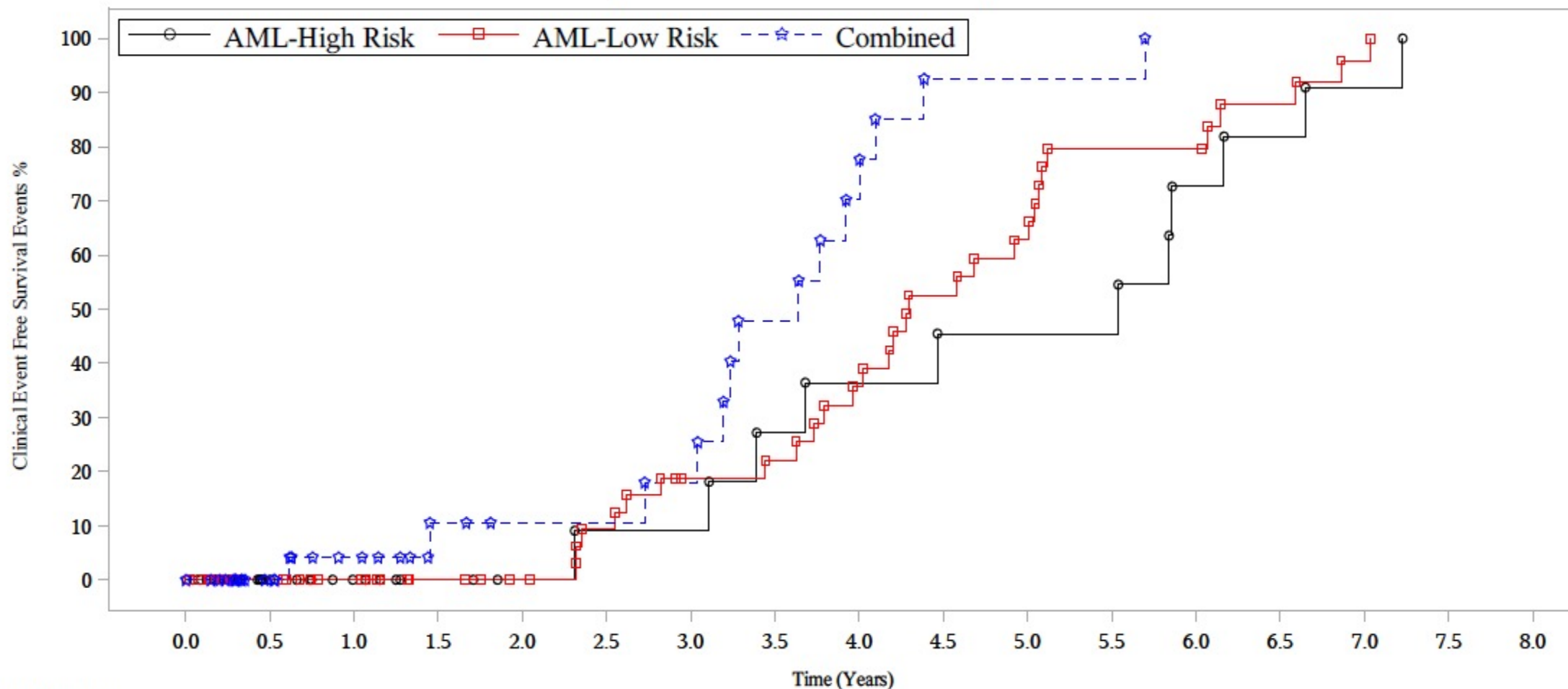
## Waterfall Plot: SAS GTL Template

```
proc template;
  define statgraph srmTemp;
    begingraph;
      entrytitle halign=center "Inverted Kaplan Meier Plot with Advanced Options";
      entryfootnote halign=left "Footnote1: Annotation colors match the step plot colors" / textattrs=(style=normal) pad=(top=3px);
      layout lattice / columndatarange=unionall rowweights=&rowweights.;
      layout overlay / walldisplay=(outline)
        yaxisopts=(offsetmin=0.05 offsetmax=0.05
          linearopts=(tickvaluesequence=(start=0 end=100 increment=10) tickvaluepriority=true) label=" ")
        xaxisopts=(label="&xlabel" labelattrs=(size=8 weight=Normal) offsetmin=0.05 offsetmax=0.03
          linearopts=(tickvaluesequence=(start=&xlv end=&xuv increment=&xinc)));
      stepplot y=survival x=aval/group=trta display=(markers) name="step" markerattrs=(size=7);
    annotate;
    discretelegend "step" / location=inside across=&lablcols valueattrs=(size=12) autoalign=(topleft) border=TRUE borderattrs=(thickness=1 color=black);
    endlayout;

    layout overlay;
      entry halign=left textattrs=(size=9.5) 'N at Risk (Events)';
    endlayout;

    layout overlay / walldisplay=none border=false yaxisopts=(display=none)
      xaxisopts=(display=none offsetmin=0.05 offsetmax=0.03
        linearopts=(tickvaluesequence=(start=&xlv end=&xuv increment=&xinc) tickvaluepriority=true));
      axistable value=left_fail x=aval/class=trtb colorgroup=trtb labelattrs=(size=8) valuejustify=right labeljustify=right valueattrs=(size=8);
    endlayout;
  endlayout;
endgraph;
end;
run;
```

Inverted Kaplan Meier Plot with Advanced Options



N at Risk (Events)

|                |        |        |        |        |        |        |        |        |         |         |         |        |        |        |        |        |        |
|----------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| AML-High Risk: | 45 (0) | 23 (0) | 17 (0) | 13 (0) | 11 (0) | 10 (1) | 10 (1) | 8 (3)  | 7 (4)   | 6 (5)   | 6 (5)   | 6 (5)  | 3 (8)  | 2 (9)  | 1 (10) | 0 (11) | 0 (11) |
| AML-Low Risk:  | 54 (0) | 47 (0) | 42 (0) | 36 (0) | 33 (0) | 29 (3) | 24 (6) | 23 (7) | 19 (11) | 14 (16) | 11 (19) | 6 (24) | 6 (24) | 3 (26) | 1 (28) | 0 (29) | 0 (29) |
| Combined:      | 38 (0) | 26 (0) | 20 (1) | 14 (2) | 12 (2) | 12 (2) | 11 (3) | 7 (7)  | 4 (10)  | 1 (13)  | 1 (13)  | 1 (13) | 0 (14) | 0 (14) | 0 (14) | 0 (14) | 0 (14) |

```

library(ggplot2)
library(patchwork)

surv <- read.table("clipboard",header=TRUE,sep="\t")

km <- ggplot(surv, aes(group=trta, y=SURVIVAL,x=aval,colour=trta,linetype=trta)) +
  geom_step() + geom_point(aes(shape=as.factor(trta))) +
  scale_shape_manual("", values=c(1,0,11)) + scale_color_manual("", values=c("black","red","blue")) +
  scale_linetype_manual("", values=c(1,1,2)) + scale_y_continuous(breaks=seq(0,100,10),limits=c(0,100)) +
  scale_x_continuous(breaks=seq(0,8,0.5), limits=c(0,8)) +
  xlab("\nTime (Years)") + ylab("Clinical Events Free Survival Events%") +
  labs(subtitle = "Inverted Kaplan Meier Plot with Advanced Options") + theme_bw() +
  theme(legend.justification=c(0,1), legend.position=c(0.01, 0.99),
        legend.direction = "horizontal", legend.text.align = 1,
        plot.subtitle = element_text(hjust = 0.5),
        legend.box.margin = margin(t = 0, r = 0, b = 0, l = 0, unit = "pt"),
        legend.background = element_rect(linetype = 1, size = 0,color = 1),
        legend.key = element_blank(),legend.spacing.x = unit(0.3, 'cm'),
        panel.grid = element_blank(), text=element_text(family="Times"))

atrisk <- ggplot(subset(surv, !is.na(left_fail)), aes(color=trtb)) +
  geom_text(aes(label=left_fail, x=aval,y=trtb), show.legend=F, size=3) +
  scale_x_continuous(breaks=seq(0,8,0.5), limits=c(0,8)) +
  scale_color_manual("", values=c("black","red","blue")) +
  ylab(" ") + xlab(" ") + theme_bw() + scale_y_discrete(limits = rev) +
  labs(subtitle = "N at Risk (Events)") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.line= element_blank(), axis.text.x = element_blank(), axis.ticks = element_blank(),
        panel.border=element_blank(), plot.title.position = "plot",
        plot.subtitle = element_text(hjust = 0.007,vjust = -1),axis.text.y = element_text(color="black"),
        plot.caption = element_text(hjust = 0, family="Times"),
        text=element_text(family="Times"))

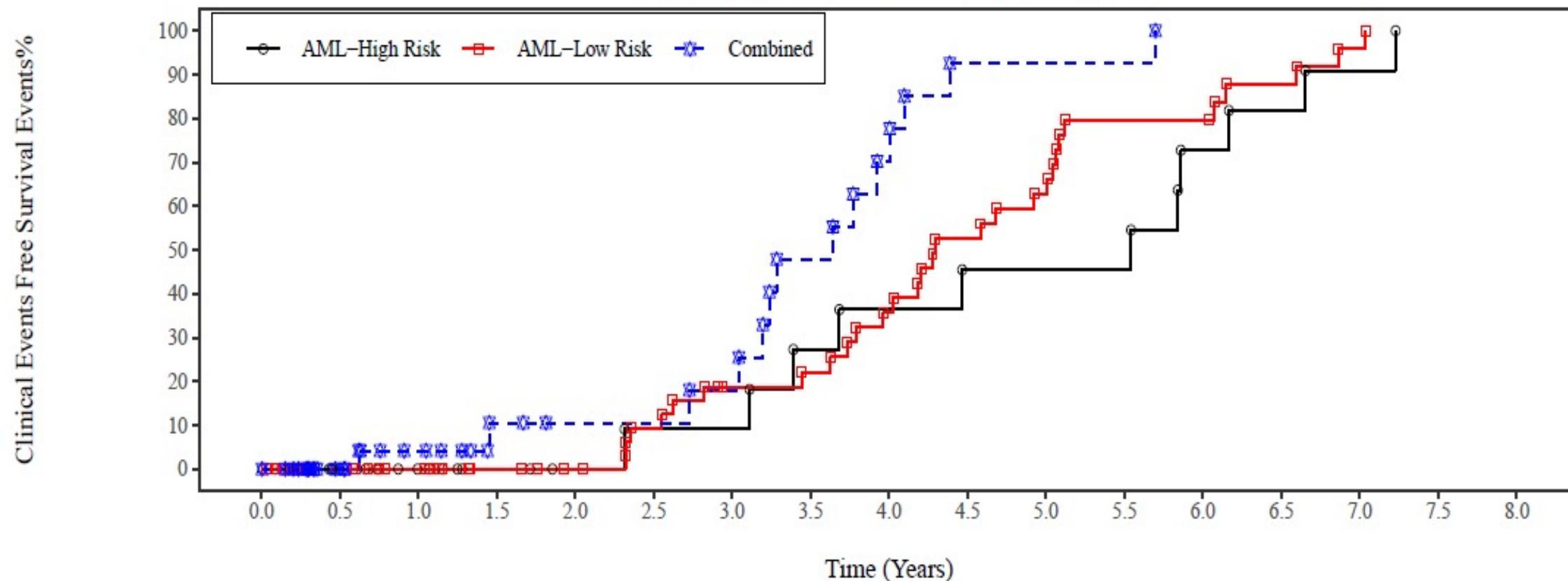
km2 <- km + atrisk + plot_layout(ncol=1, heights=c(1,0.20))

ggsave(km2, file="//KMplot.pdf", width = 10, height=5, dpi=300)

```

Generated Using R ggplot2

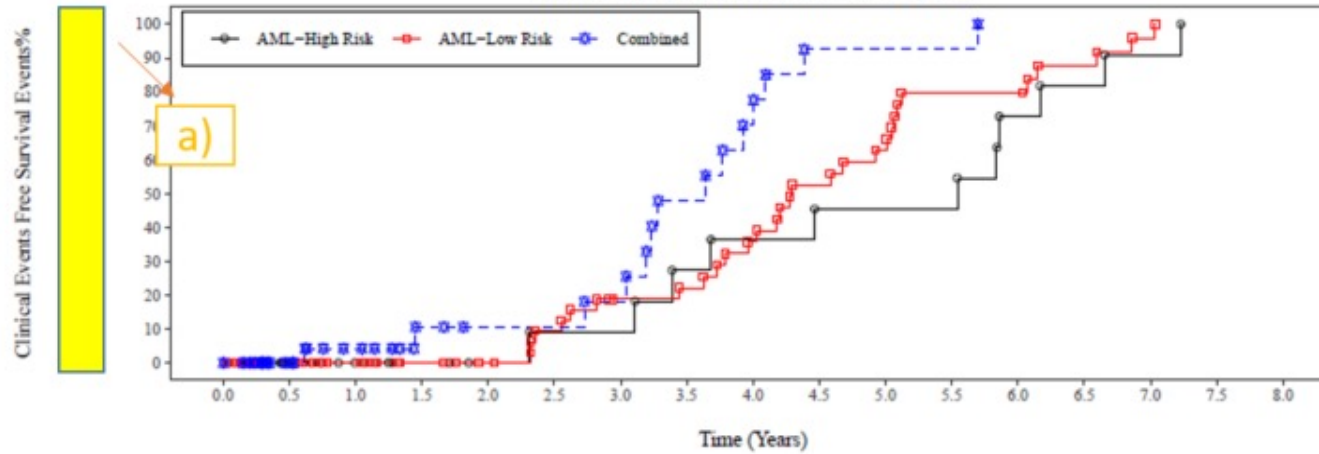
Inverted Kaplan Meier Plot with Advanced Options



N at Risk (Events)

|                |        |        |        |        |        |        |        |        |         |         |         |        |        |        |        |        |        |
|----------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| AML-High Risk: | 45 (0) | 23 (0) | 17 (0) | 13 (0) | 11 (0) | 10 (1) | 10 (1) | 8 (3)  | 7 (4)   | 6 (5)   | 6 (5)   | 6 (5)  | 3 (8)  | 2 (9)  | 1 (10) | 0 (11) | 0 (11) |
| AML-Low Risk:  | 54 (0) | 47 (0) | 42 (0) | 36 (0) | 33 (0) | 29 (3) | 24 (6) | 23 (7) | 19 (11) | 14 (16) | 11 (19) | 6 (24) | 6 (24) | 3 (26) | 1 (28) | 0 (29) | 0 (29) |
| Combined:      | 38 (0) | 26 (0) | 20 (1) | 14 (2) | 12 (2) | 12 (2) | 11 (3) | 7 (7)  | 4 (10)  | 1 (13)  | 1 (13)  | 1 (13) | 0 (14) | 0 (14) | 0 (14) | 0 (14) | 0 (14) |

## R Output:



|                    |  |        |        |        |        |        |        |        |        |         |         |         |        |        |        |        |        |        |
|--------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| N at Risk (Events) |  | 45 (0) | 23 (0) | 17 (0) | 13 (0) | 11 (0) | 10 (1) | 10 (1) | 8 (3)  | 7 (4)   | 6 (5)   | 6 (5)   | 6 (5)  | 3 (8)  | 2 (9)  | 1 (10) | 0 (11) | 0 (11) |
| AML-High Risk:     |  | 45 (0) | 47 (0) | 42 (0) | 38 (0) | 33 (0) | 29 (3) | 24 (6) | 23 (7) | 19 (11) | 14 (16) | 11 (19) | 6 (24) | 6 (24) | 3 (26) | 1 (28) | 0 (29) | 0 (29) |
| AML-Low Risk:      |  | 38 (0) | 26 (0) | 20 (1) | 14 (2) | 12 (2) | 12 (2) | 11 (3) | 7 (7)  | 4 (10)  | 1 (13)  | 1 (13)  | 1 (13) | 0 (14) | 0 (14) | 0 (14) | 0 (14) | 0 (14) |
| Combined:          |  |        |        |        |        |        |        |        |        |         |         |         |        |        |        |        |        |        |

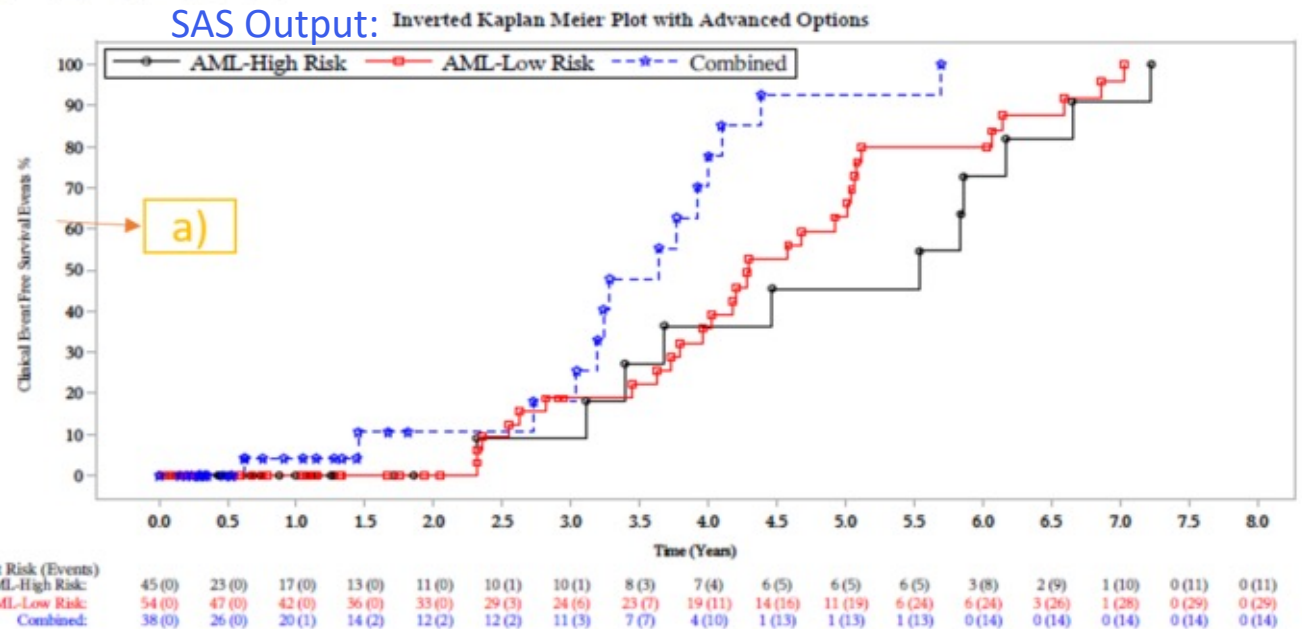
**Time:** To create, R took less time when compared to SAS. However, SAS code can be embedded in the sophisticated macros, which can be handy when we are outputting similar outputs and it's well worth the time that we spending creating the macros.

**Quality:** Almost similar.

**Differences:** Some of the observed limitations are indicated below.

a) Some space is produced in the R output between y-axis and its label (and that space is indicated in the graph with a rectangular box filled with yellow)

## SAS Output:



|                    |  |        |        |        |        |        |        |        |        |         |         |         |        |        |        |        |        |        |
|--------------------|--|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| N at Risk (Events) |  | 45 (0) | 23 (0) | 17 (0) | 13 (0) | 11 (0) | 10 (1) | 10 (1) | 8 (3)  | 7 (4)   | 6 (5)   | 6 (5)   | 6 (5)  | 3 (8)  | 2 (9)  | 1 (10) | 0 (11) | 0 (11) |
| AML-High Risk:     |  | 45 (0) | 47 (0) | 42 (0) | 36 (0) | 33 (0) | 29 (3) | 24 (6) | 23 (7) | 19 (11) | 14 (16) | 11 (19) | 6 (24) | 6 (24) | 3 (26) | 1 (28) | 0 (29) | 0 (29) |
| AML-Low Risk:      |  | 38 (0) | 26 (0) | 20 (1) | 14 (2) | 12 (2) | 12 (2) | 11 (3) | 7 (7)  | 4 (10)  | 1 (13)  | 1 (13)  | 1 (13) | 0 (14) | 0 (14) | 0 (14) | 0 (14) | 0 (14) |
| Combined:          |  |        |        |        |        |        |        |        |        |         |         |         |        |        |        |        |        |        |

**Barcharts:** We will discuss about the waterfall plot.

For data, we are going to use the following dummy data.

```
proc format;
value trtan
  1='Drug A'
  2='Drug B'
  3='Drug C'
  4='Drug A + Drug B'
  5='Drug A + Drug C'
  6='Drug B + Drug C'
  7='Placebo'
;
run;

data final;

  do trt01an=1 to 7;
    do i=1 to 3;

      pchg=85+trt01an*2*i;
      subjid+1;
      output;

      pchg=-100+trt01an*2*i;
      subjid+1;
      output;

    end;
  end;

run;
```

```

proc template;
  define statgraph srmBarchart;
    begingraph / border=false;
      entrytitle halign=center 'Waterfall plot made easy with advanced options';
      layout lattice/ rows=2 rowweights=(0.1 0.9) rowgutter=3px;

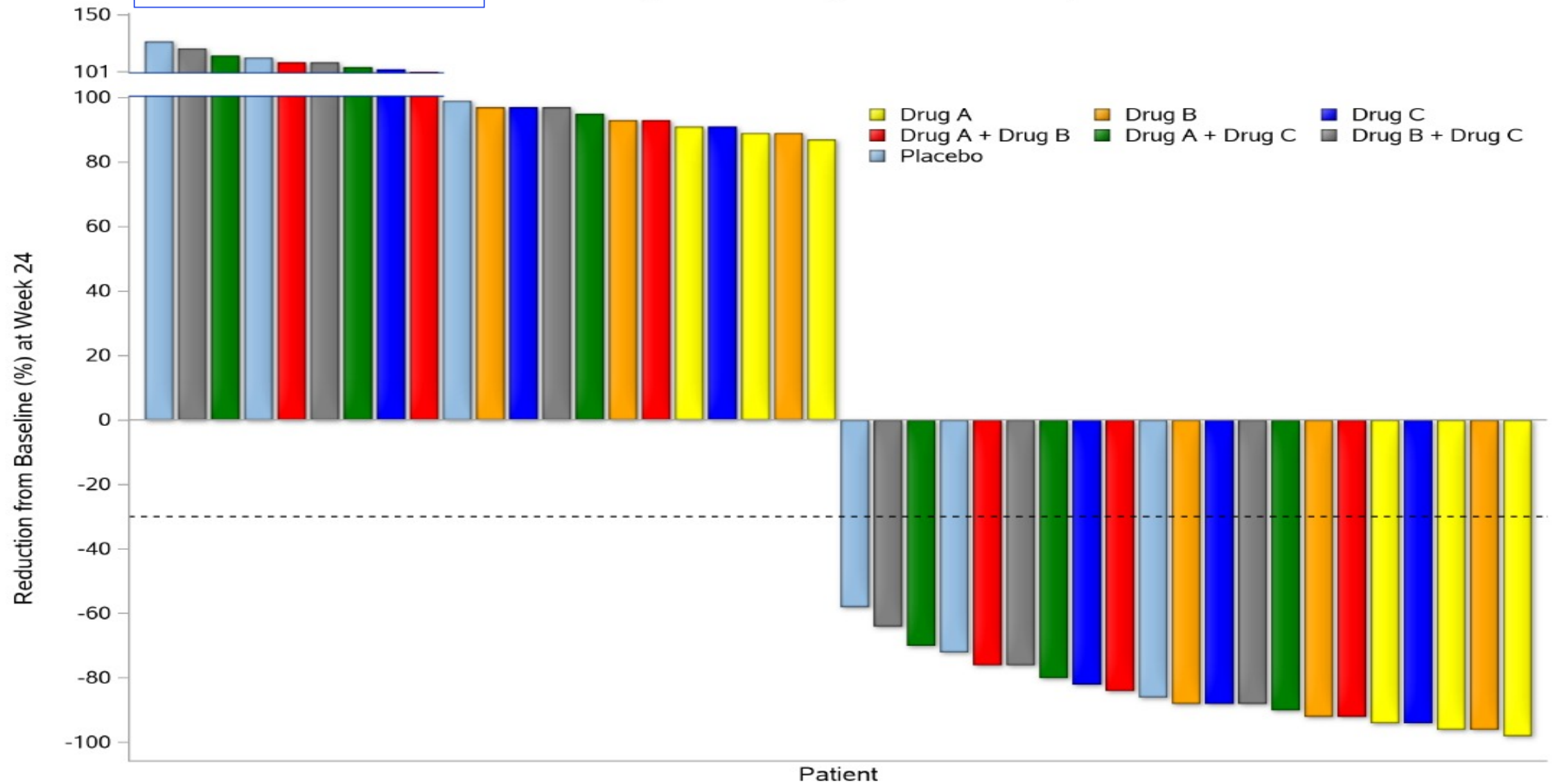
      layout overlay / border=false walldisplay=(fill)
        yaxisopts=(display=(line ticks tickvalues) griddisplay=off
          linearopts=(viewmin=101 viewmax=150 tickvaluelist=(101 150)
            ))
        xaxisopts=(display=none linearopts=(viewmin=1) )
        ;
      barchart X=xvar Y=yvar11 / index=trt01an stat=sum group=trt01an fillattrs=multicolor dataskin=matte outlineattrs=(color=black) orient=vertical;
      drawline x1=-2.2 y1=-1.5 x2=21 y2=-1.5/x1space=datapercnt y1space=datapercnt x2space=datapercnt y2space=datapercnt lineattrs=(thickness=1px);
    endlayout;

      layout overlay / border=false walldisplay=(fill)
        yaxisopts=(label='Reduction from Baseline (%) at Week 24'
          display=(line label ticks tickvalues) griddisplay=off
          linearopts=(viewmin=-105 viewmax=100 tickvaluelist=(-100 -80 -60 -40 -20 0 20 40 60 80 100)
            ))
        xaxisopts=(display=(line label) label='Patient' linearopts=(viewmin=1))
        ;
      barchart X=xvar Y=yvar12 / index=trt01an stat=sum group=trt01an name='a' fillattrs=multicolor dataskin=matte outlineattrs=(color=black) orient=vertical;
      drawline x1=-2.2 y1=100.2 x2=21 y2=100.2/x1space=datapercnt y1space=datapercnt x2space=datapercnt y2space=datapercnt lineattrs=(thickness=1px);
      referenceline y=-30 / lineattrs=(color=black pattern=2);
      discretelegend 'a' /across=3 location=inside border=false halign=right valign=top opaque=true ;
    endlayout;
  endlayout;
endgraph;
end;
run;

```

Generated Using SAS GTL

Waterfall plot made easy with advanced options



## Waterfall Plot: R Code

```
library(ggplot2)
library(dplyr)
library(patchwork)

wf0$trt01an <- factor(wf0$trt01an, levels=c(1,2,3,4,5,6,7),
                     labels=c('Drug A','Drug B','Drug C','Drug A + Drug B','Drug A + Drug C','Drug B + Drug C','Placebo'))

wf1 <- ggplot(wf0, aes(y=yvar1, x=xvar, fill=trt01an)) + geom_bar(stat='identity', show.legend=F, color="black") +
  scale_fill_manual("", values=c('Drug A'='yellow', 'Drug B'='orange', 'Drug C'='blue', 'Drug A + Drug B'='red',
                                'Drug A + Drug C'='green', 'Drug B + Drug C'='grey', 'Placebo'='lightblue')) +
  geom_segment(y=101, x=-1, yend=101, xend=9.6) + scale_y_continuous(breaks=seq(101,150,49), expand = c(0,0)) +
  scale_x_continuous(breaks=seq(0,42,1), limits=c(0,42), expand = c(0,0)) + xlab(" ") + ylab(" ") +
  theme_bw() + labs(subtitle="Waterfall plot made easy with advanced options") +
  theme(legend.background = element_blank(), legend.key = element_blank(),
        plot.subtitle = element_text(face="bold", hjust=0.5), plot.background = element_blank(),
        panel.background = element_blank(), panel.grid = element_blank(), panel.border = element_blank(),
        axis.line.x = element_blank(), axis.ticks.x = element_blank(), axis.text.x = element_blank(),
        axis.line.y = element_line(colour = "black")) + coord_cartesian(ylim = c(100.5, 150), xlim=c(0,42))

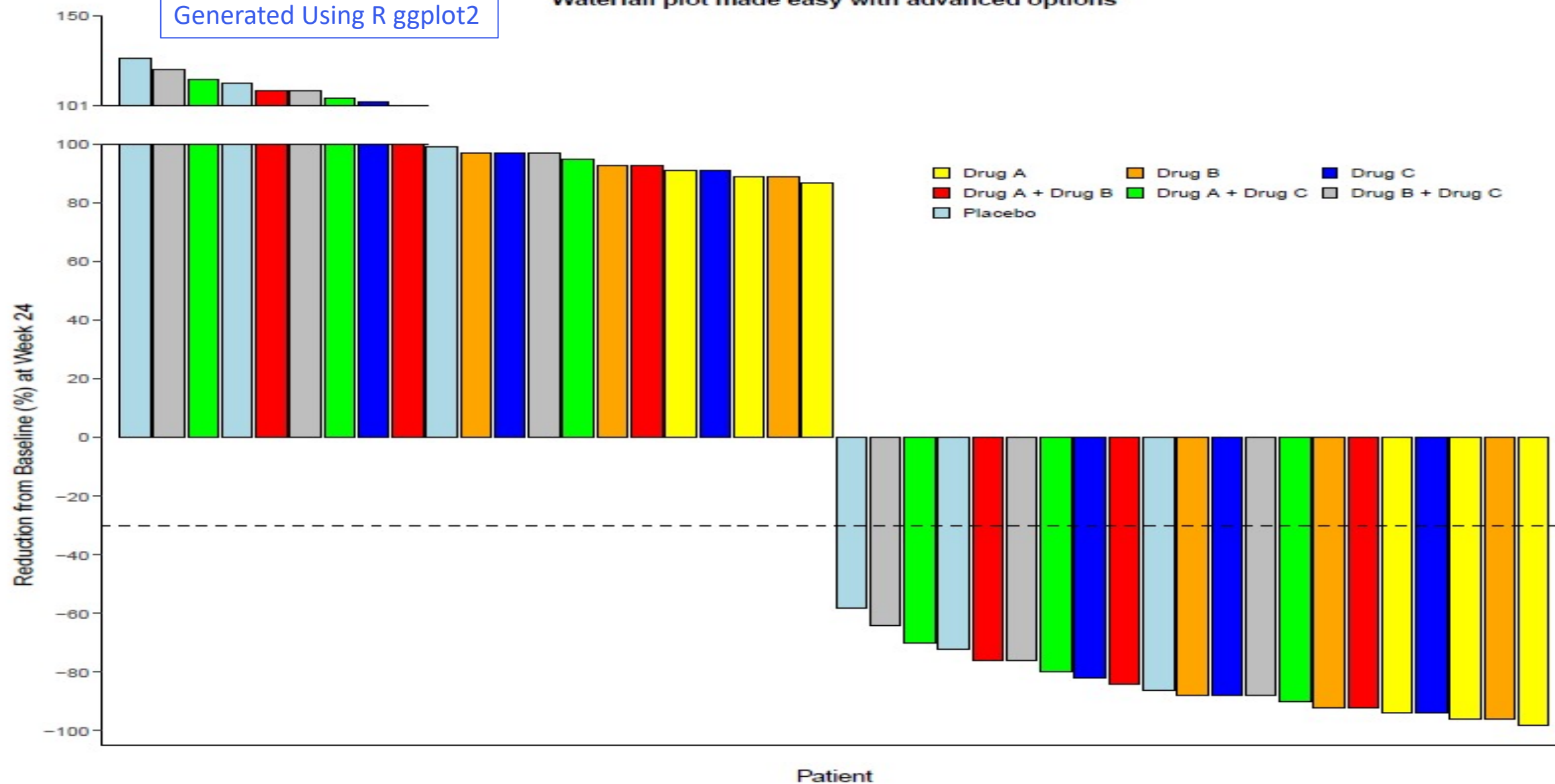
wf2 <- ggplot(wf0, aes(y=yvar1, x=xvar, fill=trt01an)) + geom_bar(stat='identity', color="black") +
  geom_hline(yintercept = -30, linetype=2, color="black") +
  scale_fill_manual("", labels=c('Drug A','Drug B','Drug C','Drug A + Drug B','Drug A + Drug C','Drug B + Drug C','Placebo'),
                    values=c("yellow","orange","blue","red","green", "grey","lightblue")) +
  geom_segment(y=100, x=-1, yend=100, xend=9.6) +
  scale_y_continuous(breaks=seq(-100,150,20), limits=c(-105,150), expand = c(0,0)) +
  scale_x_continuous(breaks=seq(0,42,1), limits=c(0,42), expand = c(0,0)) +
  xlab("\nPatient") + ylab("\nReduction from Baseline (%) at Week 24") + theme_bw() +
  theme(legend.justification=c(0,0.5), legend.position=c(0.55, 0.92),
        legend.direction = "horizontal", legend.text.align = 0, legend.key.size = unit(0.28, 'cm'),
        legend.background = element_blank(), legend.key = element_blank(), plot.background = element_blank(),
        panel.background = element_blank(), panel.grid = element_blank(), panel.border = element_blank(),
        axis.ticks.x = element_blank(), axis.text.x = element_blank(), axis.line.x = element_line(colour = "black"),
        axis.line.y = element_line(colour = "black")) + coord_cartesian(ylim = c(-105, 100), xlim=c(0,42)) +
  guides(fill = guide_legend(ncol = 3, nrow = 3, byrow = T))

wffinal <- wf1 + wf2 + plot_layout(ncol=1, heights=c(0.15,1))

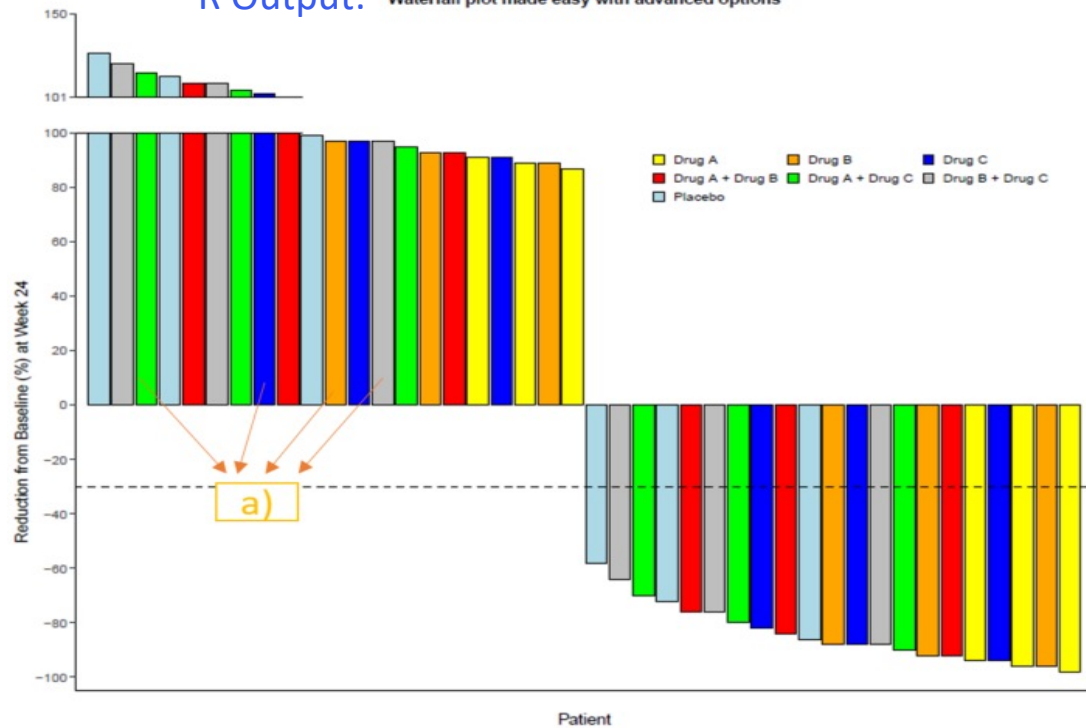
ggsave(wffinal, file="//Waterfallplot.pdf", width = 10, height=8, dpi=300)
```

Generated Using R ggplot2

Waterfall plot made easy with advanced options



## R Output: Waterfall plot made easy with advanced options



**Time:** To create, R took less time when compared to SAS.

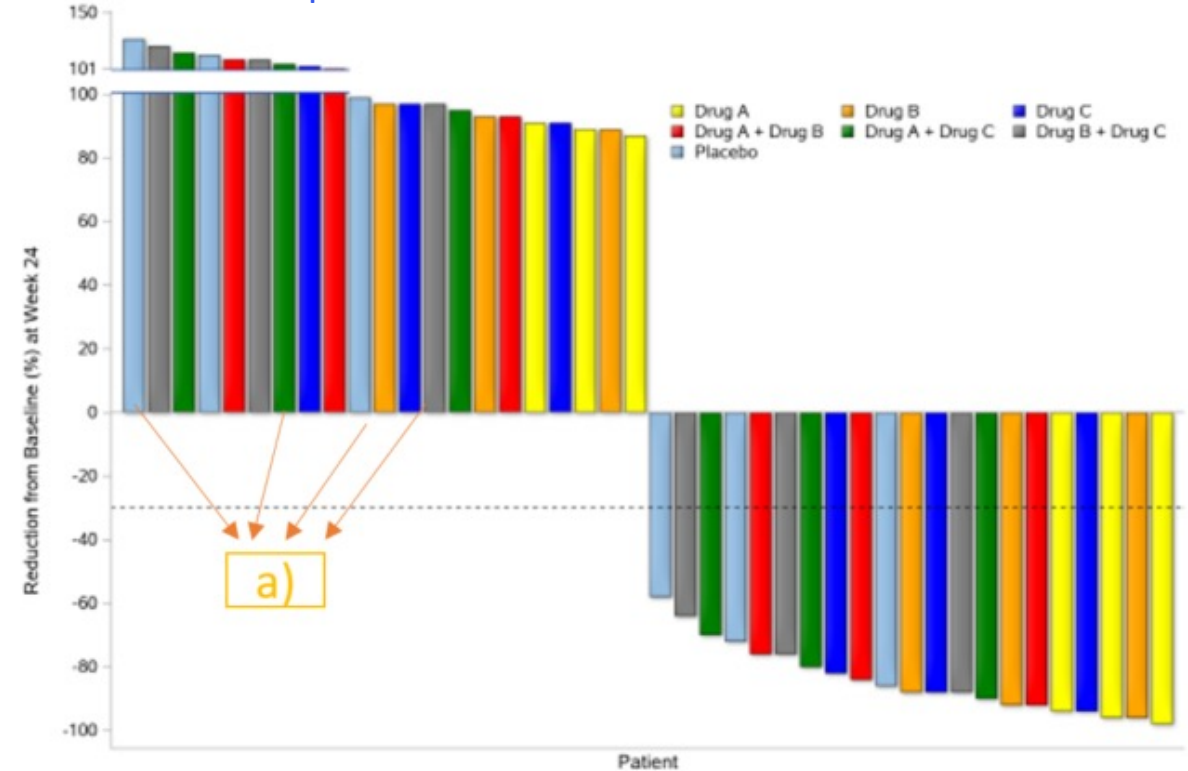
**Quality:** Almost similar but SAS has an edge with the heightened visual effects.

**Gentle note:** Similar conclusion can be attributed to Swimmer lane plot.

**Differences:** Some of the observed limitations are indicated below.

a) SAS has dataskin options like CRISP, GLOSS, MATTE, PRESSED and SHEEN, which adds a heightened visual effect to two-dimensional plots and these options are missing in R ggplot2.

## SAS output: Waterfall plot made easy with advanced options



```
proc template;
  define statgraph srmMultiplot;
    begingraph / border=false attrpriority=none axisbreaksymbol=slantedleft axisbreaktype=axis;
      entrytitle textattrs=(size=10 weight=normal) "Figure 1. &param By Treatment Among Patients with Abnormal Baseline";
      entrytitle textattrs=(size=10 weight=normal) " ";
      layout lattice /border=false columns=3 rows=2 columngutter=1px rowgutter=1px
        columndatarange=data rowdatarange=data ;

      cell;cellheader;entry "Drug A";endcellheader;
      layout overlay/ walldisplay=(fill) yaxisopts= ( tickvalueattrs=(size=1) linearopts=(includeranges=(&ylow-&ybrk &ybrkcnt-&ymax)))
      xaxisopts=( label=' ' display=(line ticks) offsetmin=0.1 offsetmax=0.1 tickvalueattrs=(size=8) linearopts=(tickvaluelist=(&xvaluelist) viewmax=&xmax) );
      seriesplot y=trt1 x=avisitn/group=usubjid index=usubjdn display=(markers) datatransparency=0.5;
      referenceline y=&ybrk/lineattrs=(color=red pattern=2) curvelabel="&ybrk" curvelabelattrs=(color=red size=1);;
      endlayout;
      endcell;

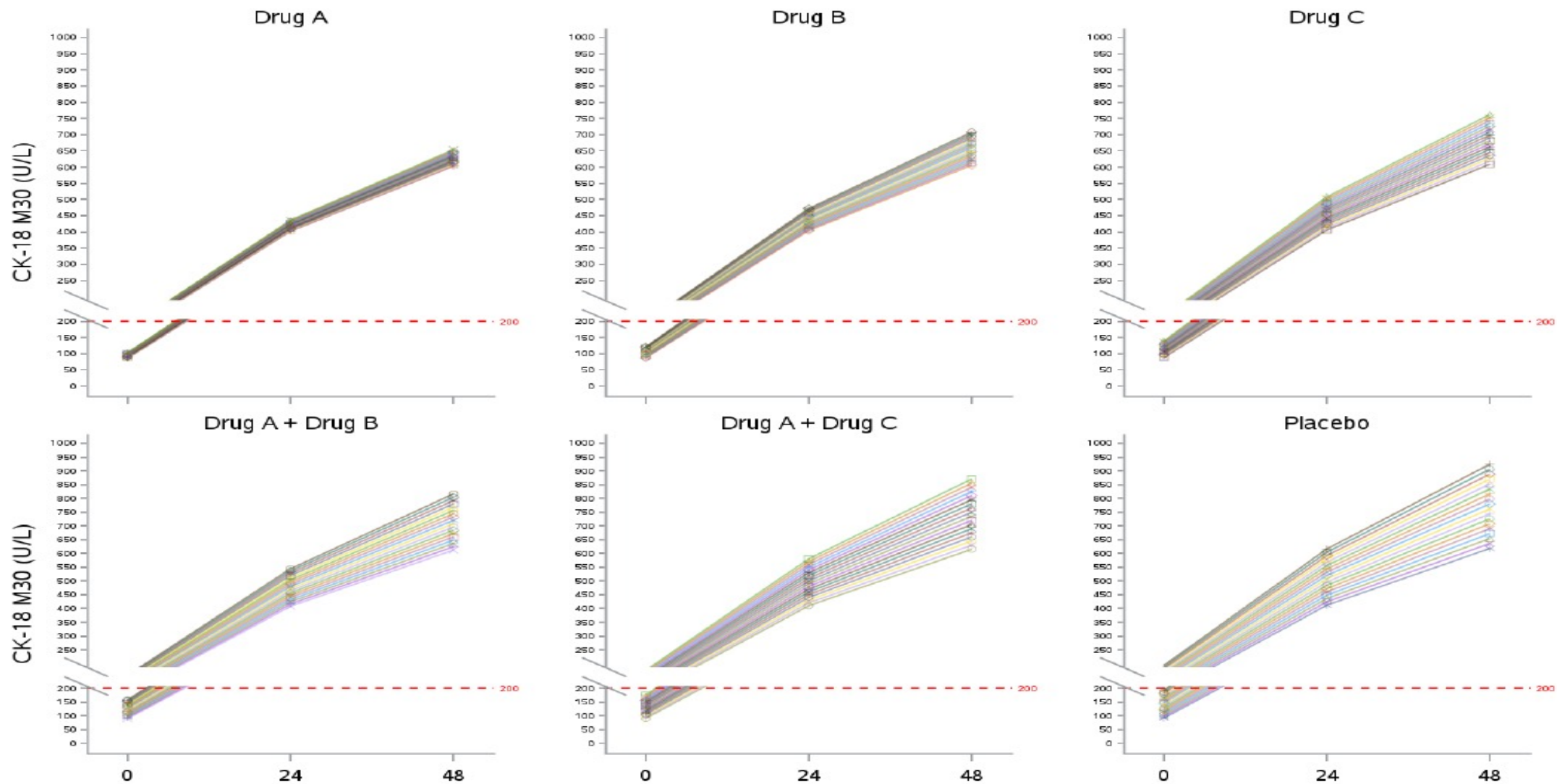
      cell;cellheader;entry "Drug B";endcellheader;
      layout overlay/ walldisplay=(fill) yaxisopts= ( label=' ' tickvalueattrs=(size=1) linearopts=(includeranges=(&ylow-&ybrk &ybrkcnt-&ymax)))
      xaxisopts=( label=' ' display=(line ticks) offsetmin=0.1 offsetmax=0.1 tickvalueattrs=(size=8) linearopts=(tickvaluelist=(&xvaluelist) viewmax=&xmax) );
      seriesplot y=trt2 x=avisitn/group=usubjid index=usubjdn display=(markers) datatransparency=0.5;
      referenceline y=&ybrk/lineattrs=(color=red pattern=2) curvelabel="&ybrk" curvelabelattrs=(color=red size=1);;
      endlayout;
      endcell;

      cell;cellheader;entry "Drug C";endcellheader;
      layout overlay/ walldisplay=(fill) yaxisopts= ( label=' ' tickvalueattrs=(size=1) linearopts=(includeranges=(&ylow-&ybrk &ybrkcnt-&ymax)))
      xaxisopts=( label=' ' display=(line ticks) offsetmin=0.1 offsetmax=0.1 tickvalueattrs=(size=8) linearopts=(tickvaluelist=(&xvaluelist) viewmax=&xmax));
      seriesplot y=trt3 x=avisitn/group=usubjid index=usubjdn display=(markers) datatransparency=0.5;
      referenceline y=&ybrk/lineattrs=(color=red pattern=2) curvelabel="&ybrk" curvelabelattrs=(color=red size=1);;
      endlayout;
      endcell;

      /*Note: Please repeat the above step by sub setting the data to the other drugs as well.*/

    endlayout;
  endgraph;
end;
run;
```

Figure 1. CK-18 M30 (U/L) By Treatment Among Patients with Abnormal Baseline



## Multipanel Plot: R Code

```
library(ggplot2)
library(grid)
library(gridExtra)
library(cowplot)

mp0$trt01an <- factor(mp0$trt01an,levels=c(1,2,3,4,5,6),
                      labels=c('Drug A','Drug B','Drug C','Drug A + Drug B','Drug A + Drug C','Placebo'))

mp1 <- ggplot(subset(mp0, mp0$trt01an == "Drug A")) + geom_line(aes(x = avisitn,y = aval,
  group=usubjid, color=as.factor(usubjid)),show.legend=F) + geom_text(aes(47.5,200,label=200,vjust=1.4),size=3.5,color="red") +
  geom_hline(aes(yintercept = 200),color="red",linetype=2) + theme_classic() + scale_colour_brewer(palette="Set3") +
  scale_x_continuous("",breaks=seq(0,48,24),minor_breaks=NULL) + theme(axis.text.x = element_blank()) +
  scale_y_continuous("",breaks=seq(0,200,25),expand = c(0,0)) + coord_cartesian(ylim=c(0,200), xlim=c(0,48))

mp2 <- ggplot(subset(mp0, mp0$trt01an == "Drug A")) + geom_line(aes(x = avisitn,y = aval,
  group=usubjid, color=as.factor(usubjid)),show.legend=F) +
  theme_classic() + scale_colour_brewer(palette="Set3") + scale_x_continuous("",breaks=c(0,24,48)) +
  scale_y_continuous("",breaks=c(201,400,600,800,1000),expand = c(0,0)) +
  coord_cartesian(ylim=c(201,1000)) + theme(axis.line.x = element_blank(),
  axis.ticks.x = element_blank(),axis.text.x=element_blank(),plot.title = element_text(hjust = 0.5))

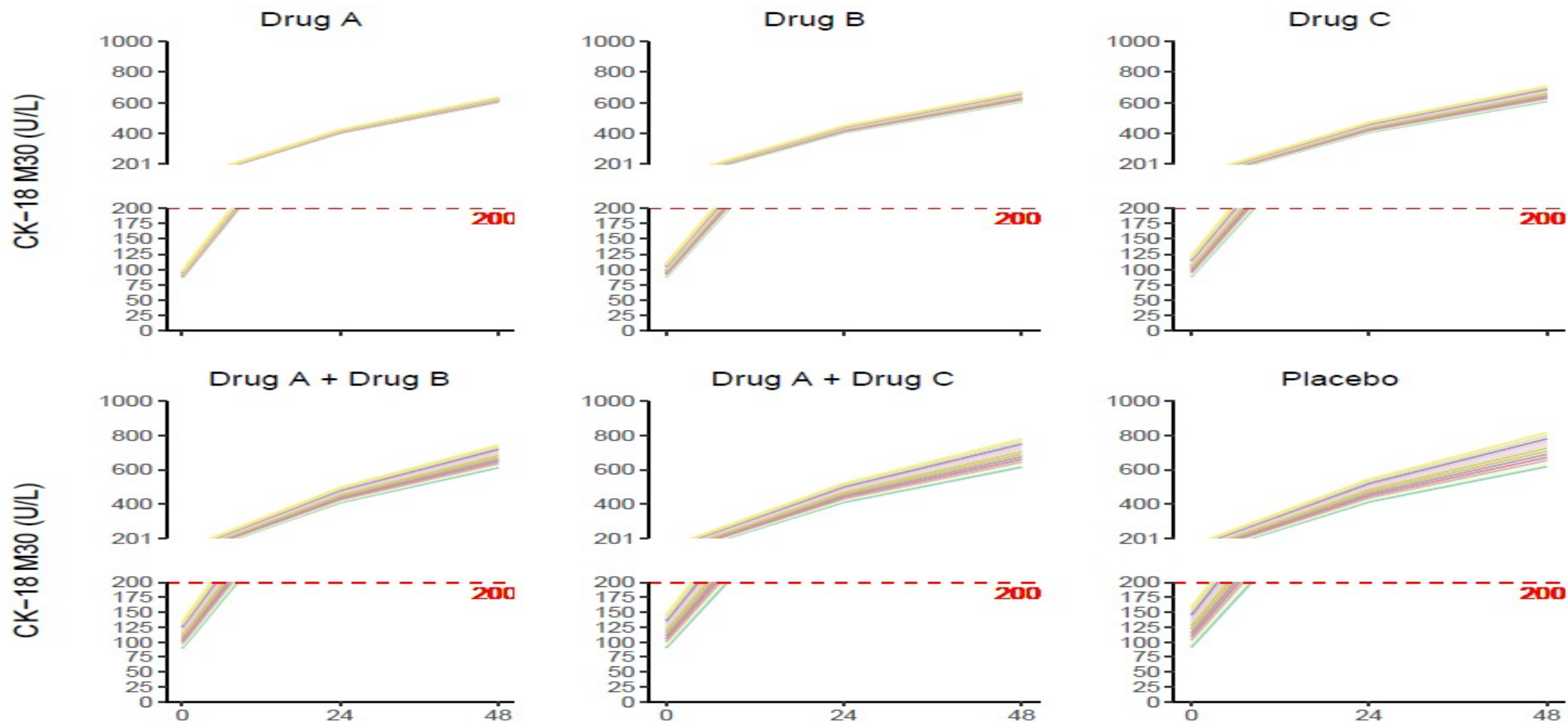
mp1a <- plot_grid(mp2,mp1, ncol = 1, align = "v",rel_heights=c(1,1))

#Note: Please repeat the above step by sub setting the data to the other drugs as well and concatenate them using the following step.

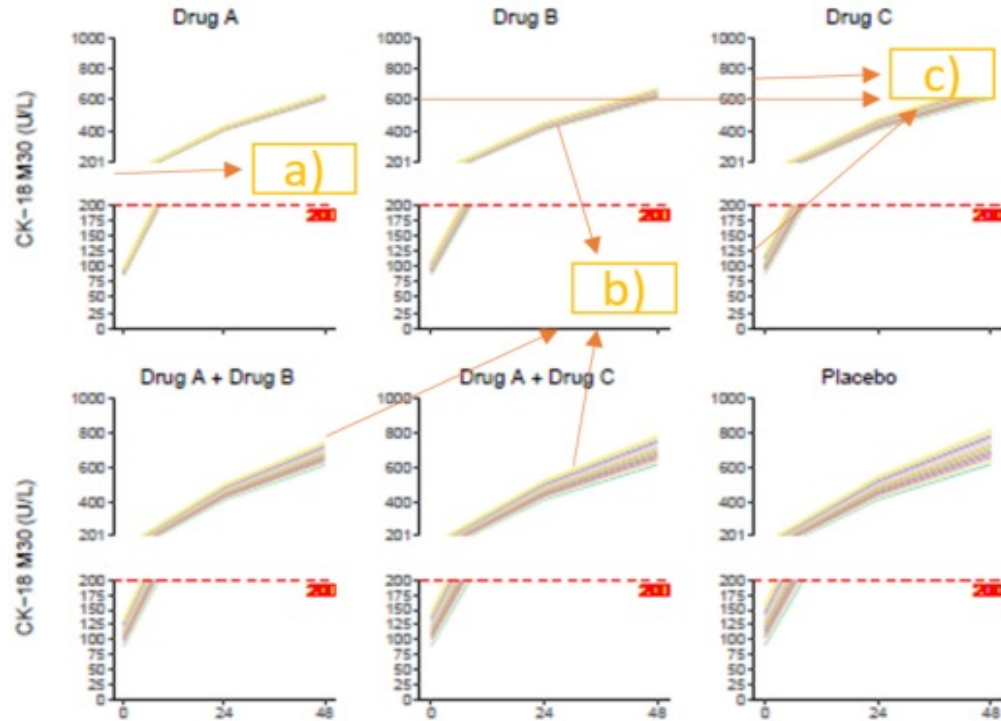
mpfinal <- grid.arrange(arrangeGrob(mp1a,top='\n      Drug A',left="CK-18 M30 (U/L)"),
  arrangeGrob(mp1b,top='\n      Drug B'),
  arrangeGrob(mp1c,top='\n      Drug C'),
  arrangeGrob(mp1d,top='      Drug A + Drug B',left="CK-18 M30 (U/L)"),
  arrangeGrob(mp1e,top='      Drug A + Drug C'),
  arrangeGrob(mp1f,top='      Placebo'),
  top = "Figure 1. CK-18 M30 (U/L) By Treatment Among Patients with Abnormal Baseline",
  ncol = 3,nrow=3)

ggsave(mpfinal, file="//Multipanelplot.pdf", width = 8, height=10, dpi=300)
```

Figure 1. CK-18 M30 (U/L) By Treatment Among Patients with Abnormal Baseline



## R Output: Figure 1. CK-18 M30 (U/L) By Treatment Among Patients with Abnormal Baseline

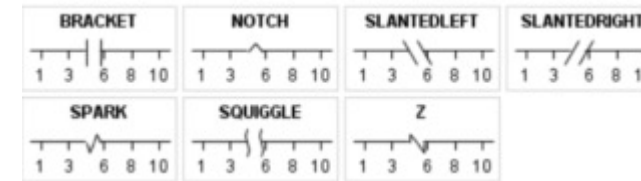


**Time:** To create, SAS took less time when compared to R.

**Quality:** Almost similar.

**Differences:** Some of the observed limitations are indicated below.

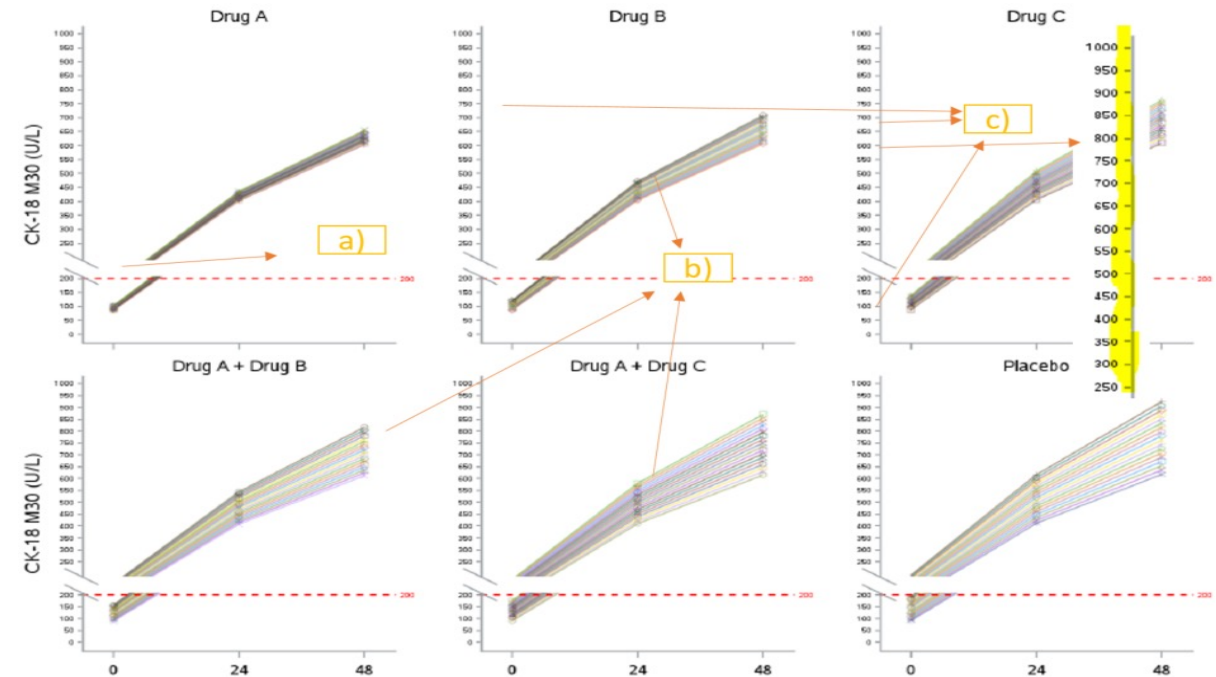
a) SAS has following pre-specified broken axis options (with some limited restrictions while displaying the data). R doesn't have these options, so we followed the similar technique that we used for Waterfall plot.



b) SAS is adjusting the thickness of the series plot; with R it is not the case.

c) For the y-axis tickvalues: Because of SAS limitations, tickvalue increments have to be consistent across the y-axis. With R, after the y-axis gap, lower limit (i.e., 201) is clearly visible and y-axis increments can be easily altered as shown in the above plot.

## SAS Output: Figure 1. CK-18 M30 (U/L) By Treatment Among Patients with Abnormal Baseline



# Conclusions

- Whatever graphs we have created so far using SAS for regulatory submissions, those can be plotted using R as well.
- R is an open source software and newer generations of software engineers are getting exposed to R courses (in the academic institutions) when compared to traditional SAS (which is used prevalently in Biostatistics course, offered in public health degree), which gives an added advantage to R.
- On the other hand, understanding the uses of R language in pharmaceutical industry, all the SAS programmers are getting (or will get) trained in R, in due course of time.
- Small and mid-size CRO companies might adopt R for their daily use because of their budgetary constraints.
- As programmers, we might need to adapt to changing environment by learning new languages like R, Python (which has already emerged as an alternative), Spotfire, etc.



Author Name: Javvaji, Girija Manohar

Labcorp Drug Development Inc. (previously Covance)

4000 Centergreen Way # 300

Cary, NC 27513

Work Phone: 224-280-4518

Email: [girija.javvaji@labcorp.com](mailto:girija.javvaji@labcorp.com)

Co-Author Name: Konda, Sivasankar

Labcorp Drug Development Inc. (previously Covance)

4000 Centergreen Way # 300

Cary, NC 27513

Work Phone: 513-426-4512

Email: [siva.konda@labcorp.com](mailto:siva.konda@labcorp.com)