PharmaSUG 2025 - Paper DV-337

Combined Waterfall and Swimmer Plot using R for Visualization of Tumor Response Data

Akshata Salian, Ephicacy Lifescience Analytics Pvt Ltd, Bengaluru, India

ABSTRACT

Visualizing tumor response data is crucial in oncology studies to assess the effectiveness of treatments and monitor patient outcomes. The waterfall plot typically displays the percentage change in tumor size from baseline for each patient and the Swimmer plot provides a detailed, patient-level representation of treatment timelines showing duration of treatment and clinical events over time.

The combined visualization integrates key elements of both plots and offers a more comprehensive understanding of treatment outcomes. Tumor progression can be directly linked to the length of therapy, offering insights into how response evolves over time. It also facilitates comparisons between subgroups (e.g., treatment arms) by incorporating multiple data dimensions into the plot. It also reduces the need for multiple plots, making it easier to present and interpret data at first sight.

In SAS, the option is available to combine them together for an effective interpretation. This paper explains a method for creating a combined Waterfall and Swimmer plot using open-source platform, enabling a unified visualization of tumor shrinkage and treatment duration. Using publicly available tumor response datasets, this paper demonstrates the step-by-step implementation of combined plot in R, including data preprocessing, customization, and annotation. The combined plot incorporates tumor response as bars and overlays patient timelines, treatment related events, and response status as lines and markers. To demonstrate this, the paper uses publicly available R packages, such as *ggplot2*, *dplyr* and *ggpubr*, and provides a reproducible workflow for statistical programmers.

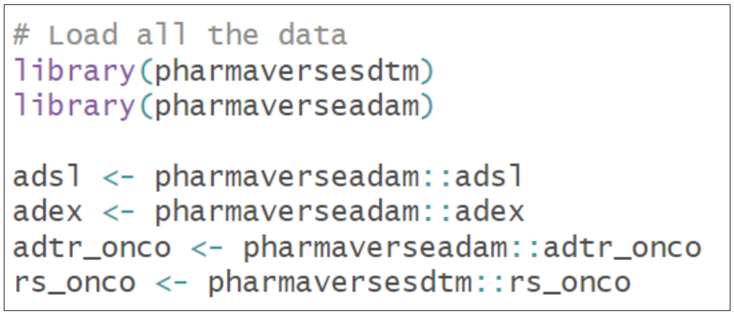
INTRODUCTION

In oncology studies, there are several ways to determine treatment efficacy and safety. Considering the complexity of the study, one needs to look at several parameters (like overall survival, progression-free survival, tumor response) and create diverse types of graphs to understand a complete picture of the trial. Two such graphs are the waterfall plot and the swimmer plot, which are generally created as separate graphs. Waterfall plots visually summarize the response of individual patients to treatment by displaying the percentage change in tumor size from baseline. Swimmer plots are used to visually represent individual patient's tumor response and treatment history, displaying aspects like treatment duration, time to response, response type (complete/partial), all at a glance. By combining these two, the aim is to enhance the clarity and depth of data interpretation, ultimately contributing to more informed clinical decisions and improved patient care.

DATA PREPARATION

SOURCE DATA

This paper utilizes data from the pharmaverse R packages, such as *pharmaversesdtm* and *pharmaverseadam*. Below listed datasets have been used in this paper.



```r
# Load all the data
library(pharmaversesdtm)
library(pharmaverseadam)

adsl <- pharmaverseadam::adsl
adex <- pharmaverseadam::adex
adtr_onco <- pharmaverseadam::adtr_onco
rs_onco <- pharmaversesdtm::rs_onco
```

R code for loading source data

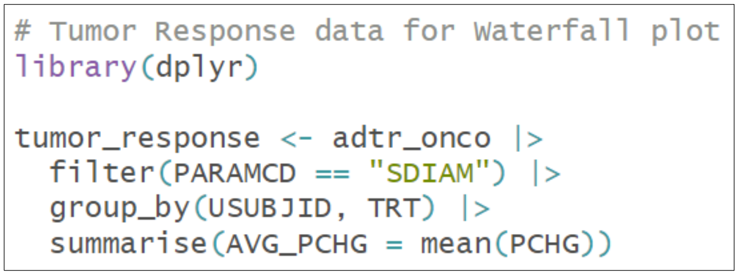
- ADSL - This is the Subject Level analysis dataset which contains one record per subject with variables describing demographics, baseline characteristics, treatment groups, and other key information.
- ADEX - This is the Analysis Exposure dataset which contains information about study drug exposure. This dataset is used to store and organize data related to a subject's exposure to a treatment or intervention, such as drug dosage, duration, and timing.
- ADTR_ONCO - This is the Analysis data for Tumor Response and is used to analyze tumor response assessments in oncology clinical trials. It is typically derived from the SDTM TU (Tumor Identification), TR (Tumor Response), and RS (Response) datasets.
- RS_ONCO - This is the Response dataset which contains data related to disease response assessments in clinical trials. This data holds the results of disease response evaluations, typically determined from measurements or assessments captured in the tumor response and is used to assess the overall response or specific response for target, non-target, or new lesions.

DATA PROCESSING

During the data processing step, datasets necessary for plotting the waterfall and swimmer graphs are created. The *dplyr* package available in R is used for data preparation. Below are the different data outputs generated for analysis. The R code and snapshots of corresponding output data are shown.

Tumor Response Data

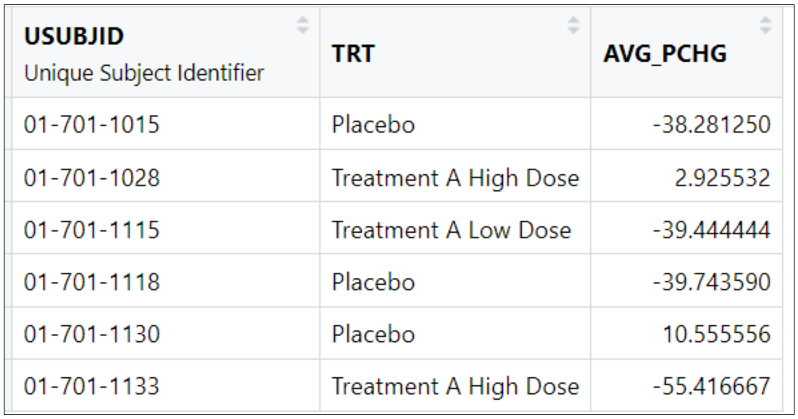
To create the waterfall plot, information on the percentage change in tumor size from baseline is necessary. This information is available in the ADTR_ONCO dataset. For this paper, the data filtering criteria considered is based on parameter ‘Target Lesions Sum of Diameters by Investigator,’ which is identified by PARAMCD = “SDIAM”. This data is summarized to compute the average PCHG (Percent Change from Baseline) for each subject (USUBJID) and treatment (TRT).



```r
# Tumor Response data for Waterfall plot
library(dplyr)

tumor_response <- adtr_onco |>
  filter(PARAMCD == "SDIAM") |>
  group_by(USUBJID, TRT) |>
  summarise(AVG_PCHG = mean(PCHG))
```

R code for Tumor Response Data



USUBJID Unique Subject Identifier	TRT	AVG_PCHG
01-701-1015	Placebo	-38.281250
01-701-1028	Treatment A High Dose	2.925532
01-701-1115	Treatment A Low Dose	-39.444444
01-701-1118	Placebo	-39.743590
01-701-1130	Placebo	10.555556
01-701-1133	Treatment A High Dose	-55.416667

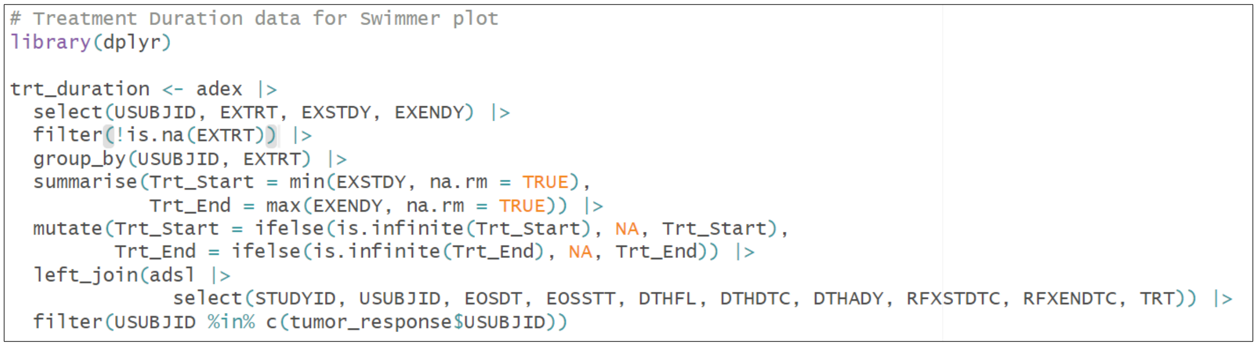
Tumor Response Data

Treatment Duration Data

To create the swimmer plot, it is necessary to compute the treatment start and end dates for each subject and treatment. This determines the treatment duration, which is then used for plotting the timeline bars of the swimmer. The ADEX dataset which contains the Study Day for the start and end of treatment, is utilized for this calculation, and the data is summarized at the subject and treatment level.

Other vital information like death flag and end of study status are usually present in ADSL data. This data is essential for adding markers to the swimmer plot for events like death, discontinued treatment, etc. Hence, required variables from ADSL are added to this data.

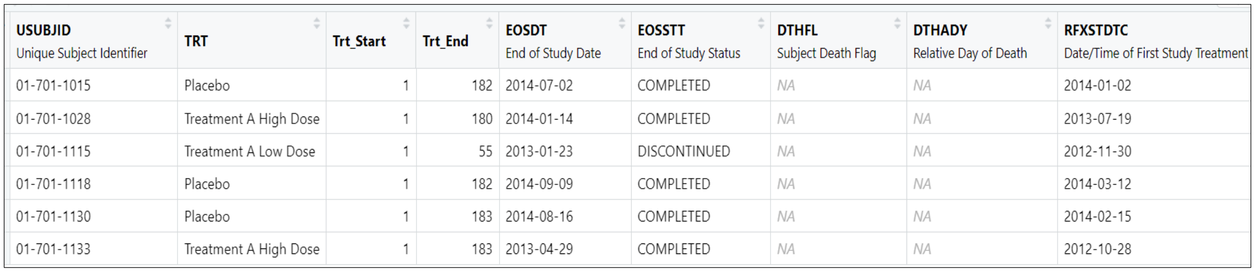
Finally, the data is subset to include only the required subjects identified in the tumor_response dataset.



```
# Treatment Duration data for Swimmer plot
library(dplyr)

trt_duration <- adex |>
  select(USUBJID, EXTRT, EXSTDY, EXENDY) |>
  filter(!is.na(EXTRT)) |>
  group_by(USUBJID, EXTRT) |>
  summarise(Trt_Start = min(EXSTDY, na.rm = TRUE),
            Trt_End = max(EXENDY, na.rm = TRUE)) |>
  mutate(Trt_Start = ifelse(is.infinite(Trt_Start), NA, Trt_Start),
         Trt_End = ifelse(is.infinite(Trt_End), NA, Trt_End)) |>
  left_join(adsl |>
              select(STUDYID, USUBJID, EOSDT, EOSSTT, DTHFL, DTHDTC, DTHADY, RFXSTDTC, RFXENDTC, TRT)) |>
  filter(USUBJID %in% c(tumor_response$USUBJID))
```

R code for Treatment Duration Data

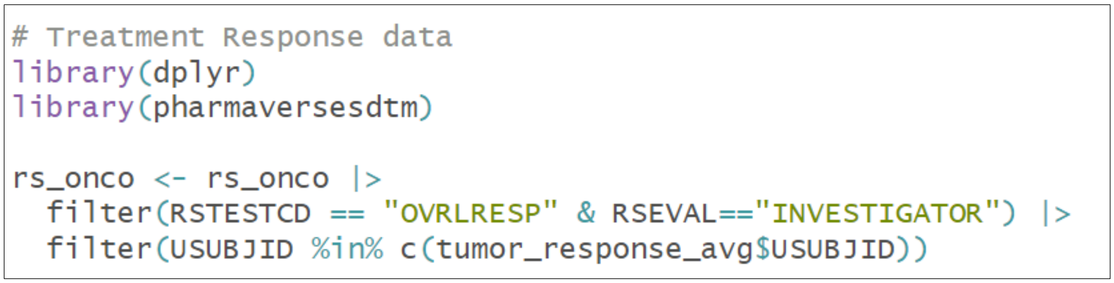


USUBJID Unique Subject Identifier	TRT	Trt_Start	Trt_End	EOSDT End of Study Date	EOSSTT End of Study Status	DTHFL Subject Death Flag	DTHADY Relative Day of Death	RFXSTDTC Date/Time of First Study Treatment
01-701-1015	Placebo	1	182	2014-07-02	COMPLETED	NA	NA	2014-01-02
01-701-1028	Treatment A High Dose	1	180	2014-01-14	COMPLETED	NA	NA	2013-07-19
01-701-1115	Treatment A Low Dose	1	55	2013-01-23	DISCONTINUED	NA	NA	2012-11-30
01-701-1118	Placebo	1	182	2014-09-09	COMPLETED	NA	NA	2014-03-12
01-701-1130	Placebo	1	183	2014-08-16	COMPLETED	NA	NA	2014-02-15
01-701-1133	Treatment A High Dose	1	183	2013-04-29	COMPLETED	NA	NA	2012-10-28

Treatment Duration Data

Response Data

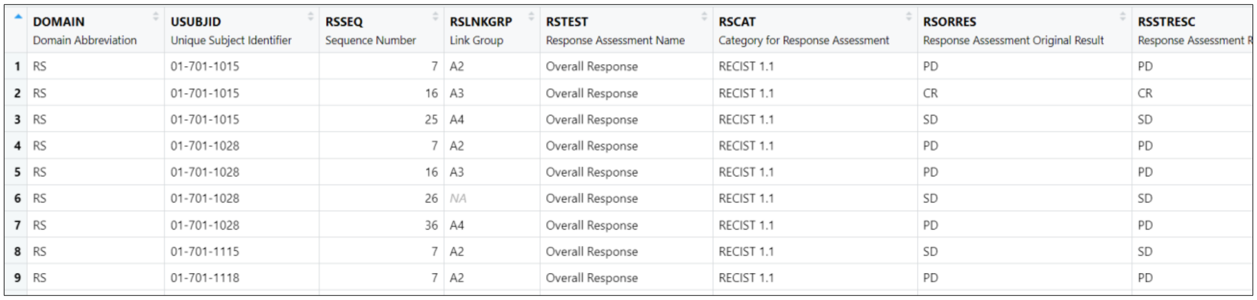
The observed response throughout the treatment duration is available in the RS_ONCO data. Hence this data is filtered for the relevant parameters and subjects.



```r
# Treatment Response data
library(dplyr)
library(pharmaversesdtm)

rs_onco <- rs_onco |>
  filter(RSTESTCD == "OVRLRESP" & RSEVAL=="INVESTIGATOR") |>
  filter(USUBJID %in% c(tumor_response_avg$USUBJID))
```

R code for Response Data

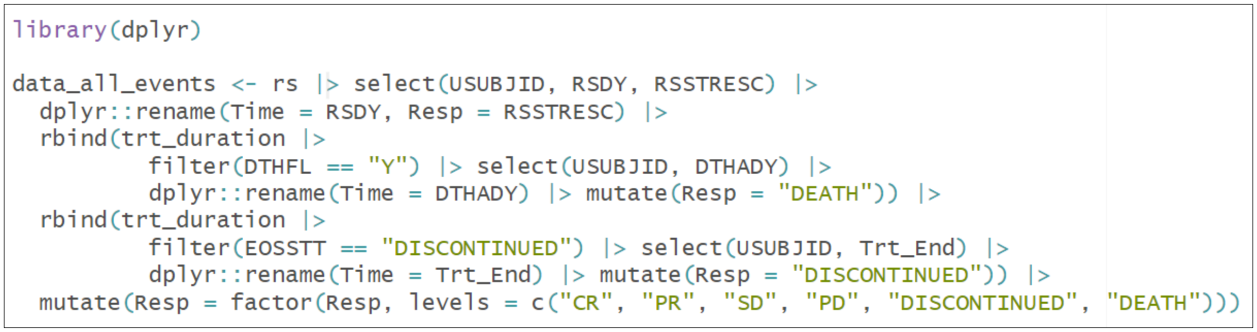


	DOMAIN Domain Abbreviation	USUBJID Unique Subject Identifier	RSSEQ Sequence Number	RSLNKGRP Link Group	RSTEST Response Assessment Name	RSCAT Category for Response Assessment	RSORRES Response Assessment Original Result	RSSTRESC Response Assessment R
1	RS	01-701-1015	7	A2	Overall Response	RECIST 1.1	PD	PD
2	RS	01-701-1015	16	A3	Overall Response	RECIST 1.1	CR	CR
3	RS	01-701-1015	25	A4	Overall Response	RECIST 1.1	SD	SD
4	RS	01-701-1028	7	A2	Overall Response	RECIST 1.1	PD	PD
5	RS	01-701-1028	16	A3	Overall Response	RECIST 1.1	PD	PD
6	RS	01-701-1028	26	NA	Overall Response	RECIST 1.1	SD	SD
7	RS	01-701-1028	36	A4	Overall Response	RECIST 1.1	PD	PD
8	RS	01-701-1115	7	A2	Overall Response	RECIST 1.1	SD	SD
9	RS	01-701-1118	7	A2	Overall Response	RECIST 1.1	PD	PD

Response Data

Data for Responses and Events

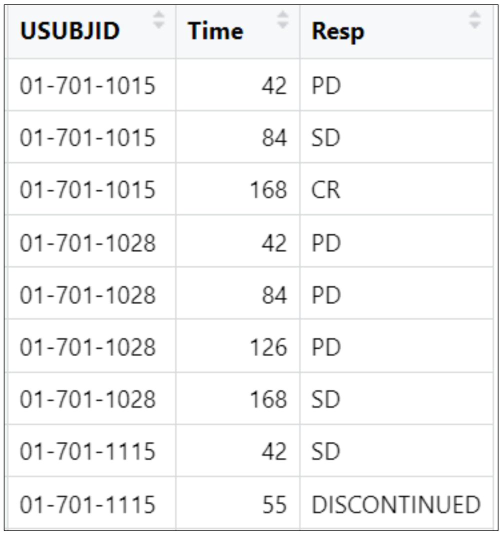
For ease of use, an additional step one needs to do is to combine data for treatment response, death flag and discontinuation flag. The ‘shape’ attribute is associated with these variables, and hence, this step is essential in terms of pooling all such variables together. Below is the code for combining all the parameters associated with shape variable for the plot.



```r
library(dplyr)

data_all_events <- rs |> select(USUBJID, RSDY, RSSTRESC) |>
  dplyr::rename(Time = RSDY, Resp = RSSTRESC) |>
  rbind(trt_duration |>
          filter(DTHFL == "Y") |> select(USUBJID, DTHADY) |>
          dplyr::rename(Time = DTHADY) |> mutate(Resp = "DEATH")) |>
  rbind(trt_duration |>
          filter(EOSSTT == "DISCONTINUED") |> select(USUBJID, Trt_End) |>
          dplyr::rename(Time = Trt_End) |> mutate(Resp = "DISCONTINUED")) |>
  mutate(Resp = factor(Resp, levels = c("CR", "PR", "SD", "PD", "DISCONTINUED", "DEATH")))
```

R code for Data for Responses and Events



USUBJID	Time	Resp
01-701-1015	42	PD
01-701-1015	84	SD
01-701-1015	168	CR
01-701-1028	42	PD
01-701-1028	84	PD
01-701-1028	126	PD
01-701-1028	168	SD
01-701-1115	42	SD
01-701-1115	55	DISCONTINUED

Data for Responses and Events

PLOT CONSTRUCTION

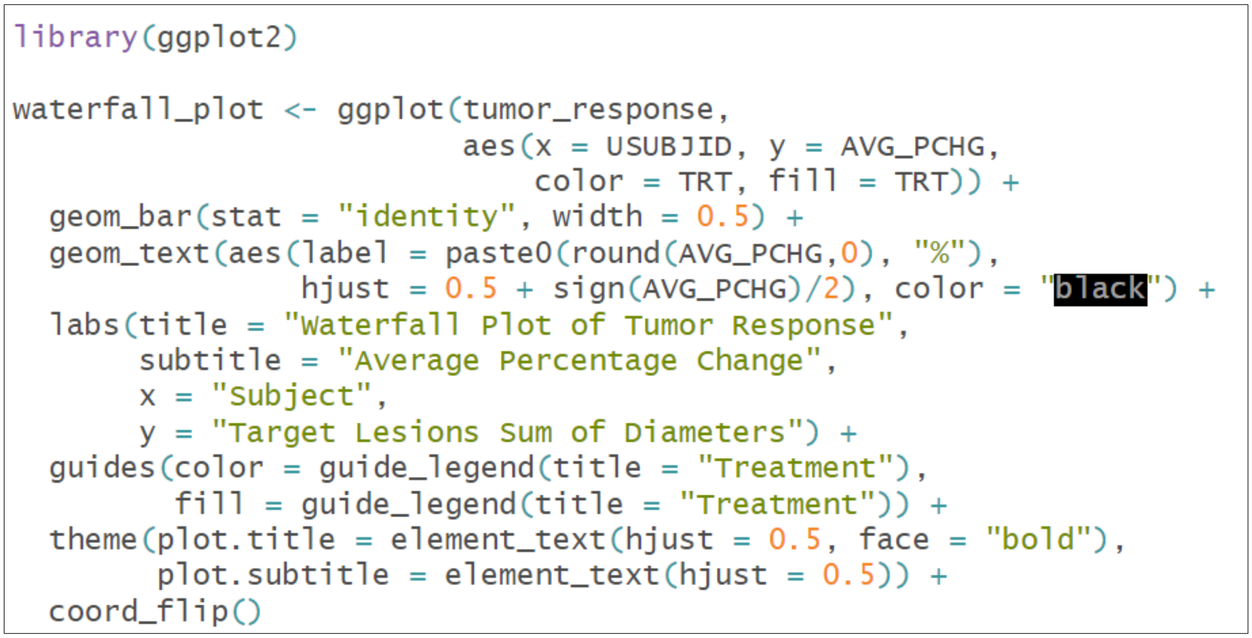
For creating a combined waterfall and swimmer plot, lets first create the individual plots using *ggplot2* and then combine them using *ggpubr*.

PLOTTING PACKAGE OVERVIEW

- ***ggplot2***: This is the most popular data visualization packages in R, designed for creating complex, multi-layered plots easily. It is based on the Grammar of Graphics concept, which allows users to build plots step by step by adding different layers. This package is used to create the individual waterfall and swimmer plots.
- ***ggpubr***: This package is an extension of *ggplot2* that simplifies creating publication-ready plots with minimal coding. It provides easier syntax, built-in statistical analysis, and better aesthetics for scientific research and reports. This package also helps in combining multiple plots together and is used in this paper to generate the combined plot.
- ***cowplot***: This is an alternative to *ggpubr*. This package is also useful for combining multiple plots into a single figure. This package also offers good aesthetics, alignment, and flexibility.
- ***gridExtra***: This package is another alternative to combine multiple plots into one. This also allows additional options for arranging non-ggplot objects like table as well. However, not as flexible in terms of aesthetics and alignment.

CREATING WATERFALL PLOT

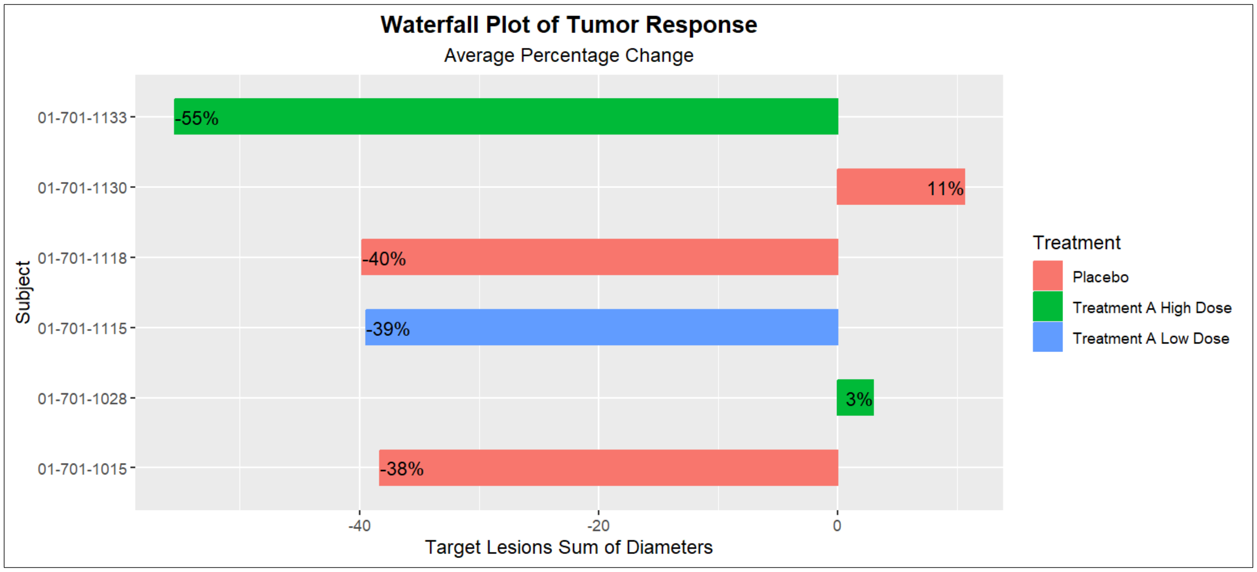
First step is to create the standalone waterfall plot. Below is the code for generating waterfall plot using tumor_response data and *ggplot2* package. The function used for creating the bars is *geom_bar()*, and *geom_text()* is used for adding the label values, *labs()* used for the titles and *theme()* for customizing the different non-data aspects of the plot.



```r
library(ggplot2)

waterfall_plot <- ggplot(tumor_response,
                         aes(x = USUBJID, y = AVG_PCHG,
                             color = TRT, fill = TRT)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_text(aes(label = paste0(round(AVG_PCHG,0), "%"),
                hjust = 0.5 + sign(AVG_PCHG)/2), color = "black") +
  labs(title = "Waterfall Plot of Tumor Response",
       subtitle = "Average Percentage Change",
       x = "Subject",
       y = "Target Lesions Sum of Diameters") +
  guides(color = guide_legend(title = "Treatment"),
         fill = guide_legend(title = "Treatment")) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5)) +
  coord_flip()
```

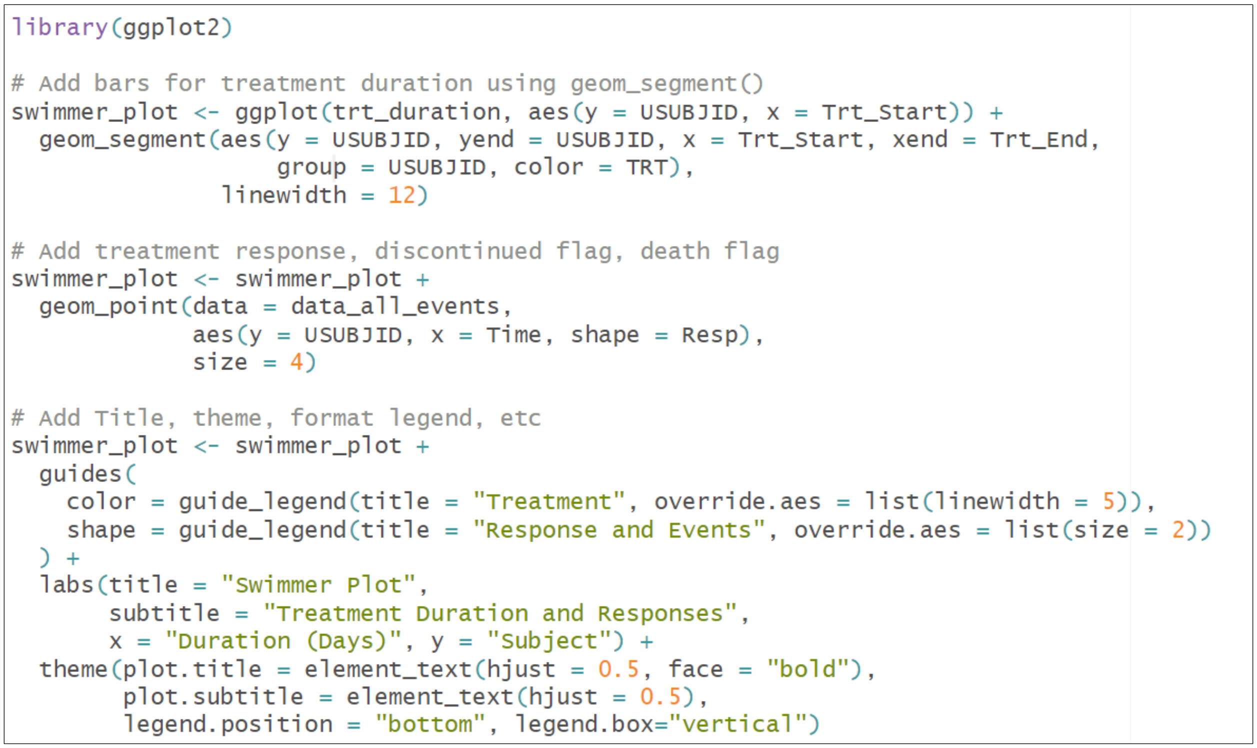
R code for Waterfall Plot



Waterfall Plot Output

CREATING SWIMMER PLOT

Next step is to create the standalone swimmer plot. Below is the code for generating swimmer plot using trt_duration data and *ggplot2* package. trt_duration data provides duration of treatment for each subject which can be plotted as bars using the *geom_segment()* function. After which data_all_events can be used to add all the responses and events as points with different shapes using *geom_point()* function.



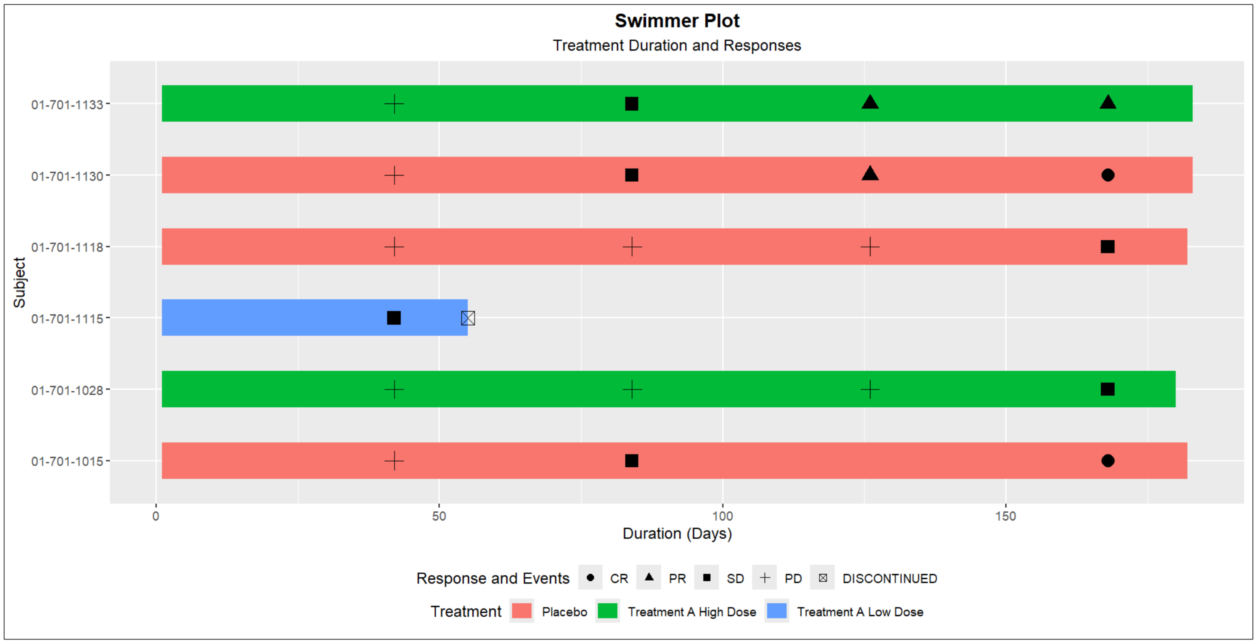
```
library(ggplot2)

# Add bars for treatment duration using geom_segment()
swimmer_plot <- ggplot(trt_duration, aes(y = USUBJID, x = Trt_Start)) +
  geom_segment(aes(y = USUBJID, yend = USUBJID, x = Trt_Start, xend = Trt_End,
                   group = USUBJID, color = TRT),
               linewidth = 12)

# Add treatment response, discontinued flag, death flag
swimmer_plot <- swimmer_plot +
  geom_point(data = data_all_events,
             aes(y = USUBJID, x = Time, shape = Resp),
             size = 4)

# Add Title, theme, format legend, etc
swimmer_plot <- swimmer_plot +
  guides(
    color = guide_legend(title = "Treatment", override.aes = list(linewidth = 5)),
    shape = guide_legend(title = "Response and Events", override.aes = list(size = 2))
  ) +
  labs(title = "Swimmer Plot",
       subtitle = "Treatment Duration and Responses",
       x = "Duration (Days)", y = "Subject") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5),
        legend.position = "bottom", legend.box="vertical")
```

R code for Swimmer Plot



Swimmer Plot Output

CHALLENGES OF COMBINING PLOT

Before beginning to create the combined plot, let's take a moment to think about possible challenges.

- **Plot alignment** – In order for the combined plot to be easily interpretable, their y axis should be aligned. For this reason, one must ensure that the number of subjects is the same for both the plots and that they are also arranged in the same order. Considering that subject filter is already applied in the code, this concern is addressed. Also, the subject ordering is retained as the default alphabetical order.

- **Bar width/thickness** – It's important to ensure that the bars thickness is the same for both the plots. This is managed in the code by adjusting the width option in *geom_bar()* for waterfall plot and linewidth option in *geom_segment()* for swimmer plot.

- **Common Y axis label** – Optionally, a common Y axis label can be added since they are same for both the plots.

- **Common legend** – Instead of duplicating the legends, it is crucial to have a common legend for the combined plot.

Considering all these points, let's proceed with the final plot creation.

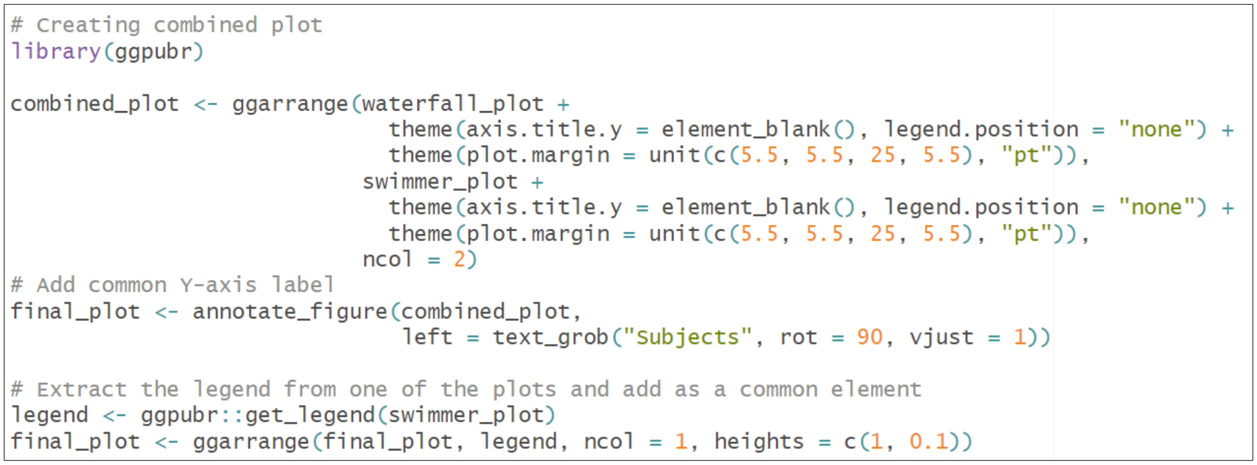
CREATING COMBINED PLOT

The final step is to combine the two plots using *ggpubr* package.

- The function *ggarrange()* can be used to add the different plots/components one wishes to combine. In this function, additional options can be specified like whether the plots should be arranged by column or row using ncol/nrow option.

- For adding a common Y axis label, the individual plot labels can be removed using theme() and then later added separately by using *annotate_figure()*. *annotate_figure()* can be used to add different types of annotations such as text, shapes, or images to the plot.

- Finally, a common legend can be added, by firstly extracting the legend part from the swimmer plot using *get_legend()* and then adding it to the combined plot using *ggarrange()*.

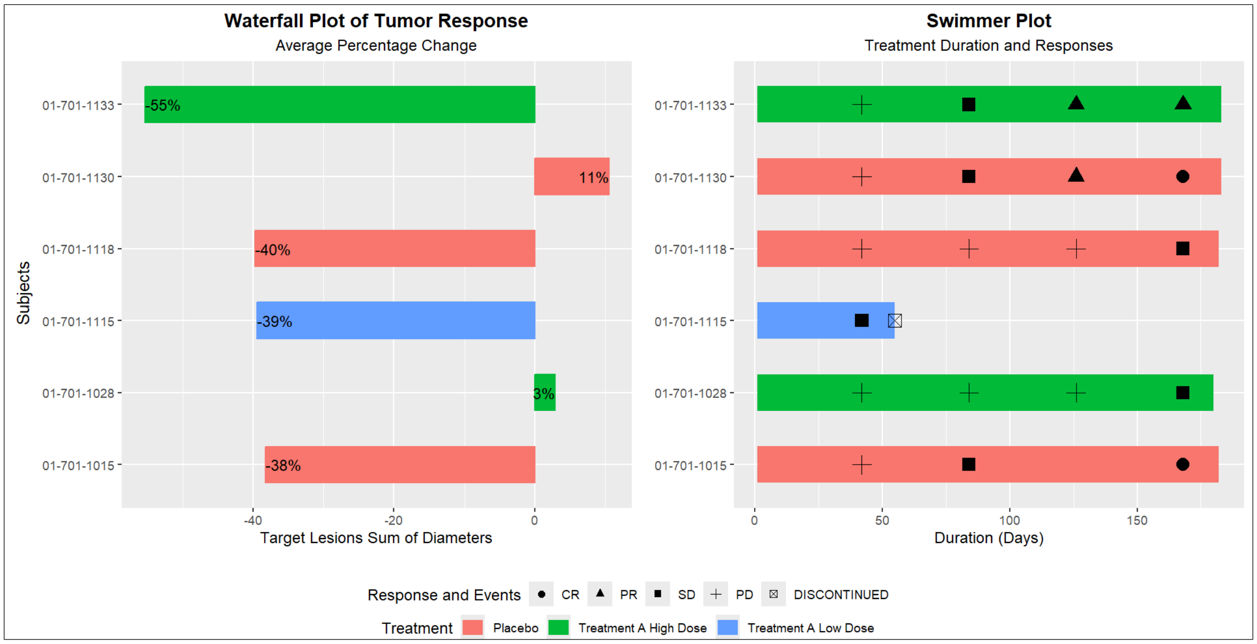


```r
# Creating combined plot
library(ggpubr)

combined_plot <- ggarrange(waterfall_plot +
                             theme(axis.title.y = element_blank(), legend.position = "none") +
                             theme(plot.margin = unit(c(5.5, 5.5, 25, 5.5), "pt")),
                           swimmer_plot +
                             theme(axis.title.y = element_blank(), legend.position = "none") +
                             theme(plot.margin = unit(c(5.5, 5.5, 25, 5.5), "pt")),
                           ncol = 2)
# Add common Y-axis label
final_plot <- annotate_figure(combined_plot,
                              left = text_grob("Subjects", rot = 90, vjust = 1))

# Extract the legend from one of the plots and add as a common element
legend <- ggpubr::get_legend(swimmer_plot)
final_plot <- ggarrange(final_plot, legend, ncol = 1, heights = c(1, 0.1))
```

R code for Combined Plot



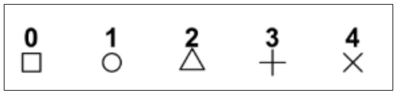
Combined Waterfall and Swimmer Plot

On the other hand, if one wants to explore other packages, there is *cowplot* and *gridExtra*. In *cowplot* package, there is *plot_grid()* function to combine the plots. And in *gridExtra* package, there is *grid.arrange()* function to combine the plots.

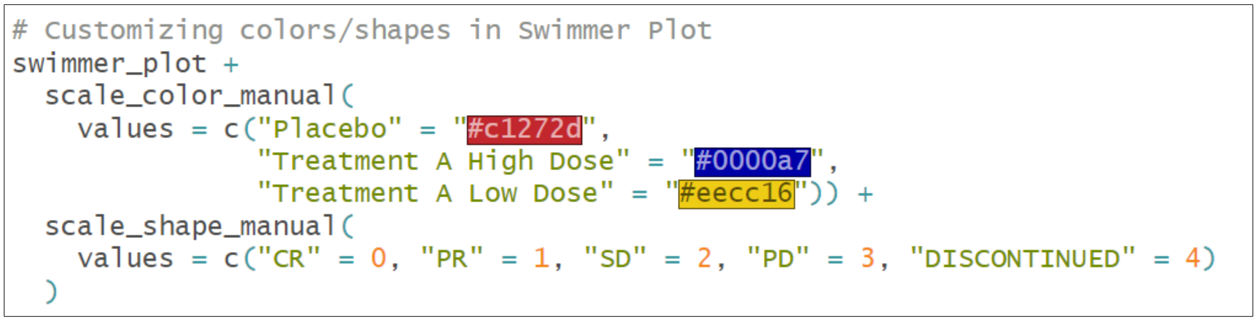
CUSTOMIZATION

More features can be added to the plot and customized to fit individual preferences to enhance the overall appearance of the plot.

- One can change/customize the colors of bars, the shapes/colors of treatment response to specific shapes/colors using *scale_color_manual()* and *scale_shape_manual()*.

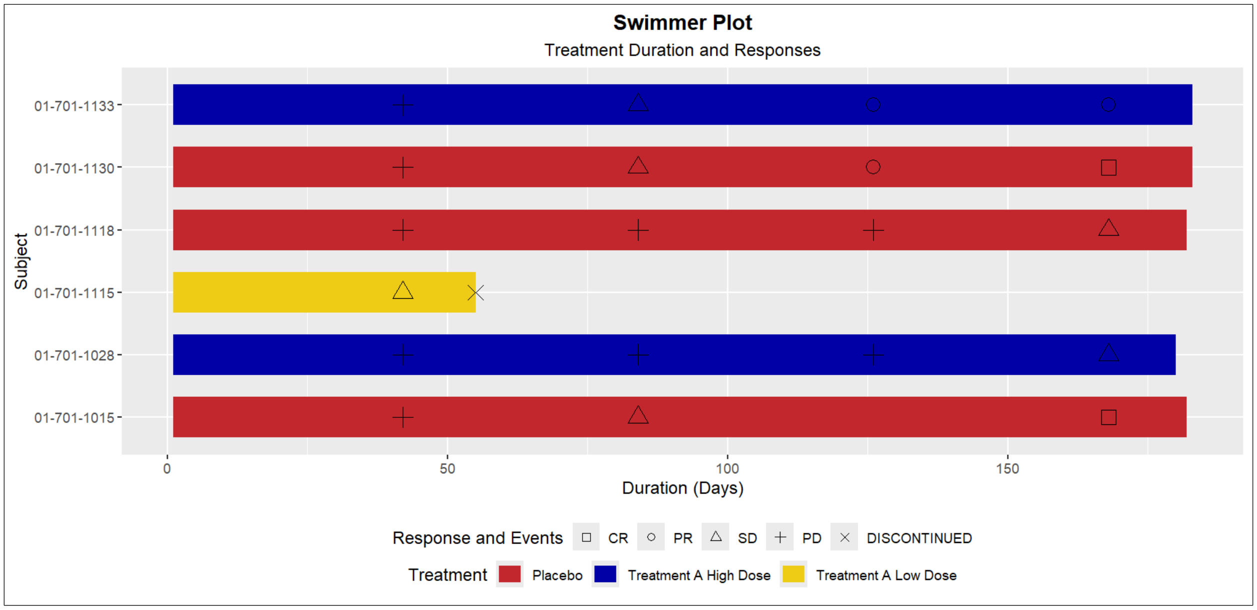


scale_shape_manual() – code with symbol



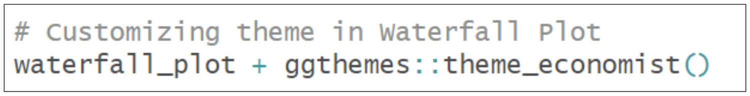
```
# Customizing colors/shapes in Swimmer Plot
swimmer_plot +
  scale_color_manual(
    values = c("Placebo" = "#c1272d",
               "Treatment A High Dose" = "#0000a7",
               "Treatment A Low Dose" = "#eecc16")) +
  scale_shape_manual(
    values = c("CR" = 0, "PR" = 1, "SD" = 2, "PD" = 3, "DISCONTINUED" = 4)
  )
```

R code for customizing colors/shapes



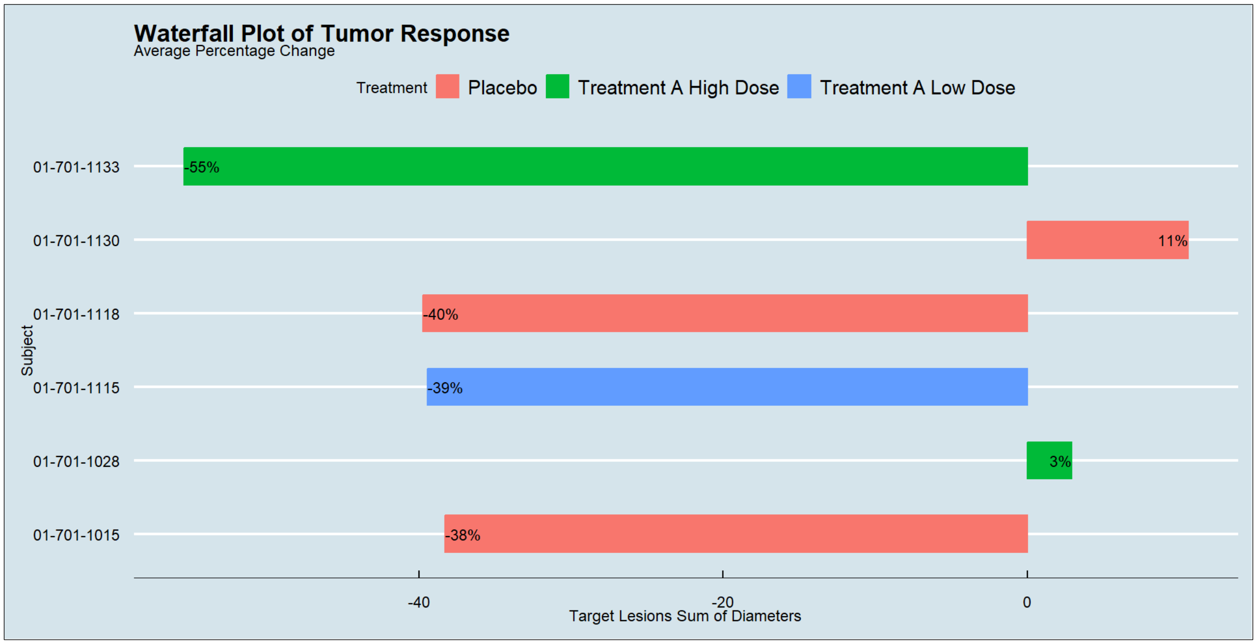
Plot with customized colors/shapes

- Different plot themes can be explored to enhance the visual appeal of the plot. *ggplot2* itself has several themes like *theme_gray()*, *theme_bw()*, *theme_minimal()*, *theme_classic()*. Furthermore, one can also use themes from other R packages like *ggthemes*, *hrbrthemes*.



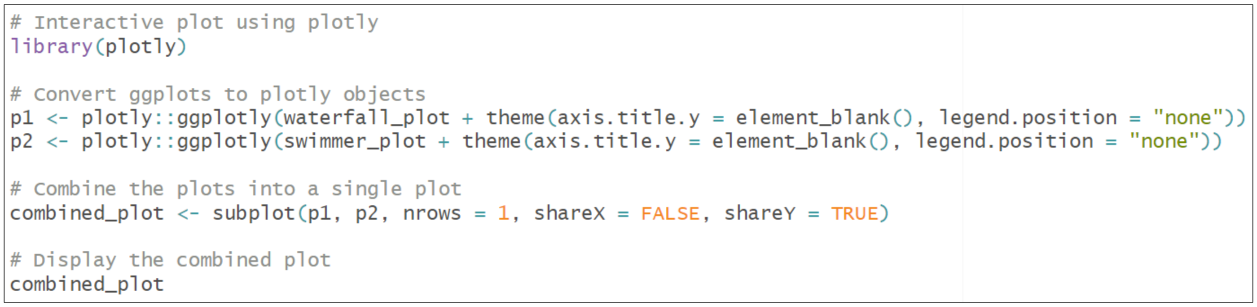
```
# Customizing theme in Waterfall Plot
waterfall_plot + ggthemes::theme_economist()
```

R code for changing plot themes



Plot with different theme

- The plot margins can be adjusted to add/remove space around the plot using *plot.margin()* option.
- Interactivity can be added to the combined plot by using *plotly* package. Below is the code for the same.



```
# Interactive plot using plotly
library(plotly)

# Convert ggplots to plotly objects
p1 <- plotly::ggplotly(waterfall_plot + theme(axis.title.y = element_blank(), legend.position = "none"))
p2 <- plotly::ggplotly(swimmer_plot + theme(axis.title.y = element_blank(), legend.position = "none"))

# Combine the plots into a single plot
combined_plot <- subplot(p1, p2, nrows = 1, shareX = FALSE, shareY = TRUE)

# Display the combined plot
combined_plot
```

R code for interactive combined plot code

ADVANTAGES

Combining waterfall and swimmer plots for tumor response visualization is highly beneficial in clinical oncology studies. Here are some key reasons why this combined approach can be considered important:

- **Comprehensive Data Representation**: A combined plot allows for the simultaneous visualization of tumor response (waterfall plot) and treatment duration (swimmer plot). This provides a more holistic view of the patient's response to treatment.
- **Enhanced Clarity**: By integrating both plots, clinicians can easily correlate the extent of tumor reduction with the duration of treatment and other clinical events. This reduces the need to cross-reference multiple charts, making the data easier to interpret.
- **Improved Decision-Making**: The combined visualization helps in identifying patterns and trends that might not be apparent when the data is presented separately. This can aid in making more informed decisions regarding treatment efficacy and patient management.
- **Efficient Communication**: For presentations and reports, a single combined plot can effectively communicate complex data to stakeholders, including clinicians, researchers, and regulatory bodies.

CHALLENGES

Below are some of the challenges in creating a combined waterfall and swimmer plot.

- **Data Complexity**: Tumor response data often involves multiple variables, such as tumor size reduction, treatment duration, and patient status. Integrating all this information into a single plot without losing clarity can be difficult.
- **Plot Design**: Ensuring that the combined plot is both informative and visually appealing requires careful design. The plot must clearly differentiate between the waterfall and swimmer components while maintaining a cohesive look.
- **Scalability**: As the number of patients increases, the plot can become cluttered, making it hard to interpret. Effective scaling and layout strategies are needed to handle large datasets.
- **Software Limitations**: Not all plotting libraries or tools support the creation of combined plots directly. This may require custom coding or the use of multiple tools to achieve the desired result.

- **Interactivity**: Adding interactive elements to the plot, such as tooltips or clickable regions, can enhance usability but also adds to the complexity of the implementation.

One of the possible ways to tackle the scalability challenge here is to create subset plots or faceted plots based on different criteria of interest like demographic variables, treatment, or response type. Additionally, the plot can be added to an R Shiny dashboard with multiple filtering options for effective data visualization.

CONCLUSION

Combined visualization of waterfall and swimmer plot allows for a more nuanced understanding of treatment efficacy and patient outcomes, facilitating better decision-making in clinical research and practice. R provides an easy way to combine plots using packages like *ggpubr*, *cowplot*, *gridExtra*. This same approach can be applied to build any type of combined plot using R. Some possible future directions are as follows:

- **Package Creation**: Developing packages in R that streamline the creation of combined waterfall and swimmer plots, making this approach more accessible to researchers with varying levels of programming expertise.
- **Interactive Visualizations**: Enhancing the plots with interactive features using libraries such as *plotly* and/or *shiny*, allowing users to explore the data dynamically and gain deeper insights.
- **User Feedback and Iteration**: Collecting feedback from end-users, including clinicians and researchers, to iteratively improve the visualization techniques and address any practical challenges encountered in real-world applications.

By pursuing these future directions, we can continue to refine and expand the utility of combined waterfall and swimmer plots, ultimately contributing to more effective and insightful oncology research.

REFERENCES

Mantage, Sanjay. 2018. “A Combined Waterfall and Swimmer Graph.” *Proceedings of the PhUSE EU Connect 2018*. Available at https://www.lexjansen.com/phuse/2018/dv/DV03.pdf

CDISC Submission Data Standards Team. “Study Data Tabulation Model Implementation Guide: Human Clinical Trials.” Available at https://www.cdisc.org/standards/foundational/sdtmig/sdtmig-v3-3/html

Ilya Krivelevich, Ran Xie, Simon Lin. 2020. “Efficacy ADaMs in Oncology – Step by Step (Dataset by Dataset)”. *Proceedings of the PhUSE US Connect 2020*. Available at https://www.lexjansen.com/phuse-us/2020/si/SI07.pdf

Sandra E. Kurtin, Rashida Taher. September 2020. “Clinical Trial Design and Drug Approval in Oncology: A Primer for the Advanced Practitioner in Oncology.” *Journal of the Advanced Practitioner in Oncology*.

Edoardo Mancini, Gayatri G, Kangjie Zhang, Pooja Kumari, Stefan Bundfuss, Zelos Zhu, Sadchla Mascary, Vladyslav Shuliar, Vinh Nguyen. “Package ‘pharmaversesdtm’.” January 23, 2025. Available at https://cran.r-project.org/web/packages/pharmaversesdtm/pharmaversesdtm.pdf.

Edoardo Mancini, Kangjie Zhang, Stefan Bundfuss, Gayatri G, Daphne Grassely, Zelos Zhu, Sadchla Mascary. “Package ‘pharmaverseadam’.” October 25, 2024. Available at https://cran.r-project.org/web/packages/pharmaverseadam/pharmaverseadam.pdf.

Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, Davis Vaughan, Posit Software, PBC. “Package ‘dplyr’.” November 16, 2023. Available at https://cran.r-project.org/web/packages/dplyr/dplyr.pdf.

Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, Dewey Dunnington, Teun van den Brand, Posit Software, PBC. “Package ‘ggplot2’.” April 23, 2024. Available at https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf.

Alboukadel Kassambara. “Package ‘ggpubr’.” February 10, 2023. Available at https://cran.r-project.org/web/packages/ggpubr/ggpubr.pdf.

ACKNOWLEDGMENTS

I am grateful to my colleagues and seniors at Ephicacy who have been instrumental in addressing my queries throughout the writing of this paper. My sincere gratitude to Shashikant Kumar, Mrityunjay Kumar and Pradeep Acharya for their constant support and belief in me.

CONTACT INFORMATION

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

Akshata Salian
Ephicacy Lifescience Analytics Pvt Ltd
akshata.salian@ephicacy.com
https://www.linkedin.com/in/akshata-salian-15754222/