PharmaSUG 2025 - Paper DV-251

Integrated R scripts to Power BI

Jun Yang, Avidity Biosciences

Abstract

Visualization plays a critical role in reviewing, estimating data, and identifying outliers in clinical trials. By combining the capabilities of Power BI and R, researchers can create a user-friendly and efficient platform for developing professional visualization packages with data automation and interactive features.

Power BI, developed by Microsoft, is a powerful business analytics tool that enables users to transform raw data into actionable insights through interactive dashboards and reports. On the other hand, R is a widely used programming language in clinical trials, known for its open-source nature, robust statistical capabilities, and extensive community support. Together, these tools offer an effective solution for addressing the data visualization and analytical needs of the pharmaceutical and healthcare industries.

INTRODUCTION

Modern businesses rely on data visualization to make informed decisions. Power BI, a product by Microsoft, has become a leading platform for self-service analytics. Although it includes a variety of built-in visuals, complex data relationships and specific analytical requirements may necessitate more advanced techniques. The integration of R scripting allows users to leverage the extensive R ecosystem for statistical computing and visualization within Power BI.

SETTING UP R IN POWER BI

Although Power BI offers a wide range of features-including data loading, transformation, and various visualization options- it does not natively support R functionality out of the box. To use R scripts within Power BI, users must:

1. Install R on their local machine (e.g., CRAN R or Microsoft R Open).
2. Configure Power BI to recognize the R installation via Options > R scripting.
3. Use the R visual icon in Power BI to insert a custom R script-based visualization.

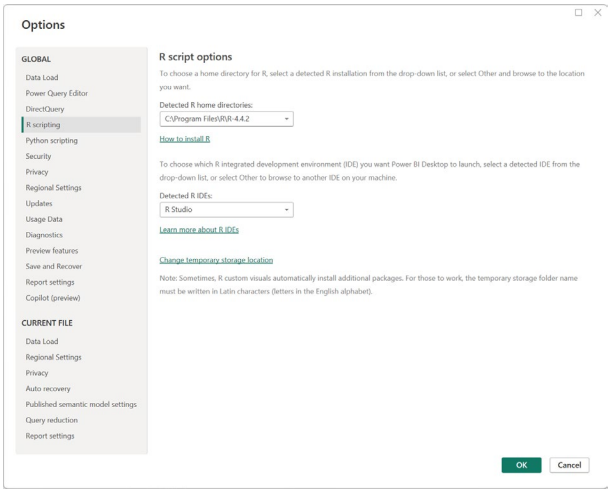
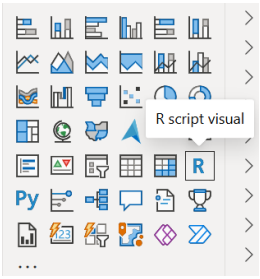


Figure 1: display where to configurate Power BI for R script.

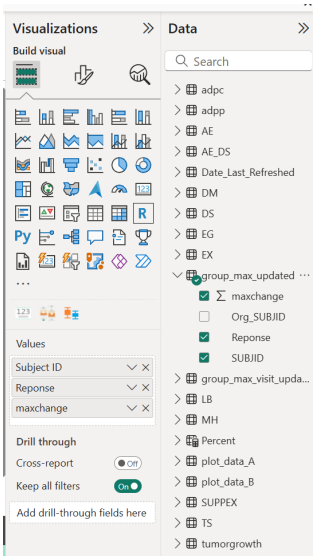
CREATING R PLOTS IN POWER BI

When creating an R visual, you can follow the steps below.

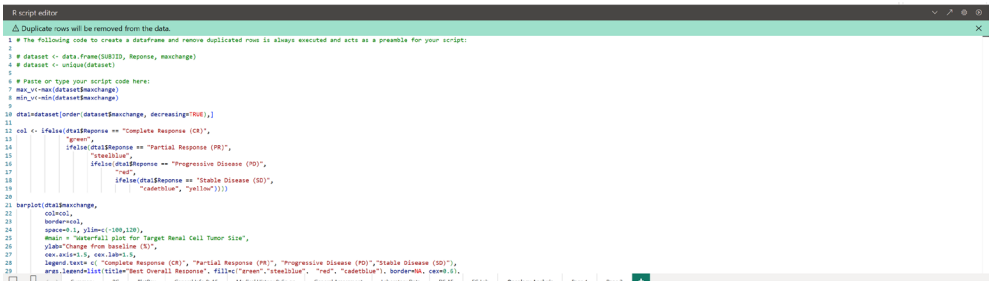
1. Click R script visual icon in the visualizations pane.



2. Drag the desired fields into the Values section to define the dataset available to the R script.



3. Write the R script in the provided editor to generate the desired plot.



WATERFALL PLOT

```
# The following code to create a dataframe and remove duplicated rows is always executed
and acts as a preamble for your script:

# dataset <- data.frame(SUBJID, Reponse, maxchange)
```

```
# dataset <- unique(dataset)

# Paste or type your script code here:

dta1=dataset[order(dataset$maxchange, decreasing=TRUE),]

col <- ifelse(dta1$Reponse == "Complete Response (CR)",
              "green",
              ifelse(dta1$Reponse == "Partial Response (PR)",
                     "steelblue",
                     ifelse(dta1$Reponse == "Progressive Disease (PD)",
                            "red",
                            ifelse(dta1$Reponse == "Stable Disease (SD)",
                                   "cadetblue", "yellow"))))

barplot(dta1$maxchange,
        col=col,
        border=col,
        space=0.1, ylim=c(-100,120),
        #main = "Waterfall plot for Target Renal Cell Tumor Size",
        ylab="Change from baseline (%)",
        cex.axis=1.5, cex.lab=1.5,
        legend.text= c( "Complete Response (CR)", "Partial Response (PR)", "Progressive
Disease (PD)","Stable Disease (SD)"),
        args.legend=list(title="Best Overall Response",
fill=c("green","steelblue",  "red", "cadetblue"), border=NA, cex=0.6),
        )

abline(h=20, col = "black", lwd=0.5) # The "PD" line
abline(h=-30, col = "black", lwd=0.5) # This "PR" line
```

Here is an example of an R script used to create a waterfall plot to visualize the best overall response of individual patients to a particular drug. This type of plot is commonly used to present patient-level changes in a clinical parameter, such as tumor burden.

In a typical waterfall plot:

- The horizontal (x) axis represents individual patients, often aligned along a baseline.
- The vertical (y) axis displays the maximum percent change from baseline, such as the percent reduction or growth of a tumor, typically measured via radiologic assessments.
- Bars extending above the baseline indicate non-responders or progressive disease (PD).
- Bars extending below the baseline represent patients who achieved tumor reduction, with the negative percentage indicating the extent of response.

The dataset used for the plot includes three variables:

- Subject ID
- Response category (e.g., CR, PR, SD, PD)
- MaxChange (maximum percent change from baseline)

To create the plot:

- Sort the dataset in descending order of MaxChange, as waterfall plots typically display responses from PD to CR.
- Assign custom colors to different response categories for clarity.
- Use the barplot() function in R to generate the waterfall chart.

- Add reference lines to indicate clinically meaningful thresholds, such as the cutoffs for PD and PR.

SPIDER PLOT

```
library(ggplot2)

col <- ifelse(dataset$Reponse == "Complete Response (CR)",
              "green",
              ifelse(dataset$Reponse == "Partial Response (PR)",
                     "steelblue",
                     ifelse(dataset$Reponse == "Progressive Disease (PD)",
                            "red",
                            ifelse(dataset$Reponse == "Stable Disease (SD)",
                                   "cadetblue", "yellow"))))

response_cat<- c("Complete Response (CR)","Partial Response (PR)","Progressive Disease
(PD)","Stable Disease (SD)","Others")

p <- ggplot(dataset, aes(x=Month, y=PCHG, group=SUBJID)) +
  theme_bw(base_size=14) +
  theme(axis.title.x = element_text(face="bold"), axis.text.x =
element_text(face="bold")) +
  theme(axis.title.y = element_text(face="bold"), axis.text.y =
element_text(face="bold")) +
  theme(plot.title = element_text(size=18, hjust=0.5))+
  theme(legend.position = "right")+
  labs(#title = "Spider Plot for Target Tumor Size",
       x = "Time (month)", y = "Change from baseline (%)")
## Now plot
p <- p + geom_line(aes(color=col )) +
  geom_point(aes(shape="Reponse", color=col), show.legend=FALSE) +
  scale_colour_manual(name="Best Overall Reponse",breaks=col,values =
col,labels=dataset$Reponse)+
  coord_cartesian(xlim=c(0, 20))
p
```

The spider plot presents the individual changes in tumor measurements over time relative to baseline tumor burden. Baseline tumor size is a well-known confounding factor of drug effect which must be accounted for when analyzing data in early clinical trials.

To create the plot:

- Assign custom colors to different response categories for clarity.
- Use the ggplot2 library in R to generate the spider chart.
- Customized titles, line thickness, and other visual elements.

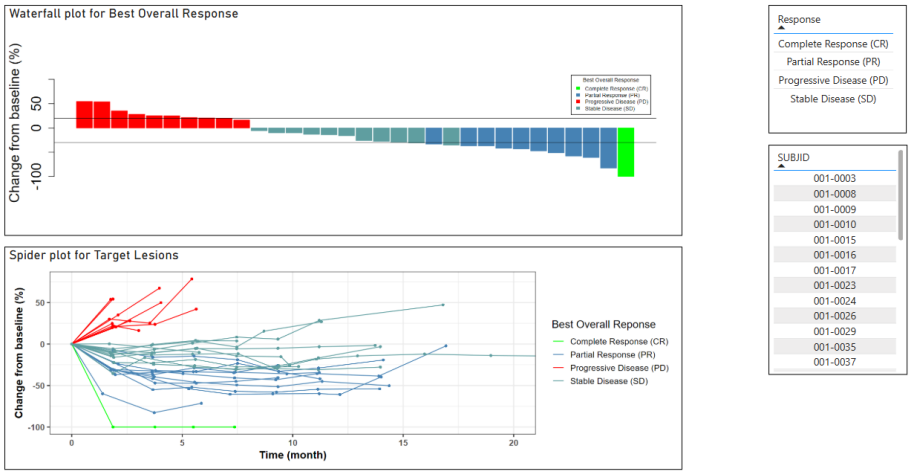


Figure 2: the above is the waterfall plot and the below is the spider plot and the right two boxes are filter.

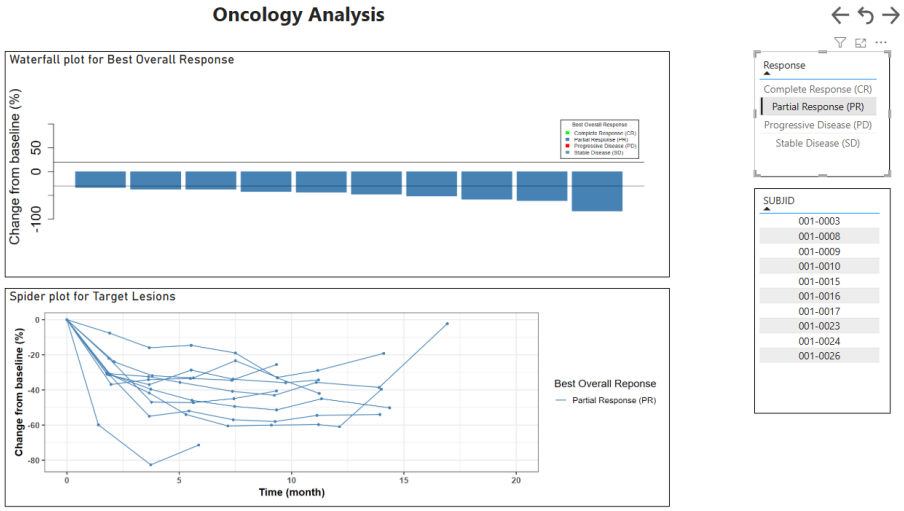


Figure 3: display PR (partial response) data on both waterfall and spider plots.

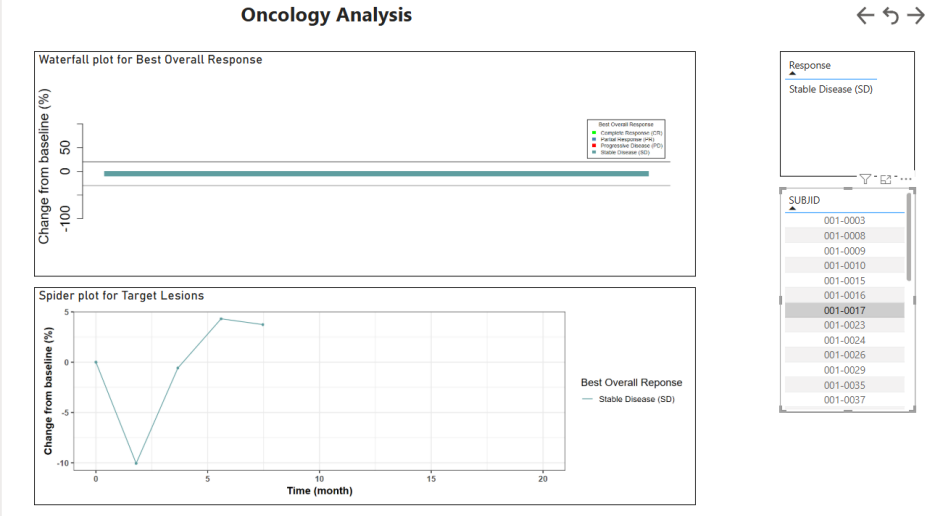


Figure 4: display only one subject.

This approach offers users an innovative way to review data. Not only can they view specific plots generated using R scripts, but they can also leverage Power BI's interactive features. This enables users to create customized views by simply clicking buttons, eliminating the need to re-write multiple R scripts with different parameters.

LIMITATIONS AND CONSIDERATIONS

1. Performance: R visuals are rendered on the client side and may lag with large datasets
2. Dependencies: All required R packages must be installed on the local machine

BEST PRACTICES

1. Pre-process data within Power BI before passing it to R
2. Minimize dataset size to improve performance
3. Use well-supported R packages like ggplot2, plotly, or lattice
4. Document R scripts clearly for maintainability

CONCLUSION

R integration significantly enhances Power BI’s visualization capabilities, especially for users with a background in data science or statistics. It empowers analysts to go beyond the limitations of built-in visuals and create bespoke visualizations tailored to specific analytical needs. While there are limitations, thoughtful implementation of R visuals can lead to richer, more insightful dashboards.

REFERENCES

[1] Use R scripts in Power BI Desktop. https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-r-scripts

[2] ggplot2: Elegant Graphics for Data Analysis. Springer. https://ggplot2-book.org/

[3] The R Project for Statistical Computing. https://www.r-project.org/

CONTACT INFORMATION

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

Jun Yang
Avidity Biosciences
jun.yang@aviditybio.com