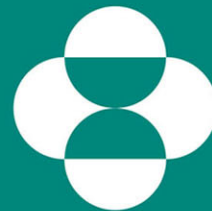


Constructing Multilingual Visualization Toolset for Clinical Study Reports and Submission

PHUSE US Connect 2024



MERCK

INVENTING FOR LIFE

February 2024

Bingjun Wang, Jane Liao, Yulia Sidi, Nan Xiao

Merck & Co., Inc., Rahway, NJ, USA

Agenda

- Introduction
- Framework of visualization toolset
- Visualization standards and branding
- Graphic repository
 - Standard programs
 - Customizable programs
 - Visualization gallery
 - Visualization training
 - Benefits
- Conclusion

Introduction

- Increasing demand for insightful data visualization
- Growing trend of using multiple programming languages to support Analysis & Reporting (A&R)
- Enhancing the capabilities to deliver the most impact and value

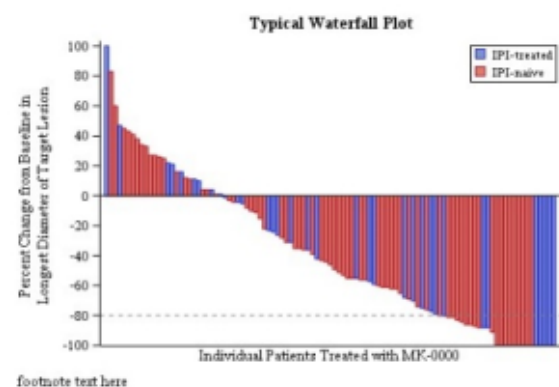
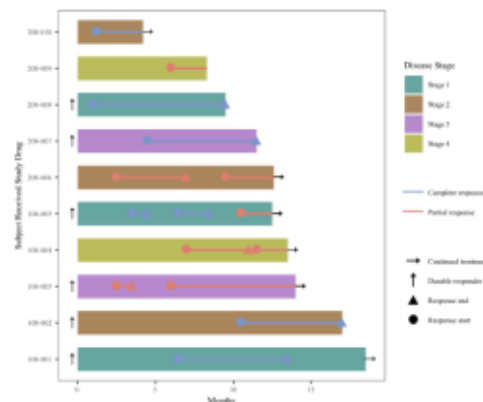
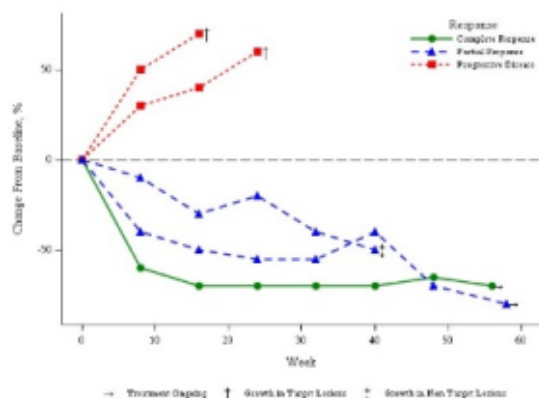
Framework of Visualization Toolset

- Visualization standards: set the standardization, principle, and recommendation
- Branding: implementation of standardized figure layout and theme
- Standard programs: standard SAS macros, internal R packages to create commonly used figures
- Customizable programs: template programs to provide flexibility beyond standard programs
- Visualization gallery: collection of commonly used figures with examples
- Internal training: upskill staff with multilingual data visualization techniques



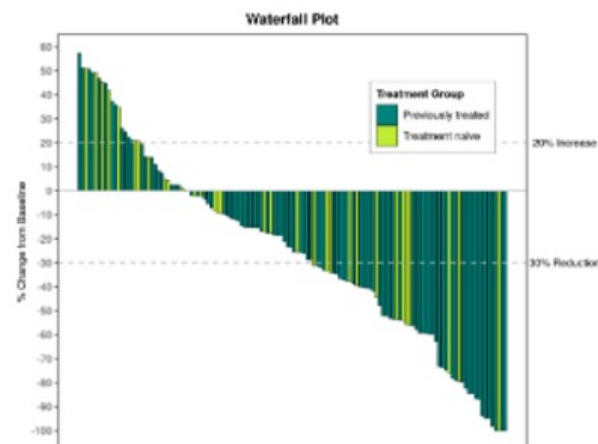
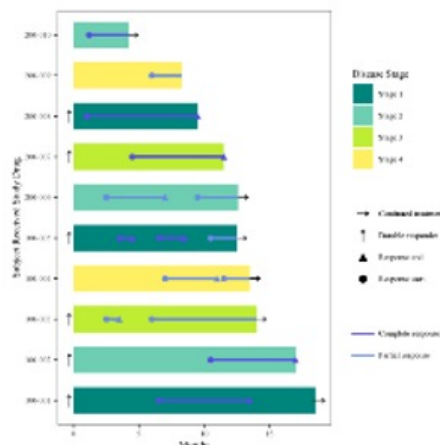
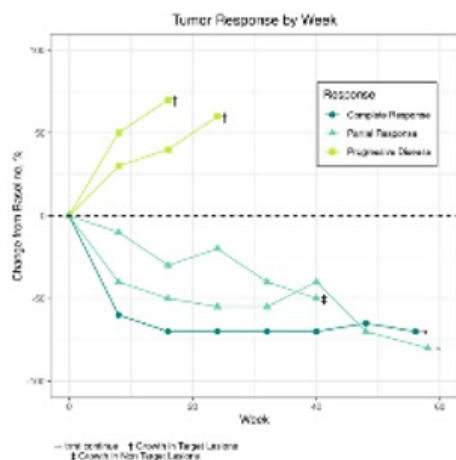
Visualization Standards and Branding

- The manual of TLF formatting standards for deliverables sets the foundation for Visualization Standards
- Lack of uniform standards and branding results in a noticeable inconsistency across study deliverables
 - Below examples, colors are not standardized
 - Difficult to review and compare same information across deliverables within the same study
 - Colors selected are not color-blind friendly (red, green)
 - No branding, colors are randomly used



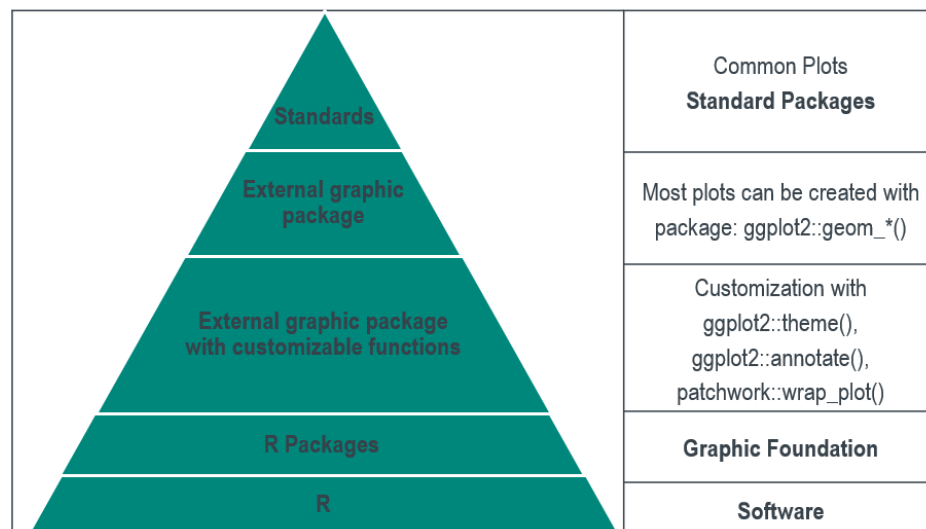
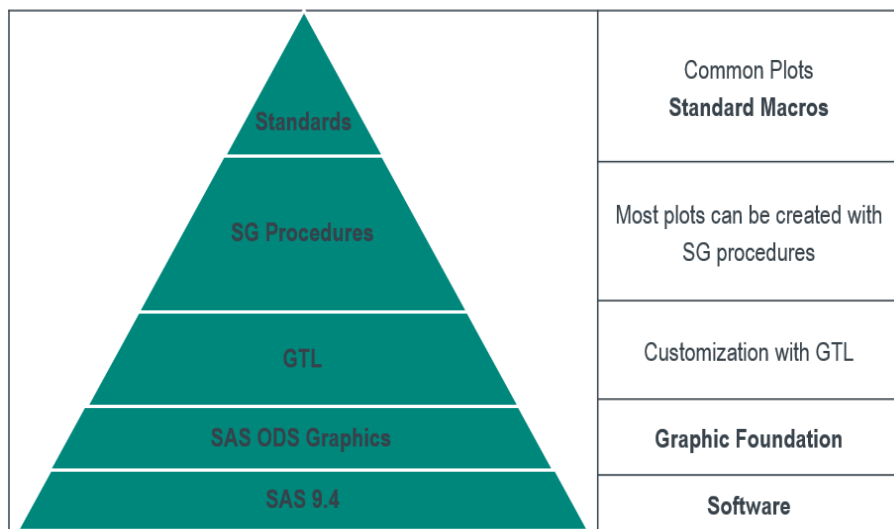
Visualization Standards and Branding

- Branding: Ensures recognizability and consistency in a professional context
 - Use of company branding color
 - Deliverable is easily recognized both internally and externally
 - Improves the consistency of the visual appearance of the output
 - Graph wall frame is used
 - Symbols are consistent, i.e., filled circle, filled triangle
 - Legend box is used when display inside the figure



Graphic Repository

- Multilingual toolset is developed using SAS and R: Graphic Repository
 - Software to be used: SAS, R
 - SAS ODS Graphics, R Packages provide foundation
 - SAS GTL, R package customization functions enable customization
 - Frequently requested figures: SAS SG procedures, R `ggplot2::geom_*()`
 - Standardized common plots: SAS macros, R internal standard packages



Standard Programs

- Enforce standardized formatting of the output
 - SAS utility standard macro: %graph0param0init
 - R package: mkthemes
- Are developed to use across projects
 - SAS: standard macros: %kmplot, %forestplot, %rainfallplot, %volcanoplot
 - R: internal package: mkvis includes KM plot, forest plot, rainfall plot, volcano plot, box plot, line plot

Customizable Programs

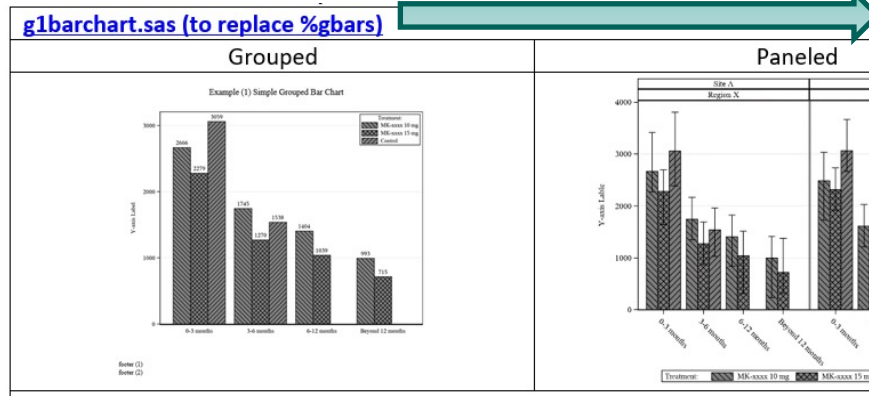
- Standard programs don't meet all the stakeholder needs
- Figures cannot be standardized completely
- Customizable programs provide better starting point
 - When reporting needs are different from standardized figure
 - Can be easily adopted to address project specific requirements
- SAS: Template programs with GTL and SG procedures
- R: Template programs with ggplot2 package

Visualization Gallery

- Showcases an easy-to-follow solution to generate figures

- SAS:

[g1barchart.sas \(to replace %gbars\)](#)



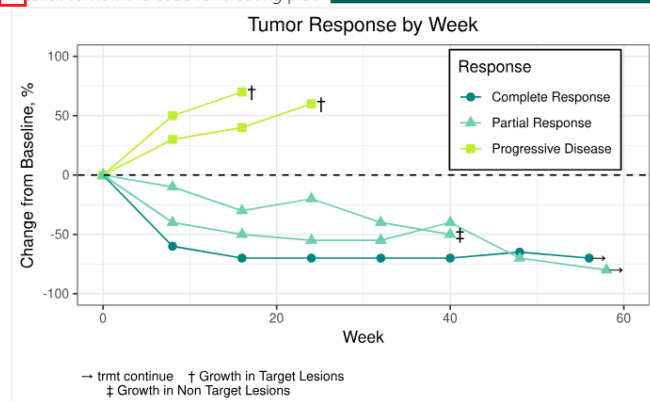
```
options orientation=portrait nodate nonumber
          topmargin=6pm1.in bottommargin=6pm2.in leftmargin=6pm3.in rightmargin=6pm4.in;
* Define an escape character for soft return when multiple titles/footnotes;
* escapechar+n or escapechar+newline between text will display title/footnote in a sepa
ods escapechar="%std_escapechar.";
ods _all_ close;
ods graphics / reset=index outputfmt=png imagename="g1barchart0example1" noborder width=5
ods rtf file="%_qpath./g1barchart0example1.rtf" dpi=4rtf_dpi. style=wmastyle bodytitle no
```

```
proc sgplot data=lptda.g1barchart1;
  title "Example (1)&std_escapechar. Simple Grouped Bar Chart";
  vbarparm category=period response=mean / group=treated groupdisplay=cluster datalabe
  xaxis display=(noticks nolabel) offsetmin=0.15 offsetmax=0.15;
  yaxis grid label="Y-axis Label";
  keylegend / location=inside position=topright across=1 title="Treatment:.";
  format period prdfmt. treated trtfmt.;
  footnote1 j=1 "Footer (1)";
  footnote2 j=1 "Footer (2)";
run;
title; footnote;
```

```
* ods to close the file destination;
ods graphics off;
ods rtf close;
```

- R:

Click to view the code for creating plot



Click to view the code for creating plot

Copy to clipboard

```
# generate the spider plot
p1 <-
  spider0 %>%
  ggplot(aes(x = Week, y = Change, group = Subject, color = Response)) +
  geom_point(aes(shape = Response), size = 2.5) +
  geom_line() +
  scale_color_manual(name = "Response",
    values = c("#00857C", "#6ECEB2", "#BFED33")) +
  geom_abline(slope = 0, intercept = 0, col = "black", lty = 2) +
  coord_cartesian(xlim = c(0,60), ylim = c(-100,100)) +
  geom_text(aes(label = Stlabel), nudge_x = 1.2, color = "black", na.rm = TRUE)
  labs( y = "Change from Baseline, %", x = "Week",
    title = "Tumor Response by Week",
    caption = "\n \u2192 trmt continue \u2020 Growth in Target Lesions
\u2021 Growth in Non Target Lesions ") +
  theme_bw()+
  theme(legend.position = c(.8,.75),
    legend.background = element_rect(linetype = "solid", color = "black"),
    plot.caption = element_text(hjust = 0),
    plot.title = element_text(hjust = 0.5))
```

```
# to display the plot, un-comment the following statement
# p1
```

Visualization Training

- Help to gain the unique skillset of SAS and R graphic programming
- Tailored to address specific visualization needs in our organization
- Shorten learning curve and provide smooth learning experience
- Training covers
 - Standards and recommendations for visualization deliverables
 - Best practices for developing programs/mockups
 - SAS
 - Introduce the fundamental graphic knowledge: SG procedures
 - Visualization gallery
 - Tools help to learn: GTL and ODS Graphic Designer (OGD)
 - Standard macros
 - R
 - Introduce the fundamental graphic knowledge: ggplot2, patchwork packages
 - Visualization gallery
 - Tools help to learn: esquisse package
 - Internal R visualization package: mkvis

Benefits of Graphic Repository

- Comprehensive hub for qualified graphic programs
- Streamlined approach to the creation and customization of plots
- Ensuring standardized deliverables across various projects and therapeutic areas
- Simplify the accessibility of a diverse array of programming tools
- Promote uniformity in data visualization

Benefits of SAS Graphic Repository

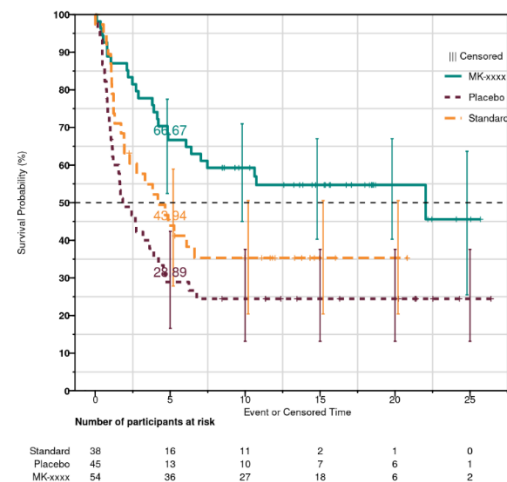
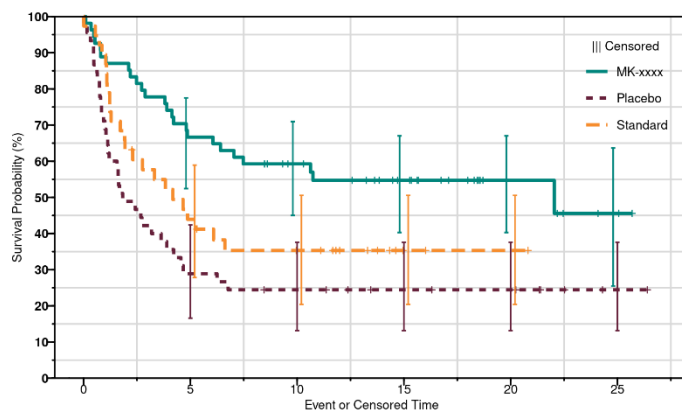
- SAS is an off the shelf product
- No need for in-house qualification
 - Widely accepted - procedures, functions already validated by the vendor
- Integrates seamlessly with the A&R workflow since most tables and listings are generated using SAS

Benefits of R Graphic Repository

- Leverage the advanced capabilities that R offers to complement the already existing SAS graphic repository
- Open-source: Extensive range of packages allow elegant solutions for various tasks
- R Markdown and knitr are powerful tools for creating dynamic, interactive documentation
 - Integration of R code directly into documents
 - Automation of data process ensuring that results are always up-to-date and reproducible
 - Conversion of documentation formats enhancing the accessibility and sharing of findings

Benefits of R Graphic Repository

- ggplot2 object offers a straightforward and flexible way to customize plots
- Each customization is added incrementally using the '+' operator without modifying the original program, ex: figure on the right side only need to add new components based on the figure on the left side (no need to re-create)



Conclusion

- Selecting a statistical analysis software for clinical study reports and submissions
 - Consider the unique advantages and limitations of each tool

	SAS (Commercial product)	R (Open-source product)
PROs	• Offers stability, reliability • Widely accepted Software Development Life Cycle (SDLC) documentation	• Community-driven development • Rapid integration of advanced methods • R Markdown ensures that documentation is consistently updated, reflecting the current status of content
CONs	• High cost • Slower adoption of advanced methods • Limitations in interactive figure generation	• Require internal qualification to guarantee package quality • Frequent updates in R and its packages

THANK YOU



MERCK

