Generating Clinical Graphs in SAS and R – A Comparison of the Two Languages

Kriss Harris: worked at GlaxoSmithKline for almost 6 years from 2005 to 2011 as a Statistician supporting drug discovery. Whilst at GSK he developed an increasing passion for teaching and taught SAS Graphics to SAS Programmers, Statisticians and Scientists. Since then Kriss has moved on to become an independent Statistical Programmer and has consulted at Eli Lilly and Eisai.

Currently Kriss is working at WCG Clinical Endpoint Solutions and helps to automate the process and develop edit checks.

Generating Clinical Graphs in SAS and R – A Comparison of the Two Languages

Endri Elnadav: Over 17 years experience in SAS Programming

Recently moved to Sydney, Australia

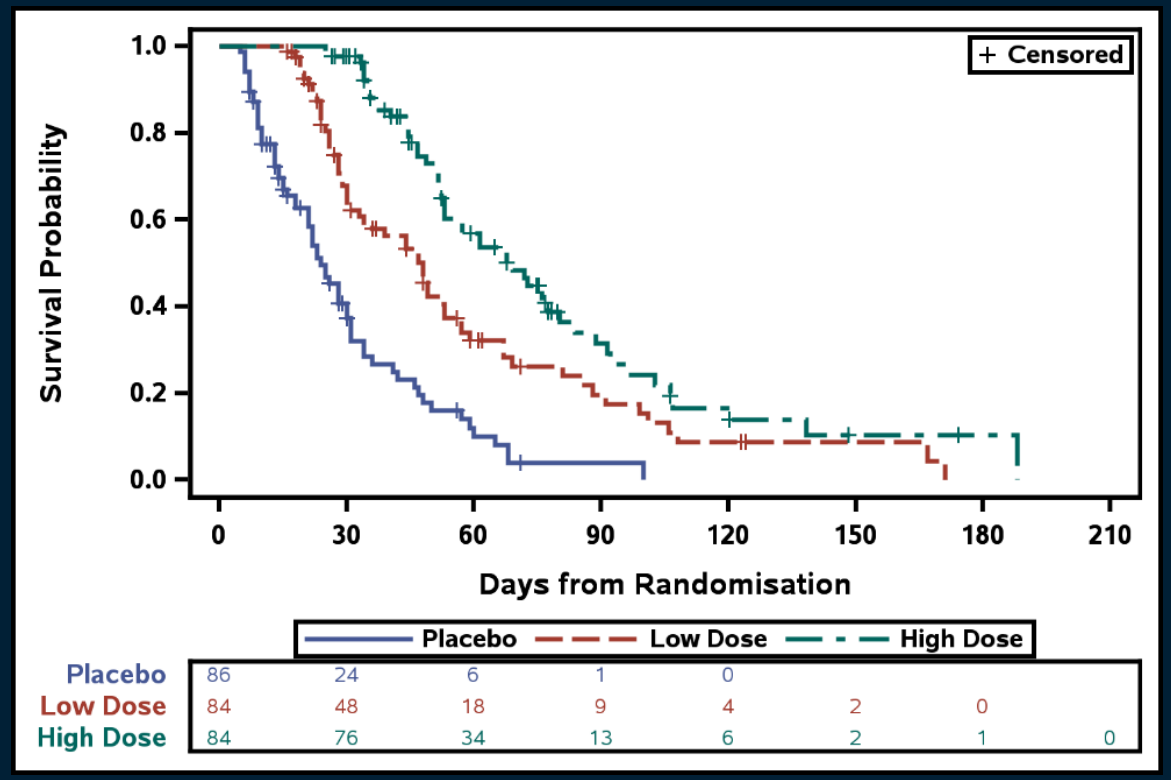
Working as a consultant for EU and US company

Passions:

- Process optimization
- Digitalization, i.e. using Python / R(Shiny)
- Code generator

Agenda

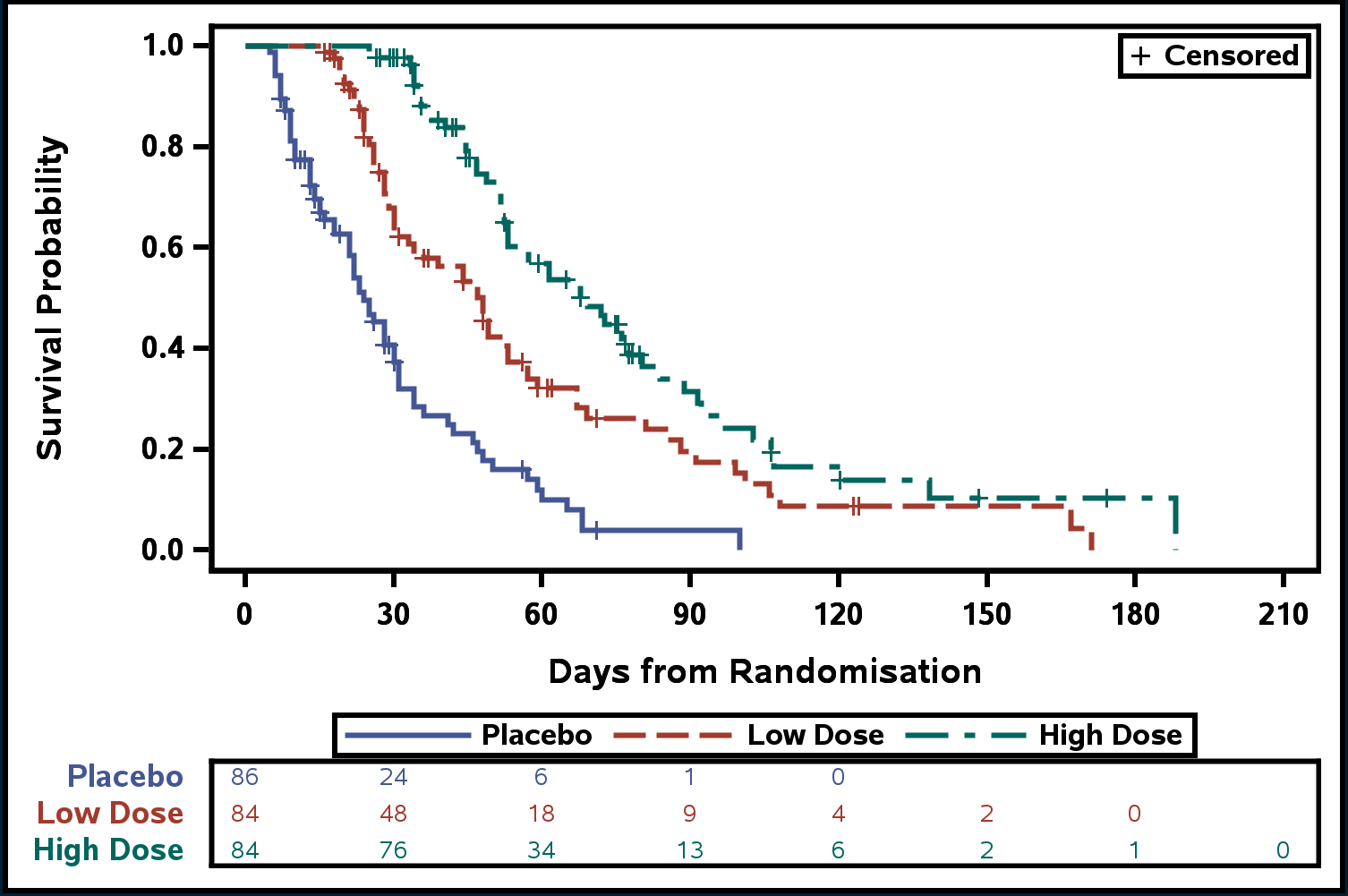
- In this hands-on workshop we will concentrate on two programming languages - SAS and R.
- We will show you how to produce frequently requested clinical graphs, such as Kaplan-Meier (KM) plots, and forest plots (and waterfall plots) in both languages.
- As a bonus, we'll show you how you can use code generators using template technology to create SAS code from R and vice versa.



Creating Kaplan Meier Plot using SAS

Kaplan-Meier Plot

LIFETEST Procedure: SurvivalPlot



Placebo	86	24	6	1	0			
Low Dose	84	48	18	9	4	2	0	
High Dose	84	76	34	13	6	2	1	0

Obtaining ODS Output Object Names

```
ods trace on;
proc lifetest data = adam.adtteeff
   plots=survival(atrisk=0 to 210 by 30);
   time aval * cnsr(1);
   strata trtpn;
run;
ods trace off;
```

ODS Output Object Names

```
--------------

Output Added:
--------------
Name:         SurvivalPlot
Label:        Survival Curves
Template:     Stat.Lifetest.Graphics.ProductLimitSurvival
Path:         Lifetest.SurvivalPlot
--------------
```

ODS Table Names

Seen in the details tab within the help guide

The LIFETEST Procedure

Overview | Getting Started | Syntax | Details | Examples | References

ODS Table Names

PROC LIFETEST assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 72.6. For more information about ODS, see Chapter 20: Using the Output Delivery System.

Table 72.6: ODS Tables Produced by PROC LIFETEST

ODS Table Name	Description	Statement / Option
BreslowEstimates	Breslow estimates	PROC LIFETEST METHOD=B
CensoredSummary	Number of event and censored observations	PROC LIFETEST METHOD=PL | B | FH
CIF	Cumulative incidence function estimates	TIME / EVENTCODE
FailureSummary	Summary of failure outcomes for competing-risks data	TIME / EVENTCODE
FlemingEstimates	Fleming-Harrington estimates	PROC LIFETEST METHOD=FH
FlemingHomCov	Covariance matrix for k-sample FLEMING statistics	STRATA / TEST=FLEMING
GrayTest	Results of k-sample test of Gray (1988) comparing CIFs	TIME / EVENTCODE; STRATA
HomStats	Test statistics for k-sample tests	STRATA / TEST=
HomTests	Results of k-sample tests	STRATA / TEST=
LifetableEstimates	Life-table survival estimates	PROC LIFETEST METHOD=LT
LogForStepSeq	Forward stepwise sequence for the log-rank statistics for association	TEST

ODS Graph Names

Seen in the details tab within the help guide

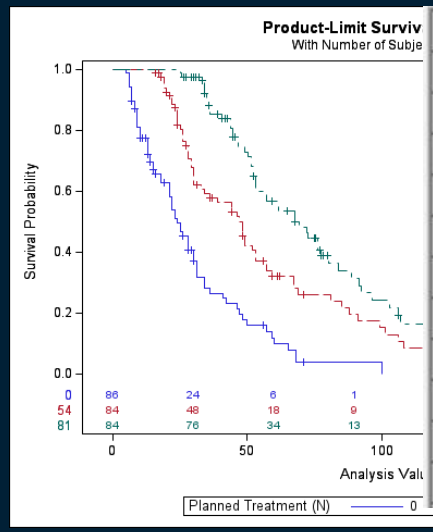
Table 72.7: Graphs Produced by PROC LIFETEST

ODS Graph Name	Plot Description	PLOTS= Option
cifPlot	Cumulative incidence function	CIF
cifPlot	Cumulative incidence function with pointwise confidence limits	CIF(CL)
cifPlot	Cumulative incidence function with Gray's test	CIF(TEST)
DensityPlot	Density function for life-table method	PDF
FailurePlot	Cumulative distribution function	survival(FAILURE)
HazardPlot	Hazard function for life-table method or smoothed hazard for product-limit, Breslow, or Fleming-Harrington method	HAZARD
LogNegLogSurvivalPlot	Log(-log(survivor function)	LOGLOGS
NegLogSurvivalPlot	Log(survivor function)	LOGSURV
SurvivalPlot	Survivor function	SURVIVAL
SurvivalPlot	Survivor function with number of subjects at risk	SURVIVAL (ATRISK)
SurvivalPlot	Survivor function with pointwise confidence limits	SURVIVAL(CL)
SurvivalPlot	Survivor function with equal-precision band	SURVIVAL (CB=EP)
SurvivalPlot	Survivor function with Hall-Wellner band	SURVIVAL (CB=HW)
SurvivalPlot	Survivor function with homogeneity test	SURVIVAL(TEST)

Kaplan-Meier Plot

LIFETEST Procedure

```
ods output SurvivalPlot = SurvivalPlot;
ods output HomTests=HomTests=(test=(where"Log-Rank"));
proc lifetest data = adam.adtteeff plots=survival(atrisk=0 to 210
by 30);
   time aval * cnsr(1);
   strata trtpn;
run;
```



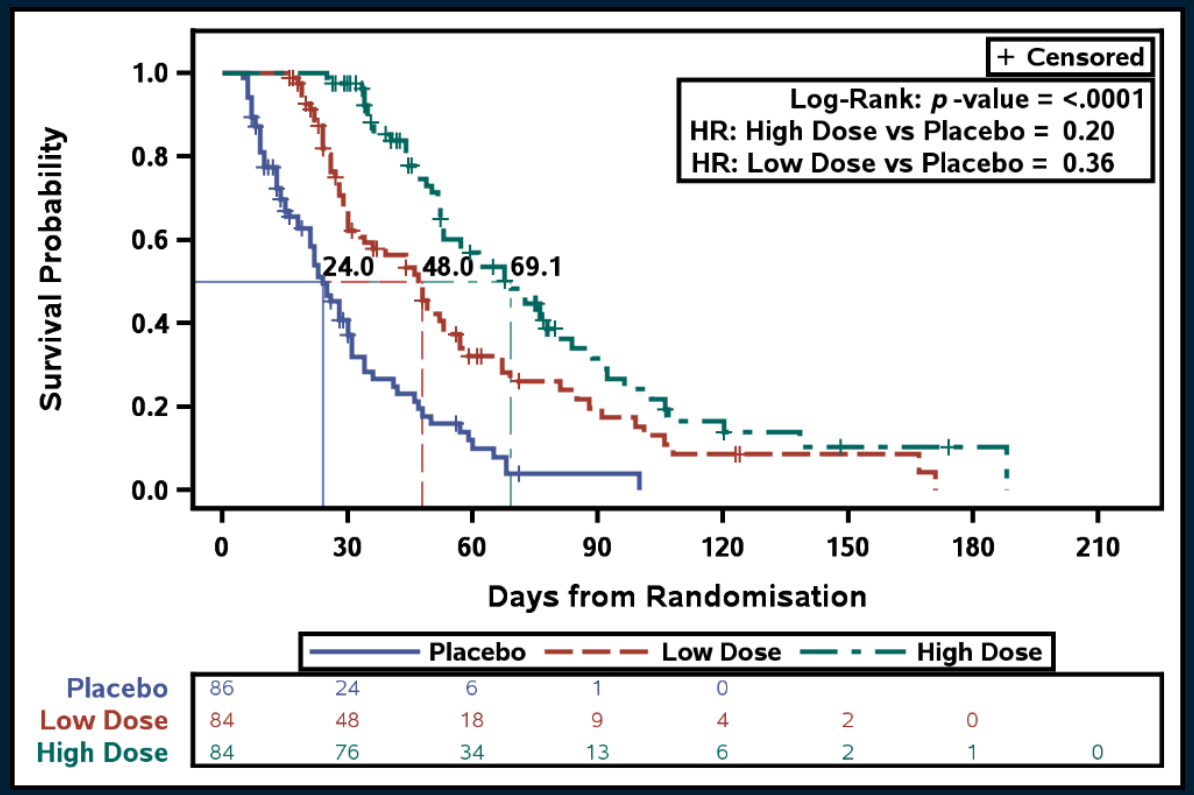
	USUBJID	TRTPN	TRTP	AVAL	CNSR	Stratum	StratumNum
1	01-701-1015	0	Placebo	5	0		1
2	01-701-1023	0	Placebo	30	1		1
3	01-701-1028	81	Xanomeline High Dose	74.7	1		1
4	01-701-1033	54	Xanomeline Low Dose	16	1		1
5	01-701-1034	81	Xanomeline High Dose	52.3	1		1
6	01-701-1047	0	Placebo	31	0		1
7	01-701-1097	54	Xanomeline Low Dose	26	0		1
8	01-701-1111	54	Xanomeline Low Dose	24	1		1
9	01-701-1115	54	Xanomeline Low Dose	44	1		1
10	01-701-1118	0	Placebo	14	0		1
11	01-701-1130	0	Placebo	12	1		1

Exercise 1-1

- Generate the ODS Object SurvivalPlot and name the dataset **KMDataset**

Graph Template Language

Creating Custom Graphs



Creating Kaplan-Meier Plot

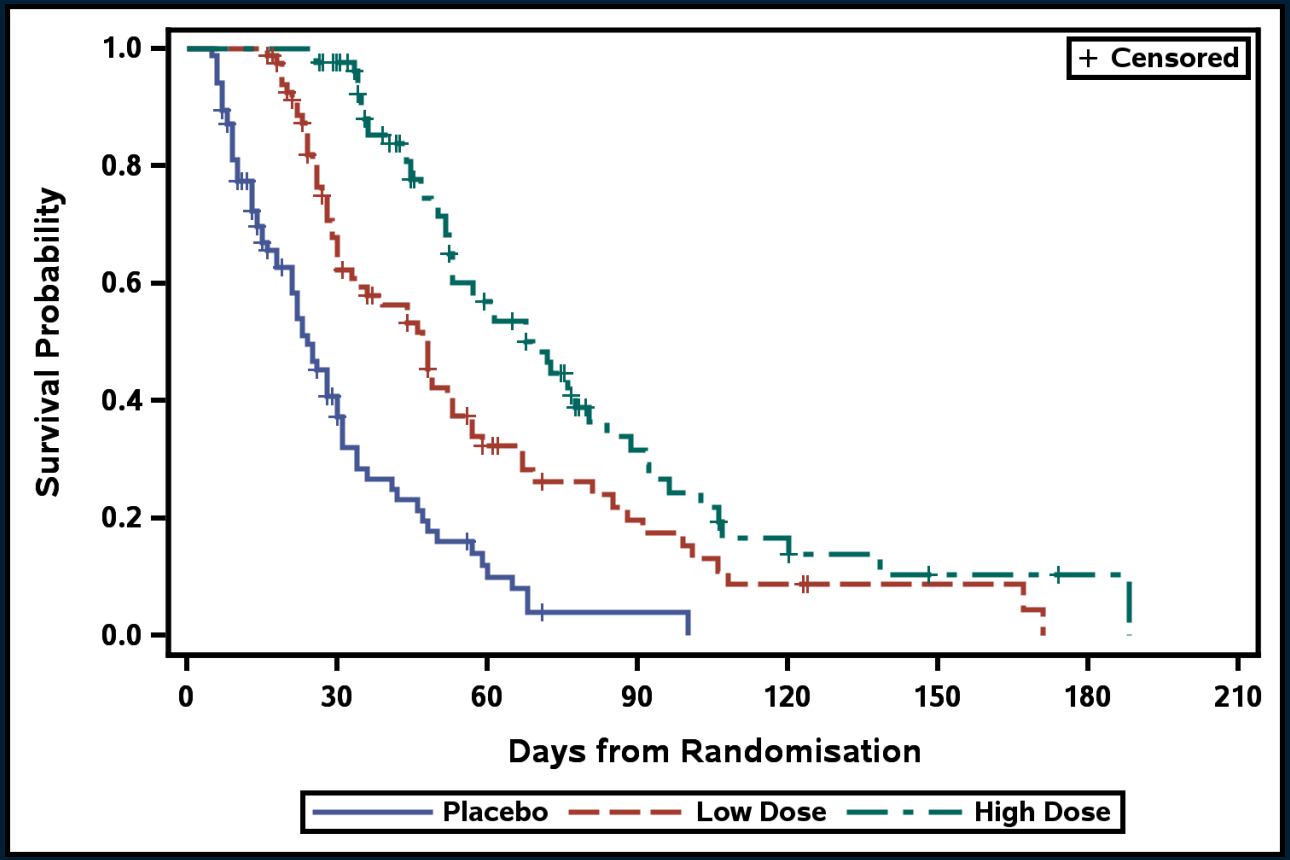
with Median Survival Times and HR table

Creating Kaplan-Meier Plot

with Median Survival Times and HR table

- Use Time-to-event Dataset, for example, ADTTE
- Use PROC LIFETEST to obtain Kaplan Meier survival dataset and median survival times
- Use PROC PHREG to obtain hazard ratios
- Create macro variables that contain the median survival times and hazard ratios
- Use GTL (or SGPLOT) to create the Kaplan-Meier plot

Creating Basic Kaplan-Meier Plot



Graph Template Language

```
proc template;
   define statgraph kmtemplate;
      begingraph;
         layout overlay;
            stepplot x = time y = survival;
            discretelegend "Survival";
         endlayout;
      endgraph;
   end;
run;
```

PROC SGRENDER

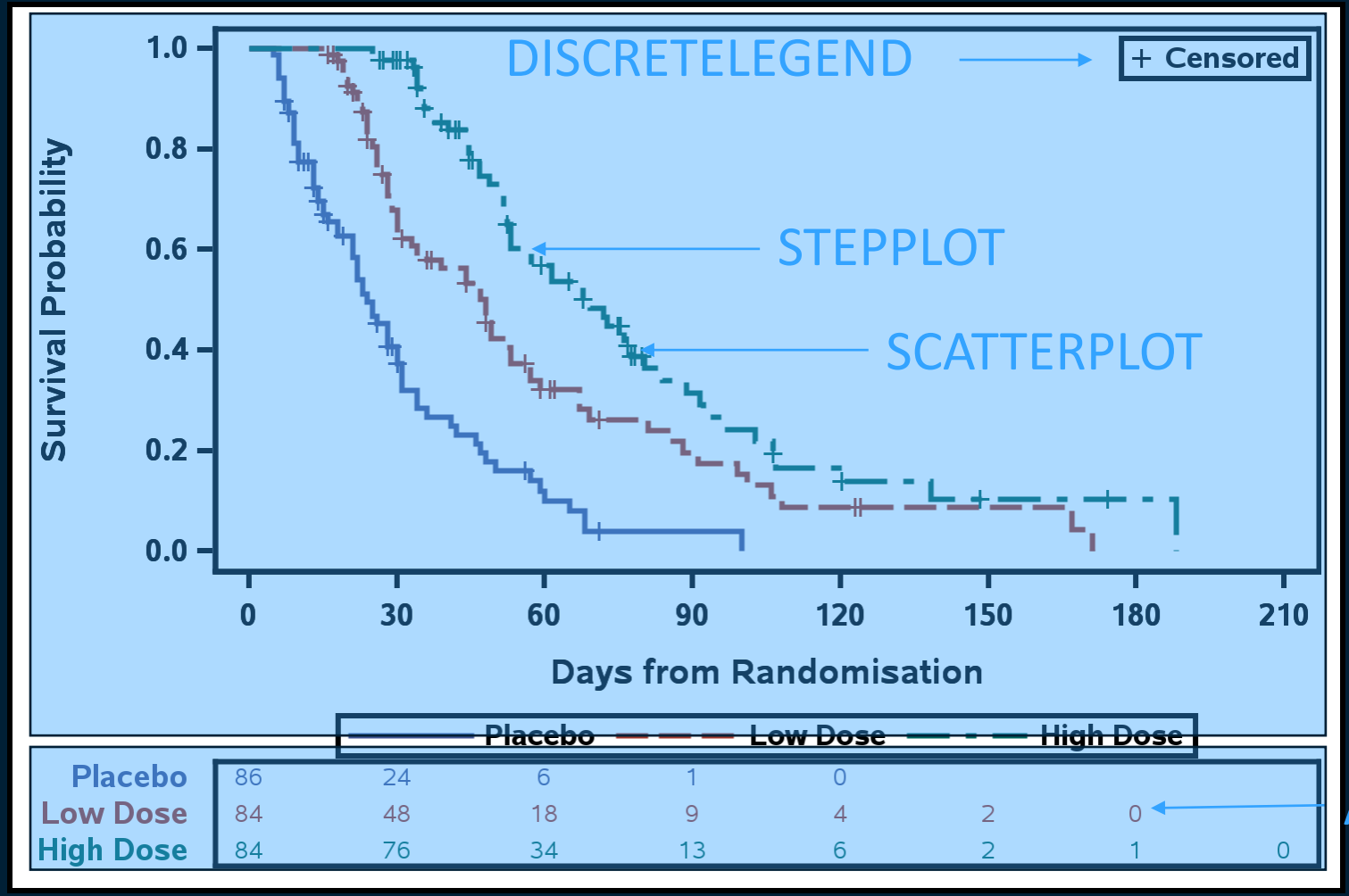
```
proc sgrender data = Survivalplot
              template = kmtemplate;
run;
```

Creating Kaplan-Meier Plot

Step 1

LAYOUT
OVERLAY

LAYOUT
LATTICE



LAYOUT
OVERLAY

AXISTABLE

Step 1 – SAS Code

KM Curve

```
stepplot x = time y = survival /
   group = stratum
   name="Survival"
   legendlabel="Survival";

scatterplot x=time y=censored /
   markerattrs=(symbol=plus)
   group=stratum;
```

Step 1 – SAS Code

Censored Legend

```
scatterplot x=time y=censored /
   markerattrs=(symbol=plus color=black)
   name="Censored";

discretelegend "Censored" /
   location = inside
   autoalign = (topright);
```

Step 1 – SAS Code

At-Risk Table

```
layout overlay /
   xaxisopts=(display=none
      linearopts=(tickvaluesequence=(start=0
      end=210 increment=30))) border=off;

   axistable value=atrisk x=tatrisk /
      class=stratum colorgroup=stratum;
endlayout;
```

Exercise 1-2

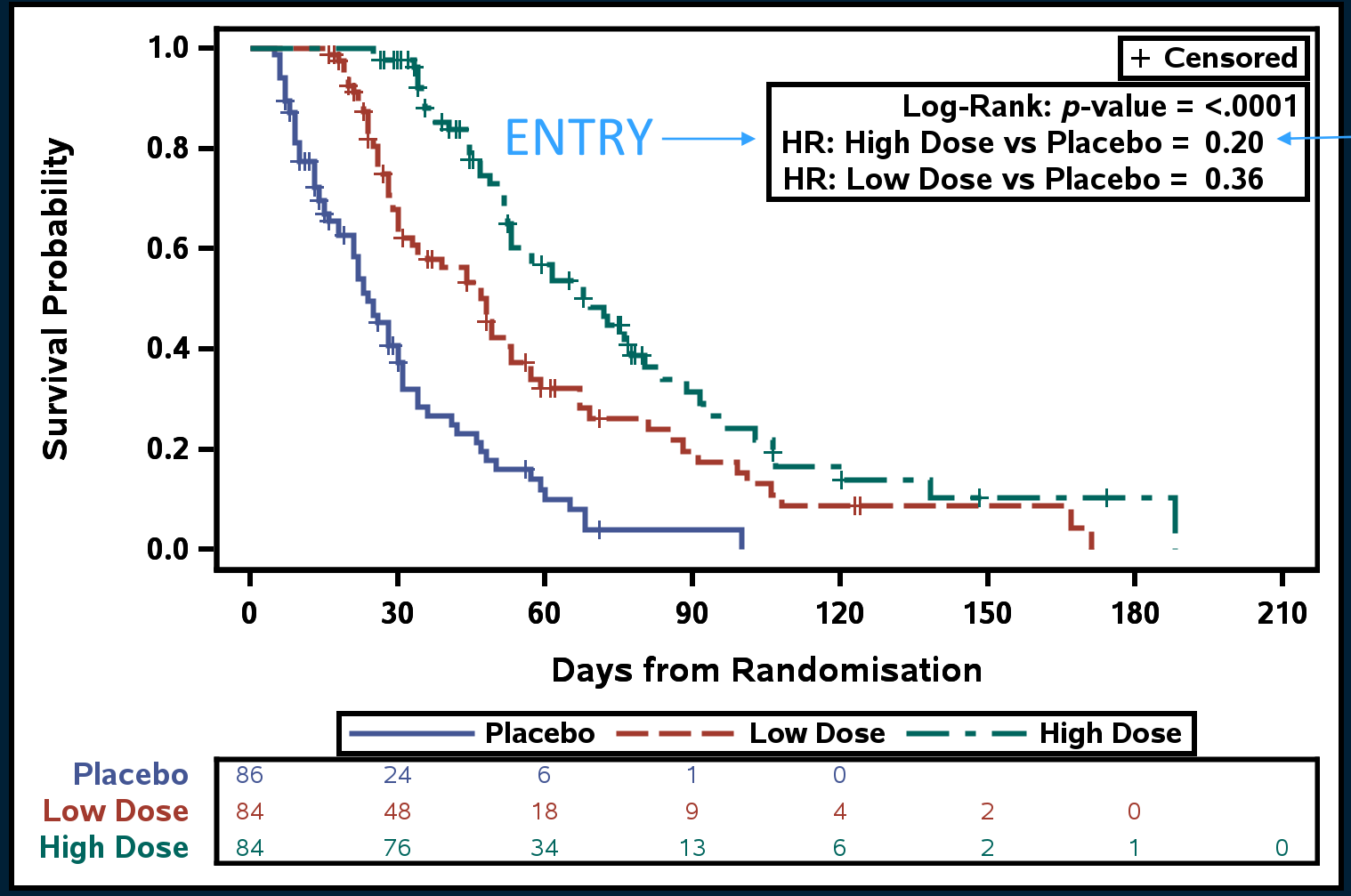
- Search for XXXXX and the KMDataset that you created in Exercise 1-1 into the right part of the code.
- Add the TIME variable (from the KMDataset) into the correct part of the code.

Exercise 1-3

- Search for XXXXXX and enter the appropriate LAYOUT statement.
- Add the number of subjects at risk variable (from the KMDataset) into the correct part of the code.

Creating Kaplan-Meier Plot

Step 2



Step 2 – SAS Code

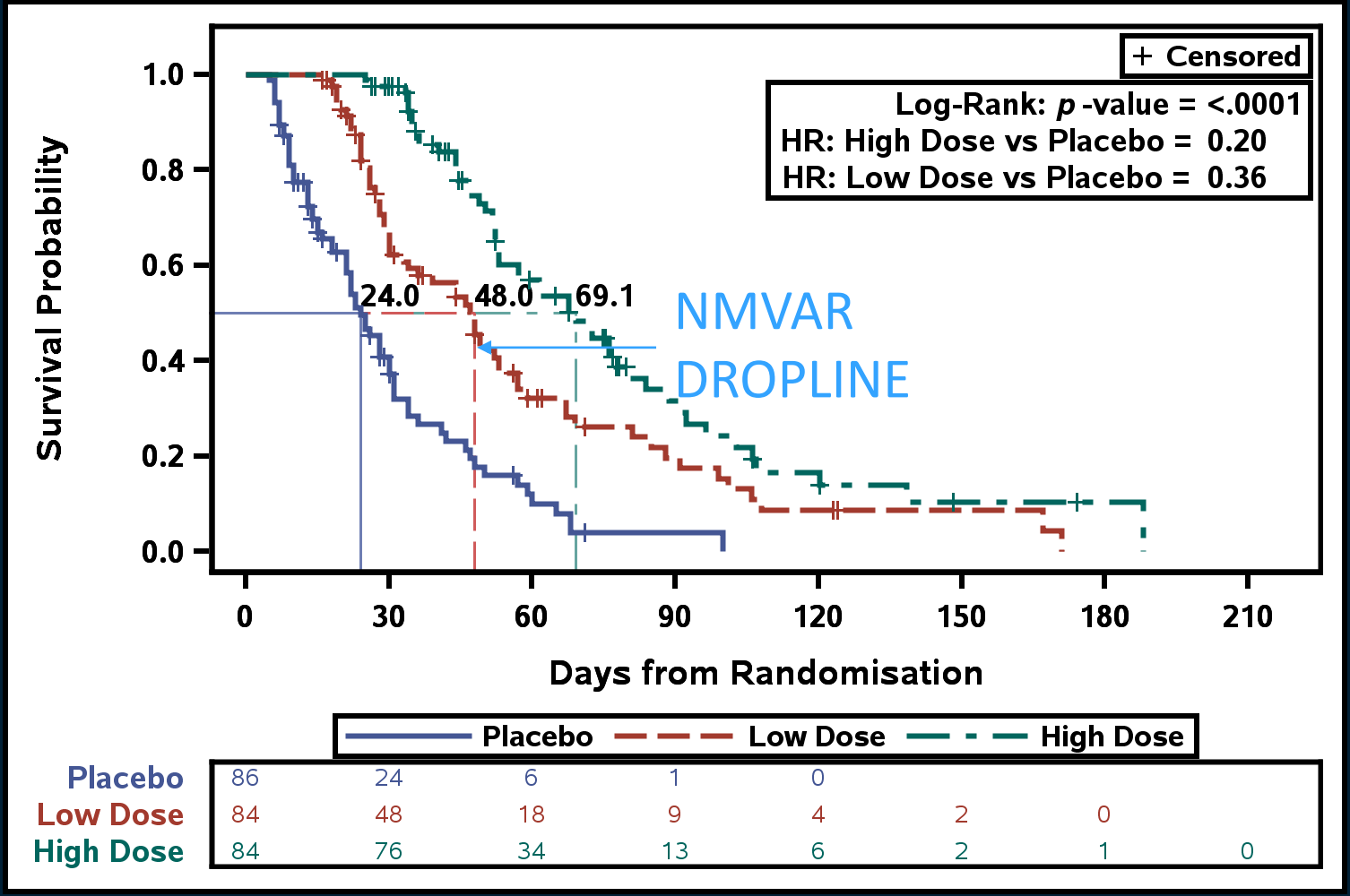
Summary Statistics Table

```
mvar log_rank_pvalue HazardRatio1 HazardRatio2;

layout gridded / columns=2 rows = 3 border = true
   halign = right valign = to outerpad=(top=25px);
   entry halign = right "Log-Rank: "
      textattrs=(style=italic) "p"
      textattrs=(style=normal) "-value = ";
   entry halign = left log_rank_pvalue;
   <Other Entry Statements>
endlayout;
```

Creating Kaplan-Meier Plot

Final Step



Placebo	86	24	6	1	0			
Low Dose	84	48	18	9	4	2	0	
High Dose	84	76	34	13	6	2	1	0

Final Step – SAS Code

Median Survival Time

```
nmvar MedianSurvival1 MedianSurvival2 MedianSurvival3;
mvar CMedianSurvival1 CMedianSurvival2 CMedianSurvival3;

%do i = 3 %to 1 %by -1;
   dropline y = 0.50 x = MedianSurvival&i  /
      dropto = both
      lineattrs=(thickness=1px
         color=graphdata&i:color
         pattern=graphdata&i:linestyle)
      label=CMedianSurvival&i;
%end;
```

R Programming

- Install Rstudio for the programming enviroment of R.
- R is open source and has ~ 14.000 packages online available
- To install packages (using console) for example:

```
install.packages("survival")
```

- Github is used for code sharing.
- All codes for ouw PhUSE - HoW are available under :

https://github.com/ee-analysis

R Programming

1.SimpleKMPlot.r

```
library(survival)
library(haven)

data_path <- "C:/Temp/CDISC Test Data/TDF_ADaM_v1.0/adtte.xpt"

data_adtte <- read_xpt(data_path)

calc_surv <- survfit(Surv(AVAL, CNSR)~TRTA, data= data_adtte)

plot(calc_surv,
     xlab  = "Time",
     ylab  = "Survival Probability",
     main  = "KM Plot",)
```

R Programming

2.AdvancedKMPlot.r

```
library(survival)
library(survminer)
library(haven)

data_adtte <- read_xpt("C:/Temp/CDISC Test Data/TDF_ADaM_v1.0/adtte.xpt")
calc_surv <- survfit(Surv(AVAL, CNSR)~TRTA, data= data_adtte)

calc_plot <- ggsurvplot(calc_surv, risk.table          = TRUE, )
calc_plot
```

→ Try to run and modify the R code in your RStudio!

R Programming

Combine SAS (calculation) and R (display) !

- Can we combine SAS and R for „simple task“ ?
- We will use SAS for the calculation part (i.e. PROC LIFETEST)
- And R will use the SAS datasets for display.
- This is the small step to see how both system can be use for much more powerfull and interactive analysis in future (i.e. using Rshiny).

R Programming

3.KMPlot_using_SAS_R.r

Steps by steps

- Calculation using PROC LIFE TEST in SAS

```
ODS OUTPUT ProductLimitEstimates = calc_010_est;
PROC LIFETEST DATA     = sasfile.adtte
            ATRISK;
   TIME aval * CNSR (0);
   STRATA trta;
RUN;
ODS OUTPUT CLOSE;
```

R Programming

3.KMPlot_using_SAS_R.r

Step 2 (modifcation in SAS)

- Modification of ProductLimitEstimate dataset for flexible display in R
- Expand Product Limit Estimate dataset from Day 1 to Day xxx
- Merge with calculation data
- Retain survival and number at risk variables over the time
- Export SAS dataset to XPT

Step 3 (GGPLOT)

R Programming

3.KMPlot_using_SAS_R.r

Step 3 (GGPLOT)

```
data_path <- "C:/Temp/PhUSE 2022 - HoW/data/lf_est.xpt"
data_calc <- read_xpt(data_path)
kmplot <- ggplot(data = data_calc, aes(x = AVAL, y = SURVIVAL, group=TRTA, colour = TRTA)) +
  geom_step() +
  geom_point(aes(shape = as.factor(CENSOR)), show.legend =F) +
  scale_shape_manual("", values = c(32, 3))
```

R Programming

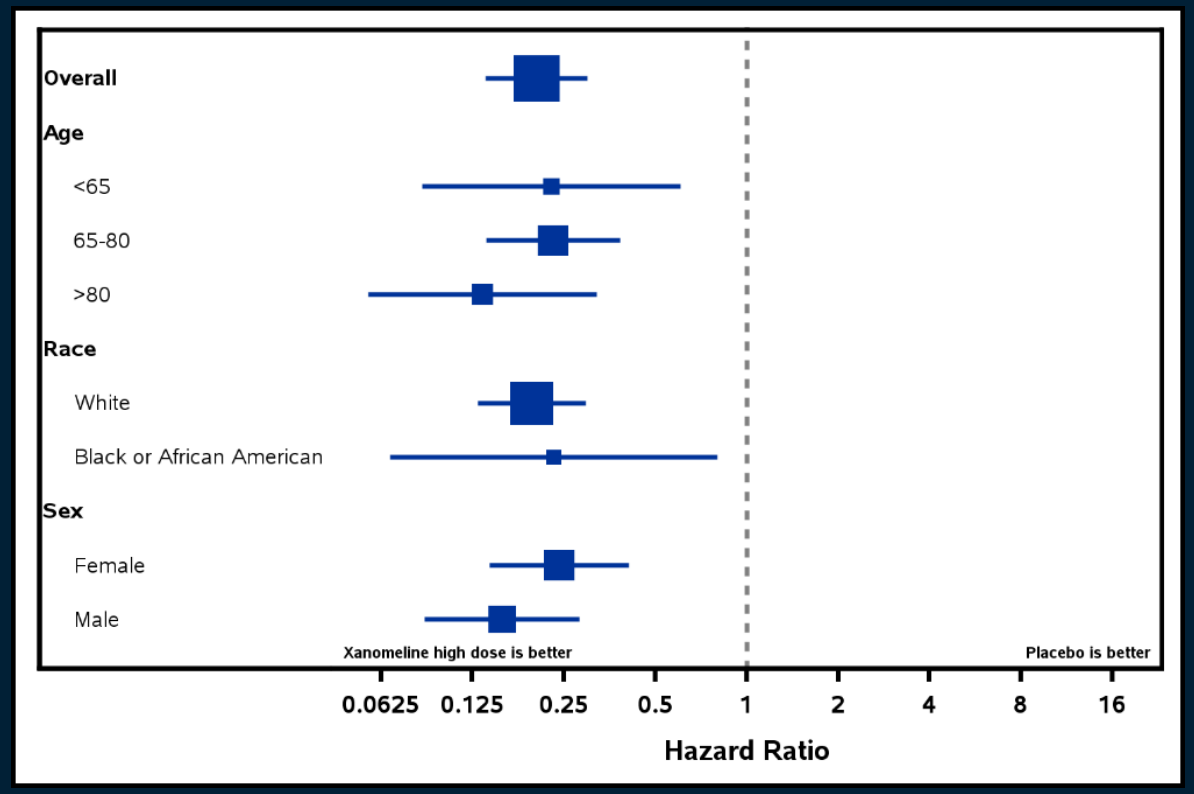
4.AdvancedKMPlot_using_SAS_R.r

Adding „Number at risk“ under the KM plot

- Similar to „LATTICE“ layout, R offers severals packages, i.e. „reshape“
- Flexible handling of plots
  - Plot 1: KM Plot (see previous slide)
  - Plot 2: Table – Number at risks
- Simple define layout of the plots and the ratio of the heights

```
kmplot + kmatrisk  + plot_layout(ncol =  1, heights = c(1, 0.2))
```

See R program!



Creating Forest Plot using SAS

Creating Forest Plot

- Use PROC PHREG to obtain the hazard ratios. Sometimes you may want to plot the odds ratio, and this can be done with PROC LOGISTIC or PROC GENMOD.
- Calculate the number of subjects in each subgroup. This can be used to display the number of subjects that went into calculating the hazard ratio.
- It is very important to order your data appropriately when creating Forest plots.

SAS Code

Subgroup Headers

```
innermargin / align=left;
  axistable y = variable value=variable /
     display=(values)
     indentweight=indent
     textgroup=factortext;
endinnermargin;
```

SAS Code

Hazard Ratio Plot

```
highlowplot y=variable
   low=WaldLower
   high=WaldUpper;

scatterplot y=variable x=HazardRatio /
   markerattrs=(symbol=squarefilled)
   sizeresponse=total_n;
```

SAS Code

Text Statements Related to Placebo

```
layout gridded / Border=false halign=left valign=bottom;
   entry halign=left "Xanomeline high dose is better" /
      textattrs=(size=5);
endlayout;

layout gridded / border=false halign=right
   valign=bottom;
   entry halign=left "Placebo is better" /
      textattrs=(size=5);
endlayout;
```

R Programming

Forest Plot using R

- Any calculation were done before
- Import result (i.e. from Excel)
- It is very important to order your data appropriately when creating Forest plots.

- ggplot2 is a powerful R package
- Each element just can be very easily added
- Layout theme can be controlled using theme function

- See R Example code on Github

R Programming

5.ForestPlot.r

```
plot1 <- ggplot(data_forest, aes(y = INDEX, x = ODDS)) +
  scale_y_continuous(breaks = 1:3, labels = data_forest$LABEL) +
  geom_point(shape = 18, size = 3) +
  geom_errorbarh(aes(xmin = LL, xmax = UL), height = 0.2) +
  geom_vline(xintercept = 1, linetype = "dashed") +
  annotate("text", x = 0  ,  y = 3.5, label = "xxx is better") +
  annotate("text", x = 2,  y = 3.5, label = "yyy is better") +

  annotate("text", x = 3.5,  y = 3.5, label = "95% CI") +
  geom_text(data = data_forest, aes(x = 3.5, label = CI))
```

For more details with comments, see the R code on github.

Conclusion

Conclusion

SAS & R

- SAS : SG procedures are great for creating standard graphs
  R : GGPLOT2 is a great R packages for generating plots
- SAS : Each plot statement within SG procedures have options that allow you to control the appearance of the graph
  R : Each plot elements can be easily added into the graph
- SAS : Data from a procedure can be saved in ODS output object to be used
  R : R has a huge number of statistical procedure and R can easily import any external data (i.e. SAS dataset, XPT, CSV etc.)
- SAS : Creating a custom template and associating with the necessary data allows you to create custom graphs
- R : Style of the layout can controlled using “theme”.

Bonus – HoW

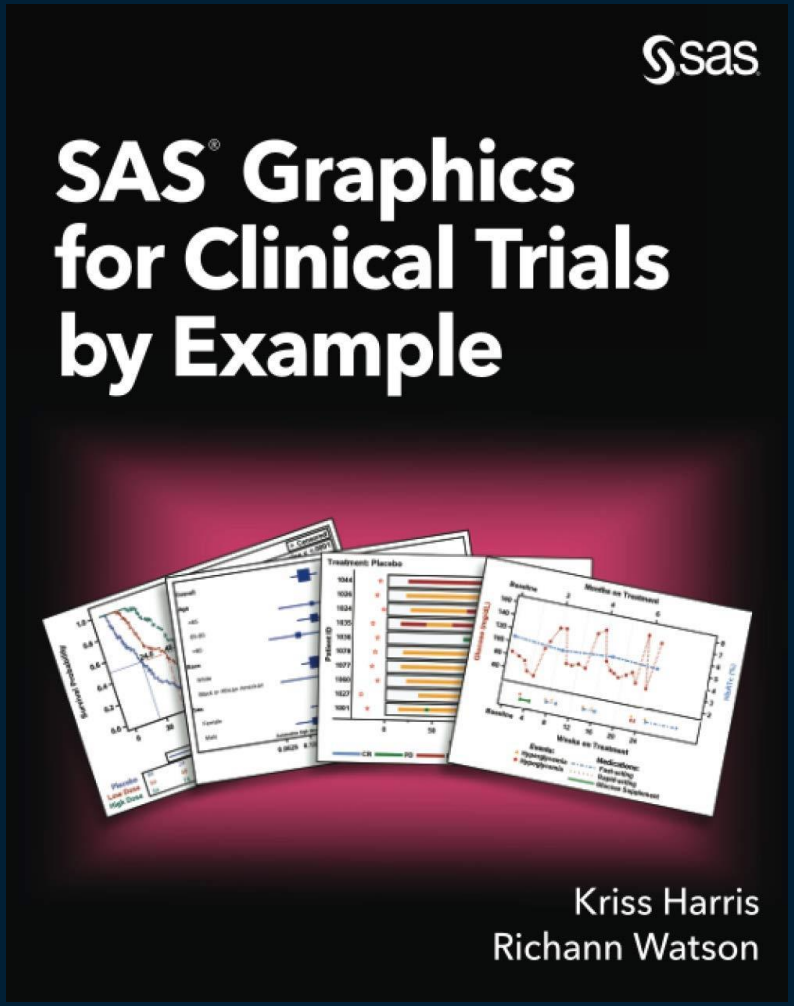
Combine SAS & R in Shiny – interactive Analysis

- One of the advantages in R is the R-Shiny packages.
- It allows us to create “web-based” interactive analysis
- Interactive analysis means to have more details on the graph data, i.e. clicks to see the subject data level information

- In some cases, we still need to generate the SAS programs for our deliverables…
- Can we then combine both systems and have the SAS code generated “automatically”?

Questions?

For More Information on Graphs



Contact Information

Endri Elnadav

Expert Data Analyst (Australia)
endri0501@gmail.com

https://www.linkedin.com/in/endri0501/

Kriss Harris

WCG Clinical Endpoint Solutions
italjet125@yahoo.com

https://www.linkedin.com/in/krissharris/

https://twitter.com/krissharris