

Visualize Tumor Response Data using ggplot2 R Package through Examples

Christine Teng

Merck & Co., Inc

Outline

- Challenges for SAS programmers learning R
- Usage of tidyverse/ggplot2 R packages
- Oncology data for plot examples
- Five plot examples using ggplot2
- Summary

Challenges for SAS Programmers

- R is free so it does not have customer service support. Debugging in R could be challenging. You need to rely on online community for technical support
- There are many R packages which are contributed by different people, and there is no quality control
- R is a functional programming language and has additional data structure (ex: vector and list) comparing to SAS
- Same function name can be created in different packages by different developers, need to prefix package name to use the correct function

Motivation

Tidyverse is a set of R packages that work in harmony because they share common data representations and APIs. These packages work efficiently together to support data modeling, wrangling, and visualization tasks. Tidyverse was created by **Hadley Wickham and his team**. These packages are enhanced and maintained more often because of the popularity.

tibble	for tibbles, a modern re-imagining of data frames
ggplot2	for data visualization
dplyr	for data manipulation
tidyr	for data tidying
readr	for data import.
purrr	for functional programming
stringr	for strings
forcats	for factors
and more	



The tidyverse, shiny, ggplot, ggvis, dplyr, knitr, R Markdown, and packrat are R packages from Rstudio.

ggplot2

ggplot2 is the data visualization package based on "The Grammar of Graphics". You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

ggplot2

- ggplot2 is Intuitive for R beginner to start learning R graphical programming.
- For this presentation, we will use pre-populated SAS data as input. We can see with effortless ggplot2 code, we can draw very sophisticated plots.
- You can test the code adding one line at a time and adjust as you go

ggplot2 Grammar

A complete Sentence: data + aes + geom

- **Aesthetic mapping** (variables represented by aesthetics, ex: axes, color, size, position, shape etc)
- **Geometric objects** (line, point, text, bar etc)
- Scales (aesthetic scales)
- Coordinates (x,y position aesthetics only)
- Labels (title, subtitle, caption, x axis, y axis etc)
- Themes (cosmetic look of the plot)
- Facetting (split plot by category)

R Packages

Packages/Libraries used in the examples:

`library(haven)` ← for reading SAS dataset

`library(ggplot2)` ← for drawing plots

`library(directlabels)` ← extension for ggplot2

`library(plotly)` ← extension for ggplot2

Read SAS dataset into data frame:

ex: `trvd <- read_sas(paste0(folder, "trvdir.sas7bdat"))`

About the input Data

Oncology Response Data for the Examples

At each time point:

- **SUBJECT** – Patients
- **WEEK** – Assessment Time Points
- **BASELINE** – Baseline Sum of Diameter
- **SOD** – Sum of Diameter (Sum of Longest Diameters and Sum of diameter of non-lymph-node tumor)
- **PCBSD** – Percent Change from Baseline in Sum of Diameter
- **NADIR** – smallest Sum of Diameter prior to current time point
- **PCNSD** – Percent Change from Nadir in SOD
- **NLR** – New Lesion (Y, N)
- **OVRLRESP** – Overall Response CR, PR, PD, SD, and NE

Oncology Response Data for the Examples

Subject Level:

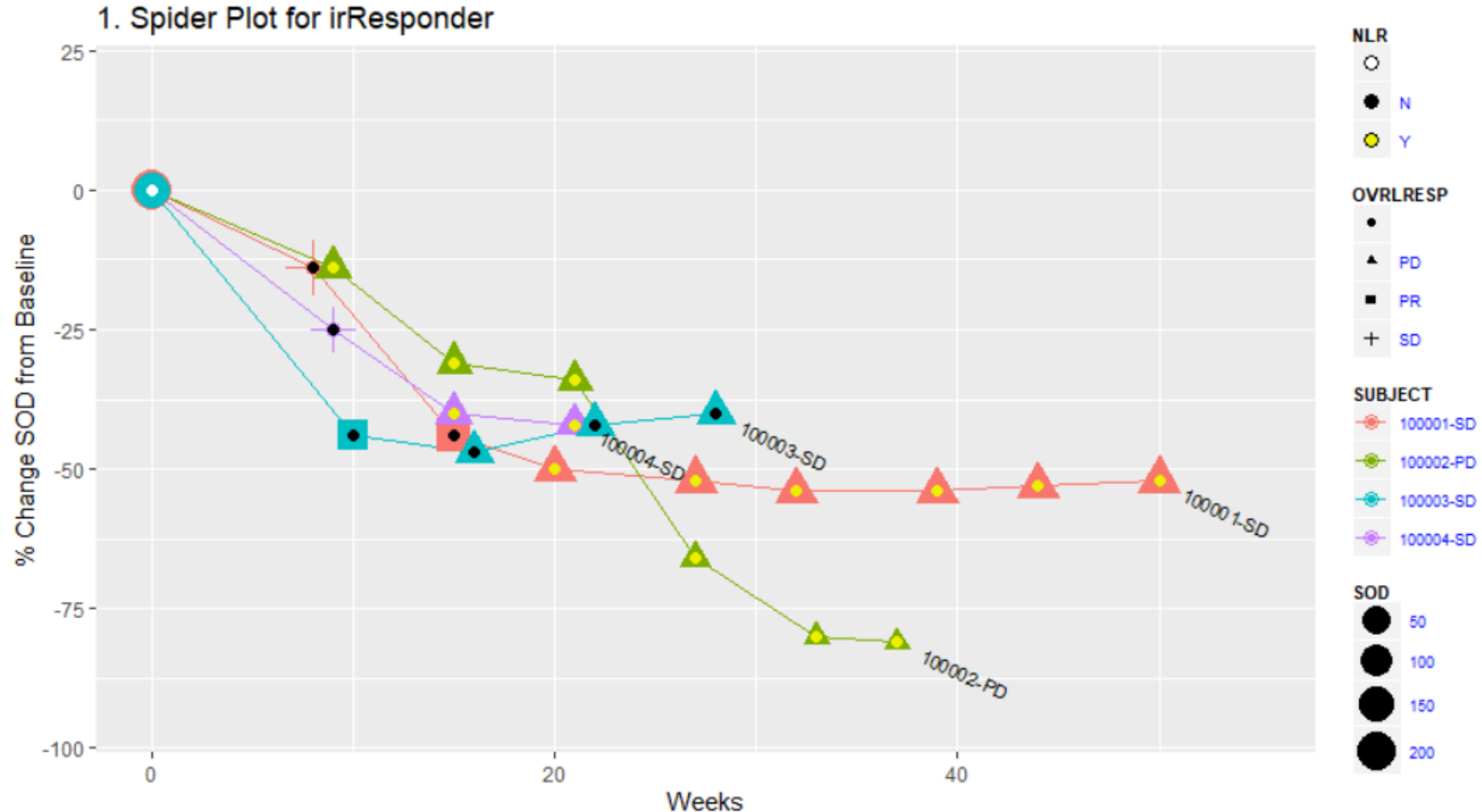
- **RSPDUR** – response duration
- **DTH** – Death Relative day from reference start date
- **FPD** – First progression day from reference start date
- **FCRPR** - First response day from reference start date
- **EVNTDESC** - Censoring/Event description
- **LASTD** – Last dose timepoint from reference start date
- **BOR** – Best Confirmed Response over time
- **BESTPCHG** – Maximum %CHG in tumor size

TL Response Criteria (RECIST)

Response	Definition
Complete Response (CR)	All non-nodal TLs disappeared; all lymph nodes short axis <10 mm
Partial Response (PR)	SOD decreased $\geq 30\%$ from baseline
Progressive Disease (PD)	SOD increased $\geq 20\%$ from nadir and the 20% has absolute increase ≥ 5 mm
Stable Disease (SD)	Not PR nor PD
Not Evaluable (NE)	Cannot determine target lesion response

Example 1

Example 1- Overlaying Two geom_point() with Customization Functions



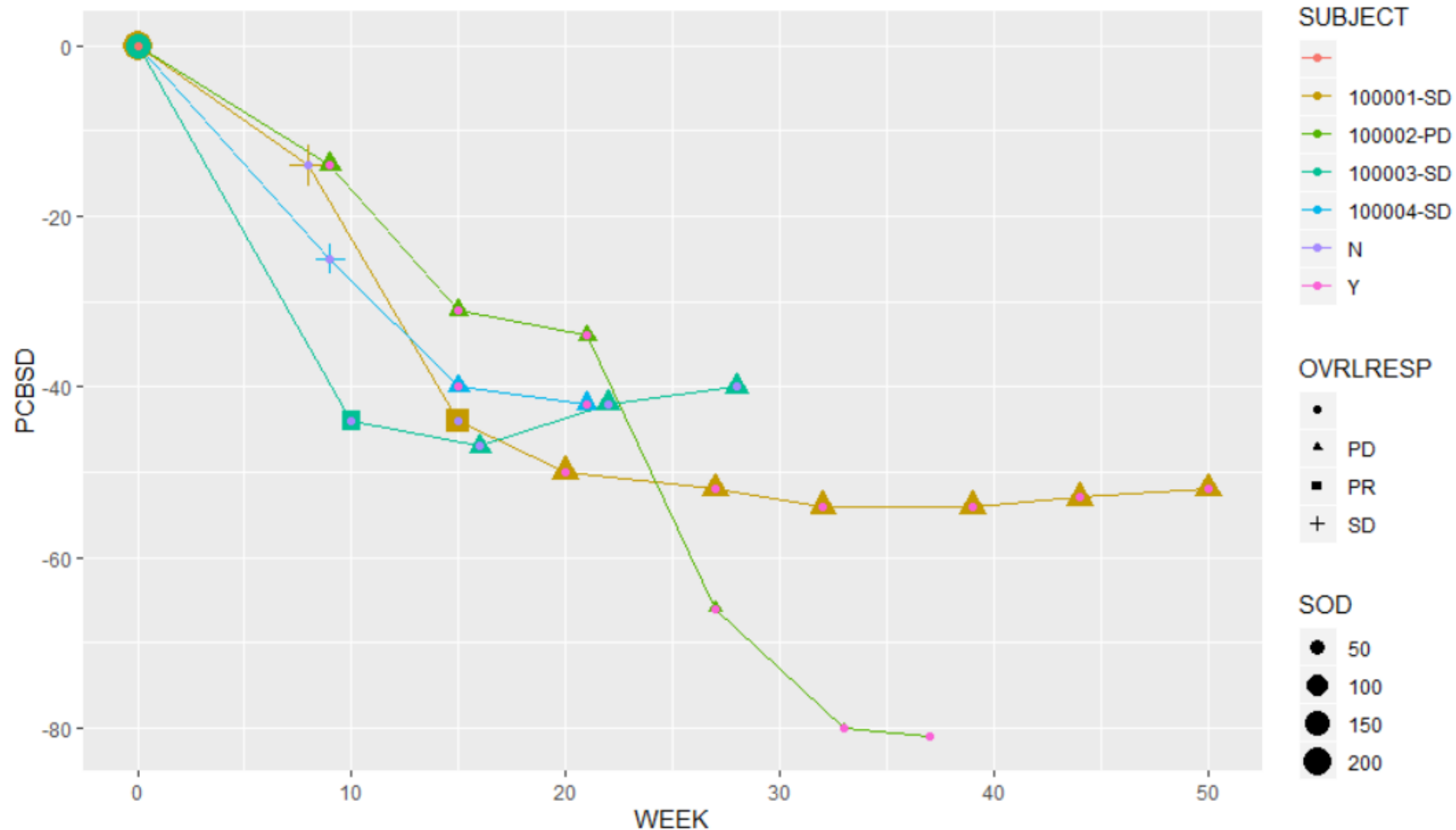
Example 1- Overlaying Two geom_point()

```
ggplot(trvd, aes(x = WEEK, y = PCBSD, color = SUBJECT)) +  
  geom_line(aes(color = SUBJECT)) +  
  geom_point(aes(shape=OVRLRESP, size=SOD)) +  
  geom_point(aes(fill=NLR), size=3, shape=21) +  
  geom_dl(aes(x = WEEK, y = PCBSD, label = SUBJECT), color="black", method = list("last.points",  
    cex = 0.7, rot = -25, hjust = -.3)) +  
  coord_cartesian(xlim=c(0,55),ylim=c(-95,20)) +  
  scale_size_continuous(range = c(4, 8)) +  
  scale_fill_manual(values=c("white","black","yellow2")) +  
  labs(y="% Change SOD from Baseline", x="Weeks", title="1. Spider Plot for irResponder") +  
  theme(legend.title = element_text(color = "black", size = 8, face="bold"), legend.text = element_text(  
    color = "blue", size = 7),legend.position = 'right',legend.spacing.y = unit(0.01, 'cm'))
```

Example 1- Overlaying two geom_point() w/o Customization Functions

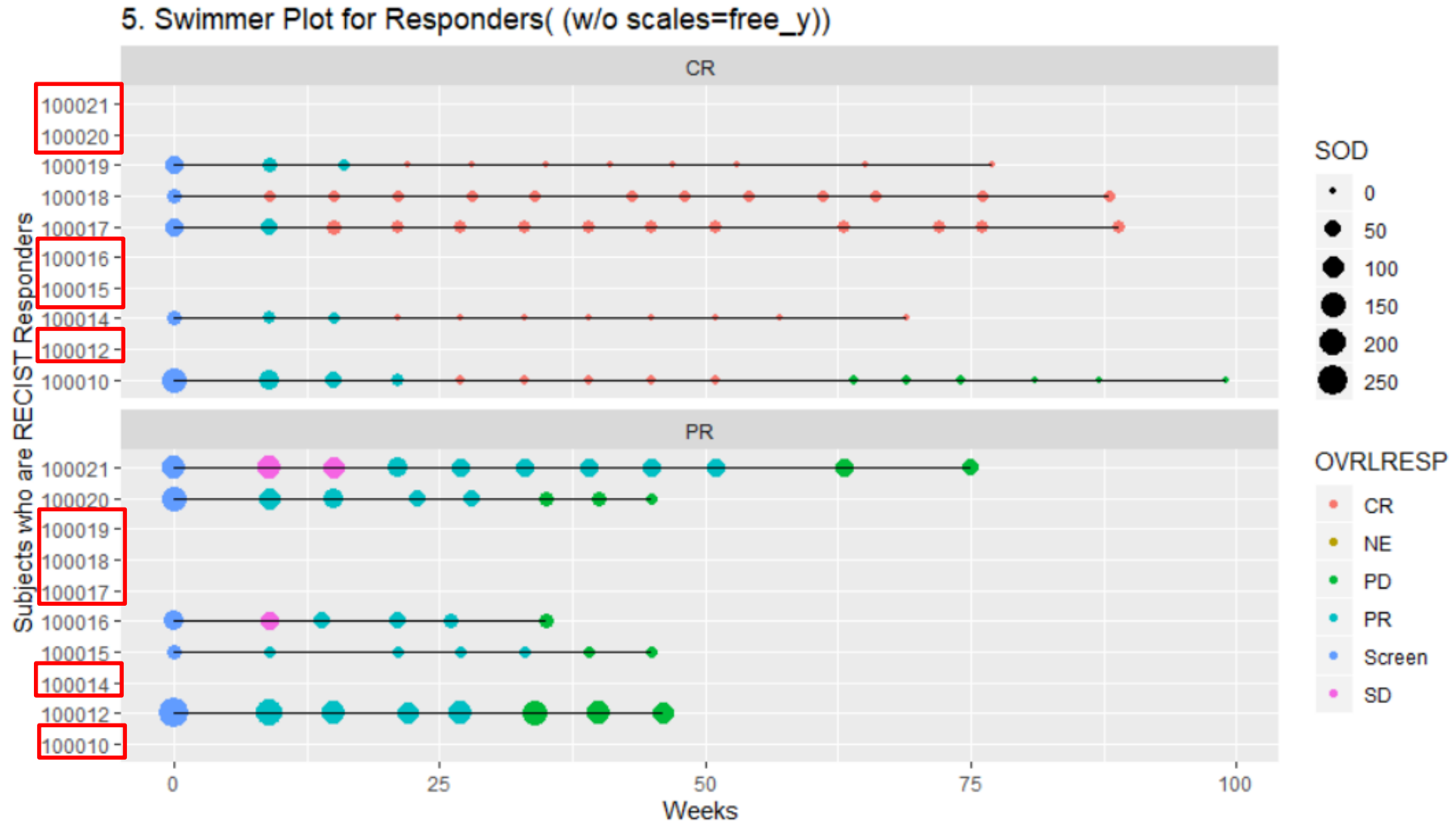
```
ggplot(trvd, aes(x = WEEK, y = PCBSD, color=SUBJECT)) +  
geom_line(aes(color = SUBJECT)) +  
geom_point(aes(shape=OVRLRESP, size=SOD)) +  
geom_point(aes(color=NLR))
```

Example 1- Overlaying Two geom_point() w/o Customization Functions



Example 2

Example 2 – Split into Multiple Subplots (1)

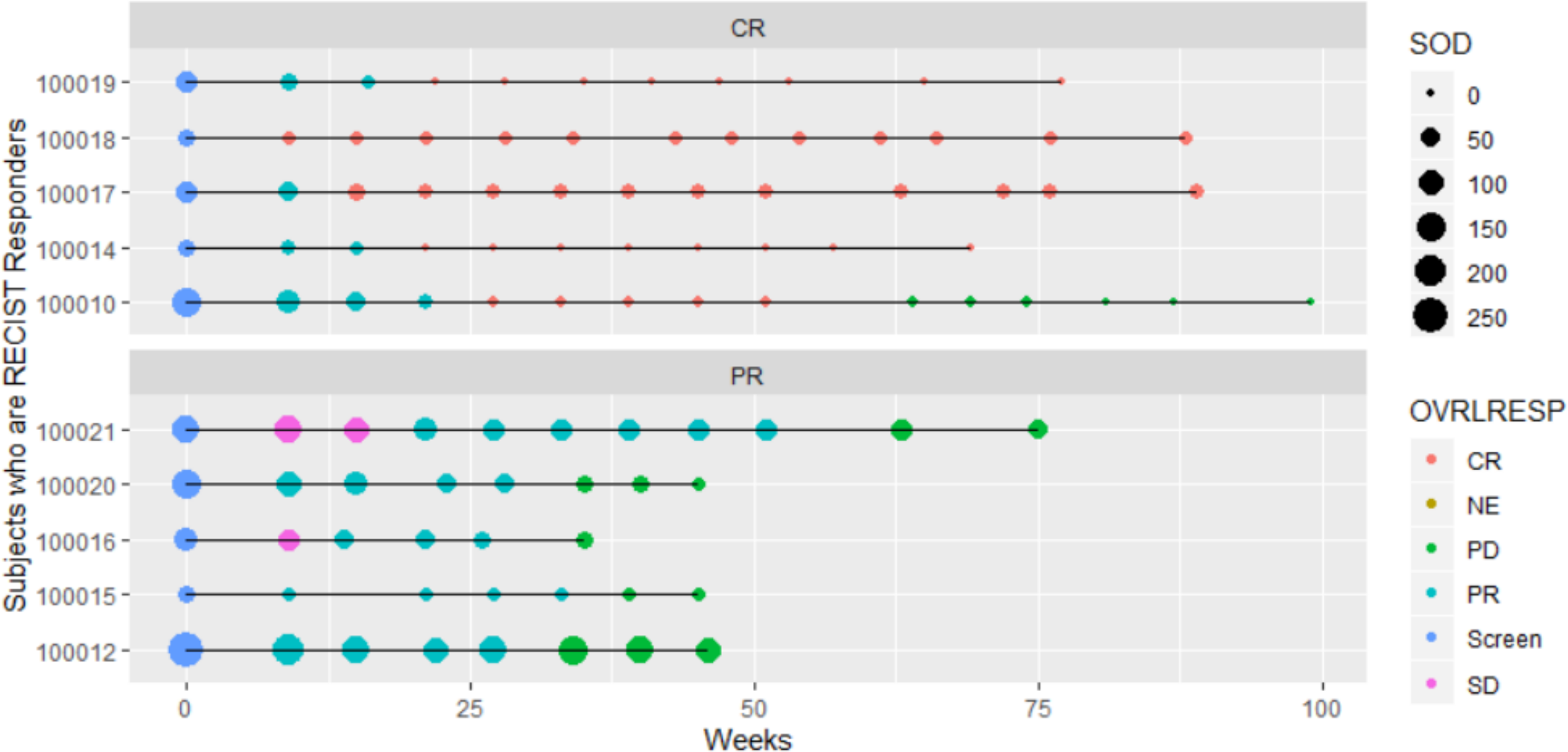


Example 2 – Split into Multiple Subplots (2)

```
ggplot(trvdcr, aes(x = WEEK, y = SUBJECT)) +  
geom_point(aes(size=SOD, color = OVRLRESP ), na.rm = TRUE) +  
geom_line() +  
labs(y="Subjects who are RECIST Responders", x="Weeks", title="5. Swimmer Plot  
for Responders(Split vertically)") +  
facet_wrap(~BOR, ncol=1, scales = 'free_y') ← split, and free-y remove the dups
```

Example 2 – Split into Multiple Subplots (2)

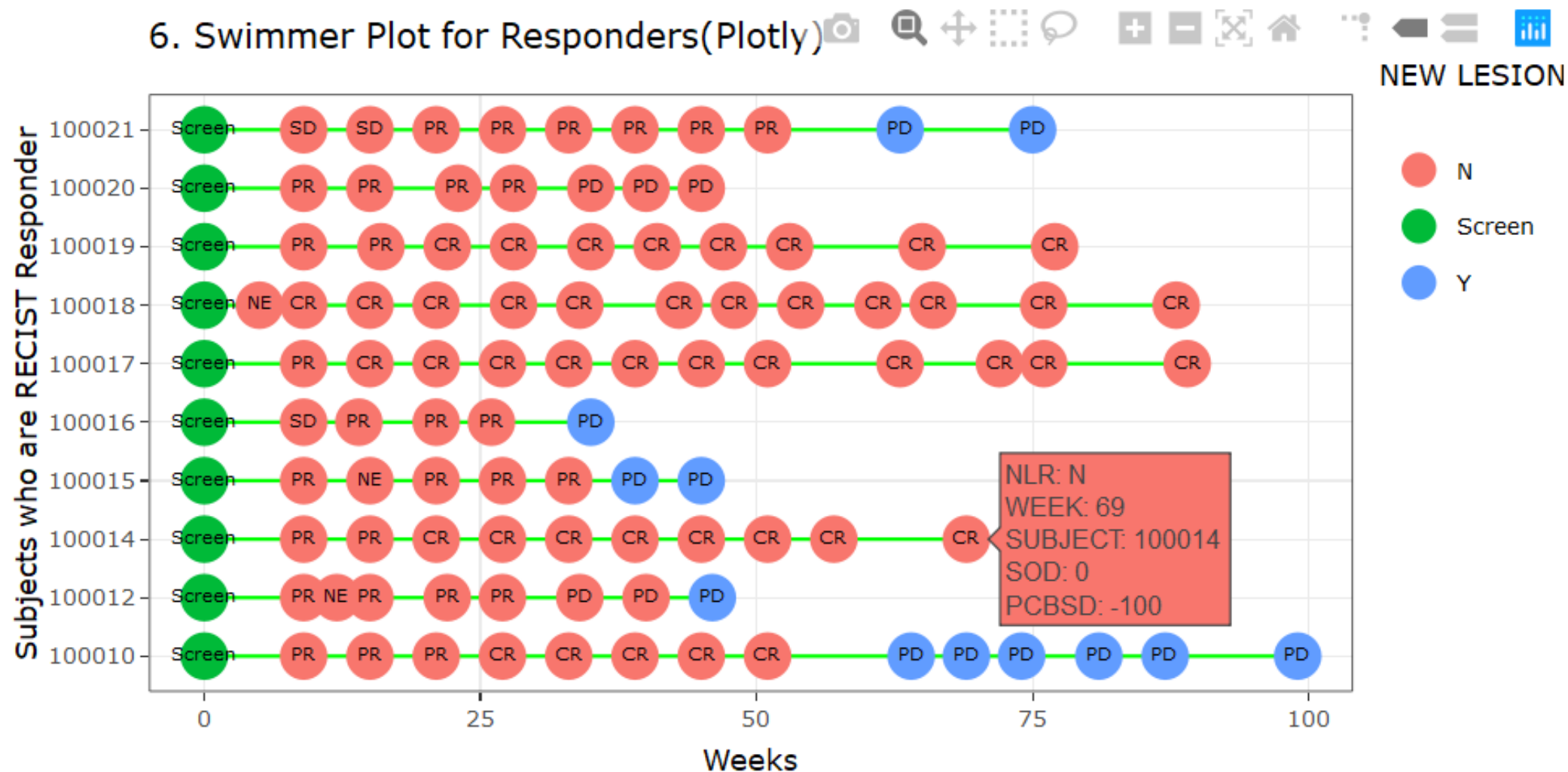
5. Swimmer Plot for Responders(Split vertically)



Example 3

Example 3 – Interactive Plotly Extension

6. Swimmer Plot for Responders(Plotly)



Example 3 – Interactive Plotly Extension

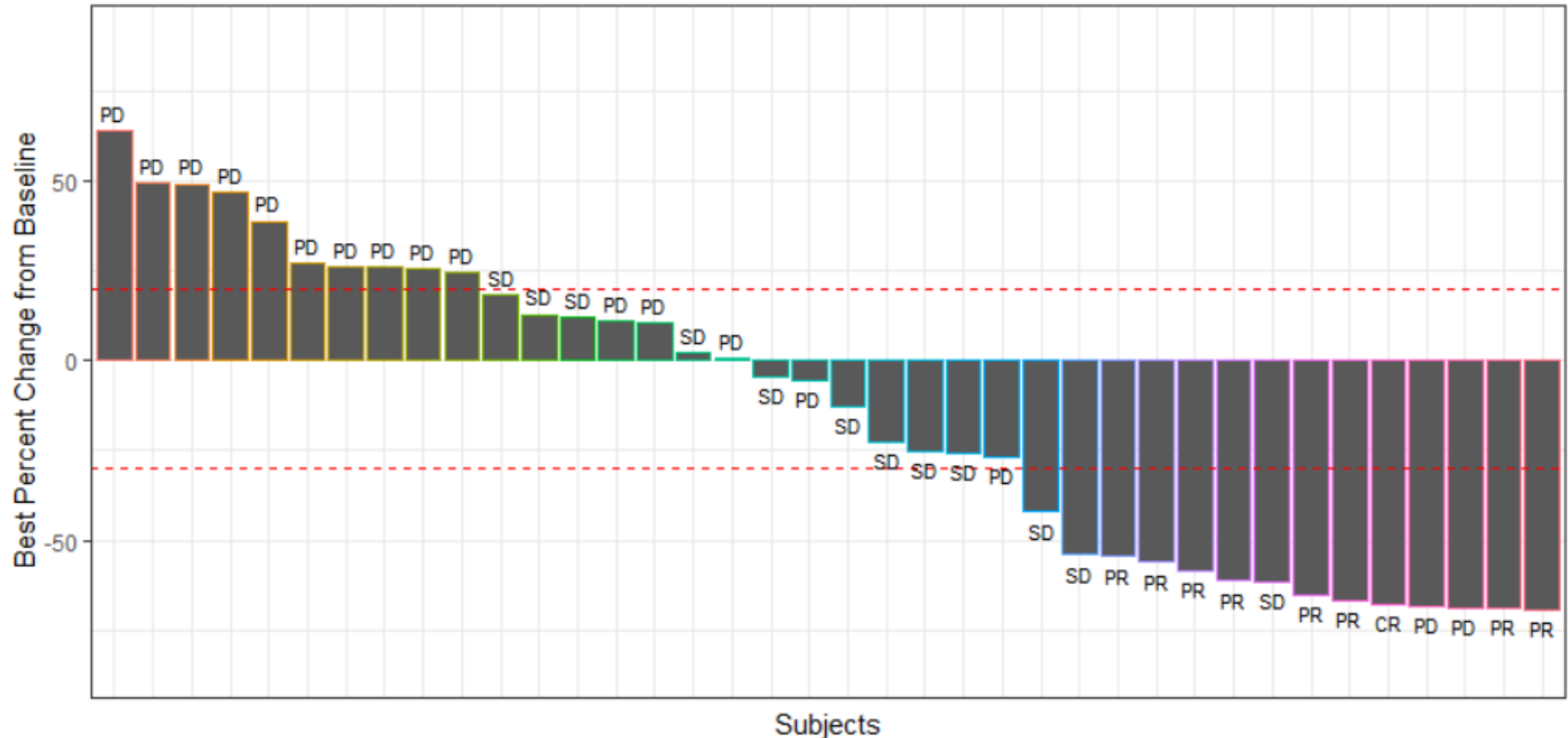
```
p<- ggplot(trvdcr, aes(x = WEEK, y = SUBJECT, name=SOD, name1=PCBSD)) +  
  geom_line(color = "green") +  
  geom_point(aes(color =NLR ), size=6, na.rm = TRUE) +  
  geom_text(aes(label=OVRLRESP), size=2.5) +  
  labs(y="Subjects who are RECIST Responder", x="Weeks",  
        title="6. Swimmer Plot for Responders(Plotly)", color = "NEW LESION")+  
  theme_bw()
```

`ggplotly(p)` ← need to save plot into an object before using it

Example 4

Example 4 – No Legends and X-axis Ticks

7. Waterfall Plot for Best Percent Change SOD from Baseline in Tumor Size(with BOR)



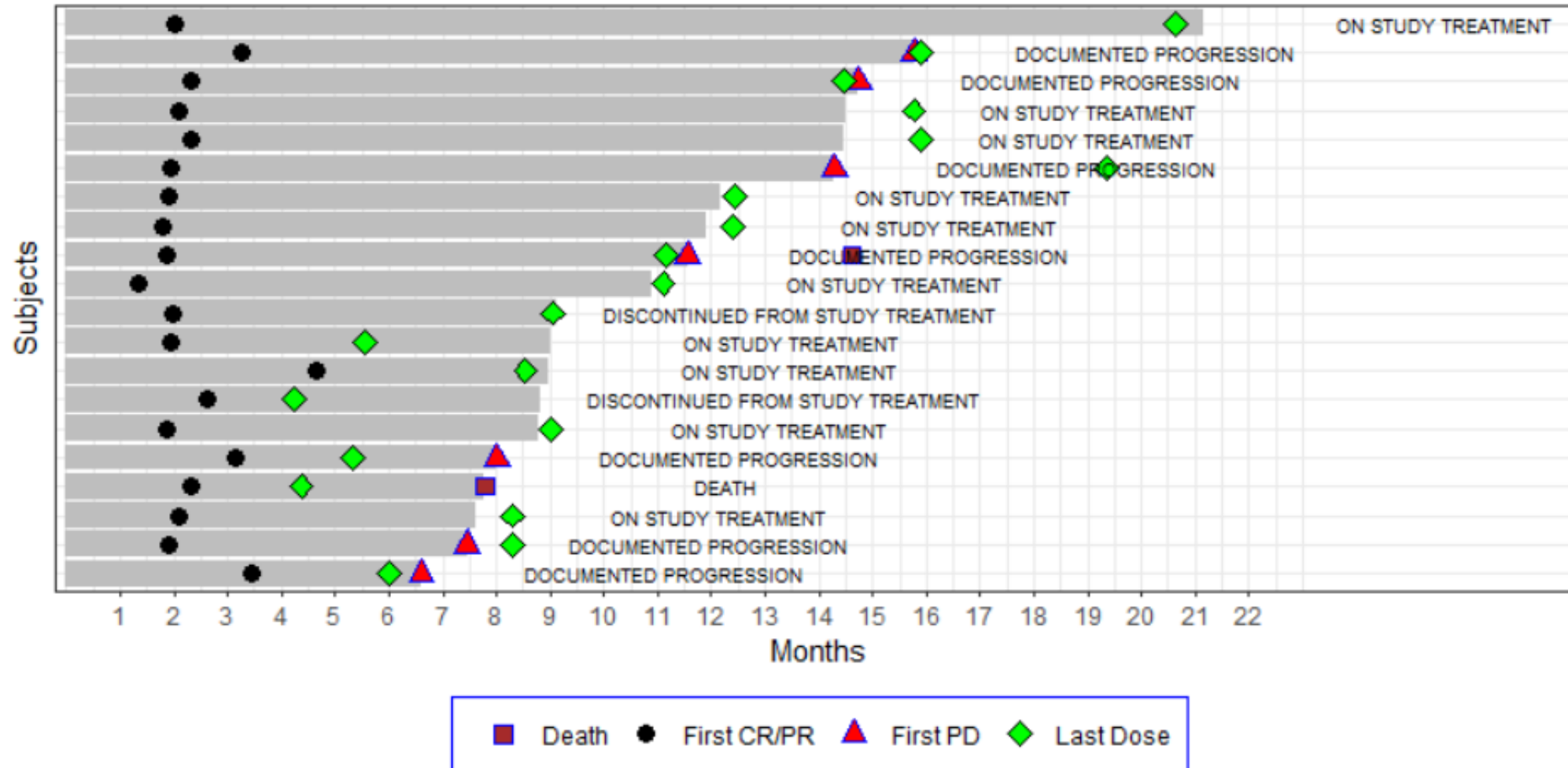
Example 4 – No Legends and X-axis Tics

```
ggplot(subjw, aes(x = SUBJECT, y = BESTPCHG, col = SUBJECT)) +  
  coord_cartesian(ylim = c(-85,90)) +  
  geom_bar(stat = "identity", width = 0.9) +  
  geom_abline(slope=0, intercept=20, col = "red", lty=2) +  
  geom_abline(slope=0, intercept=-30, col = "red", lty=2) +  
  geom_text(aes(x = SUBJECT, y = BESTPCHX, label=BOR), color="black", size=3) +  
  theme_bw() +  
  labs(title = "7. Waterfall Plot for Best Percent Change SOD from Baseline in Tumor  
Size(with BOR)", x = "Subjects", y = "Best Percent Change from Baseline") +  
  theme(  
    axis.text.x = element_blank(),  
    axis.ticks.x = element_blank(),  
    legend.position="none") ← remove legends and x-tics
```

Example 5

Example 5 – Display Descending with Event Description

10. Swimmer Plot for Responders(Use 'nudge_y' and 'reorder')



Example 5 – Display Descending with Event Description

```
ggplot(subjs, aes(x = reorder(SUBJECT, RSPDUR), y = RSPDUR) ) +  
  geom_bar(stat = "identity", fill="grey") +  
  geom_point(aes(y = FPD, col="First PD"), fill="red", shape=24, size=3, na.rm = TRUE) +  
  geom_point(aes(y = FPRCR, col="First CR/PR"), fill="black", shape = 21, size = 3) +  
  geom_point(aes(y = DTH, col="Death"), shape = 22, fill = "brown", size=3, na.rm = TRUE) +  
  geom_point(aes(y = LASTD, col="Last Dose"), shape=23, size=3, fill="green") +  
  coord_flip() +  
  geom_text(aes(x = SUBJECT, y = RSPDUR, label=EVNTDESC), nudge_y = 4.5, color="black",  
    size=2.5) + ← add space between the end of bar to the text description
```

Example 5 – Display Descending with Event Description

```
theme_bw() +  
labs(x = "Subjects", title = "10. Swimmer Plot for Responders(Use 'nudge_y' and 'reorder')") +  
scale_y_continuous("Months", limits = c(0,28), breaks=seq(1,22,1), expand = c(0,0.2)) +  
scale_color_manual(NULL, values = c("First PD" = "blue", "First CR/PR" = "black",  
                                   "Death" = "blue", "Last Dose" = "black"),  
                  guide = "legend") +  
guides(col = guide_legend(override.aes = list(shape = c(22,21,24,23),  
                                                fill = c("brown","black","red","green")))) +  
theme(plot.title = element_text(hjust = 0.5),  
      legend.position = "bottom",  
      legend.background = element_rect(linetype = "solid", color = "blue"),  
      axis.text.y = element_blank(), axis.ticks.y = element_blank())
```

Summary

Summary

- R is free and anyone can download to practice and create packages to share. However, maintenance of the packages maybe an issue.
- Tidyverse has many packages (own ecosystem) for data scientists to perform data manipulations relative simpler than older packages and it has gain popularity which enable enhancements or bug findings quicker by users. For SAS enhancements, we have to wait for releases.
- ggplot2 layer programming concept is easy to follow and it provides high quality graphs
- Preparing the data structure and selecting the right plots are important steps to convey the data story

In the pharmaceutical industry, SAS is the dominant statistical software. Most of the company-developed clinical programs were already available and validated. R comes into play when SAS can not readily provide the solution or when there is a time constraint to respond to agency requests with fancy graphs. For such cases, we can still use SAS datasets to support the R graphs.

Resource

Garrett Grolemund, Hadley Wickham. “R for Data Science”. O’Reilly, January 2017.

<https://r4ds.had.co.nz/index.html>

Rstudio Webinars

<http://www.rstudio.com/resources/webinars/>

ggplot2 Extensions/Gallery

<https://www.ggplot2-exts.org/gallery/>



Contact

Christine Teng
Merck & Co., Inc.
christine_teng@merck.com