



R and it's application in Clinical Trial data analysis

Presentation # DV11

Introduction

Name: Davuluri, Santosh Kumar

Title: Sr. Statistical Programmer

Company: Vita Data Sciences

Agenda

- Usage of R in clinical trial data analysis.
- Shiny and Shinydashboard in R.
- ggplot2 in R.
- Waterfall and Spider Plot
- Kaplan Meier plot using survival and survminer libraries.
- Conclusion

Usage of R in clinical trial data analysis

4 REASONS WHY

R can be a potential powerful tool for Data Analysis

1

OPEN SOURCE

Freely Available Language
and Environment
Community Support
Supports Extensions

2

MULTI PLATFORM

Compiles on platforms like
UNIX, Windows, MacOS
and Linux. R has many
Packages and ML capability.

3

STATISTICAL TECHNIQUES

R provides variety of statistical
(Linear and Nonlinear, Time-
series Analysis, Classification,
Clustering etc.) and Graphical
Techniques.

4

QUALITY PLOTS

Publication-quality Plots can
be produced with ease
including mathematical
symbols and formulae.

Shiny in R

- Interactive web applications
- A shiny app has 3 components to it:
 - A user interface object
 - A server function
 - A call to the app
- For every input function in UI component, we have a corresponding function in the server component.

Shiny app template

```
library(shiny)
ui <- fluidPage()
server <- function(input, output){}
shinyApp(ui = ui, server = server)
```

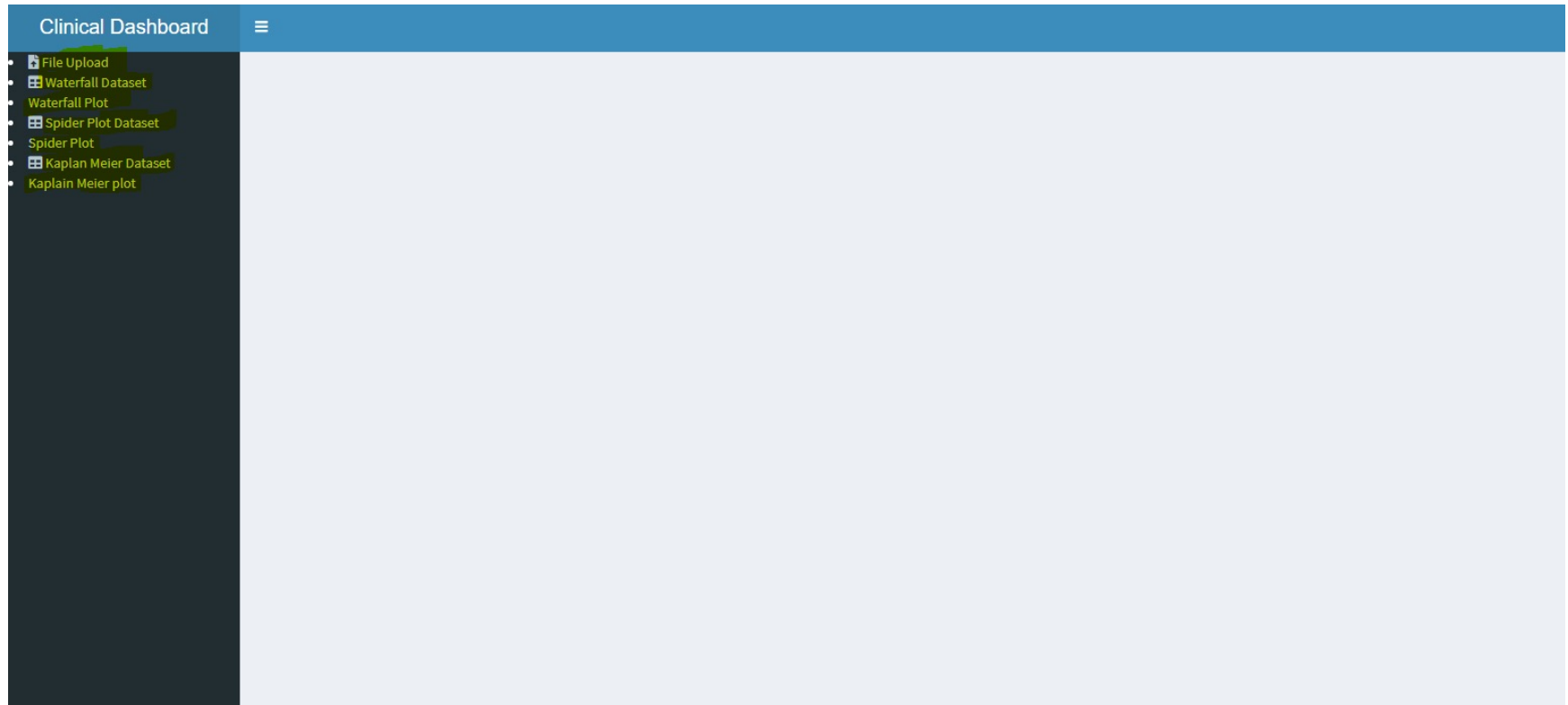
Shiny dashboard library

- Library shinydashboard is used for creating a dashboard.
- These dashboards can be used for reviewing clinical trial data.
- The library offers different layouts which can suit the need of the developers.

Code template for dashboard

```
## app.R ##  
library(shiny)  
library(shinydashboard)  
  
ui <- dashboardPage(  
  dashboardHeader(),  
  dashboardSidebar(),  
  dashboardBody()  
)  
  
server <- function(input, output) { }  
  
shinyApp(ui, server)
```

Clinical Dashboard



Plotting with ggplot2

- **ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) + <GEOM_FUNCTION>()**
- aesthetic mapping is done using **aes ()** function by selecting the variables and how to represent them on graphs.
- The **'+'** helps to add layers on plots.
- ggplot2 offers different geoms:
 - **geom_point()** for scatter plots, dot plots, etc.
 - **geom_boxplot()** for box plots.
 - **geom_line()** for trend lines, time series, etc.

File upload

Clinical Dashboard 

-  File Upload
-  Waterfall Dataset
- Waterfall Plot
-  Spider Plot Dataset
- Spider Plot
-  Kaplan Meier Dataset
- Kaplan Meier plot

Pick Dataset for Waterfall Plot

Browse...

No file selected

Pick Dataset for Spider Plot

Browse...

No file selected

Pick Dataset for Kaplan Meier Plot

Browse...

No file selected

Visualizing dataset

Clinical Dashboard

Waterfall dataset

Show 10 entries

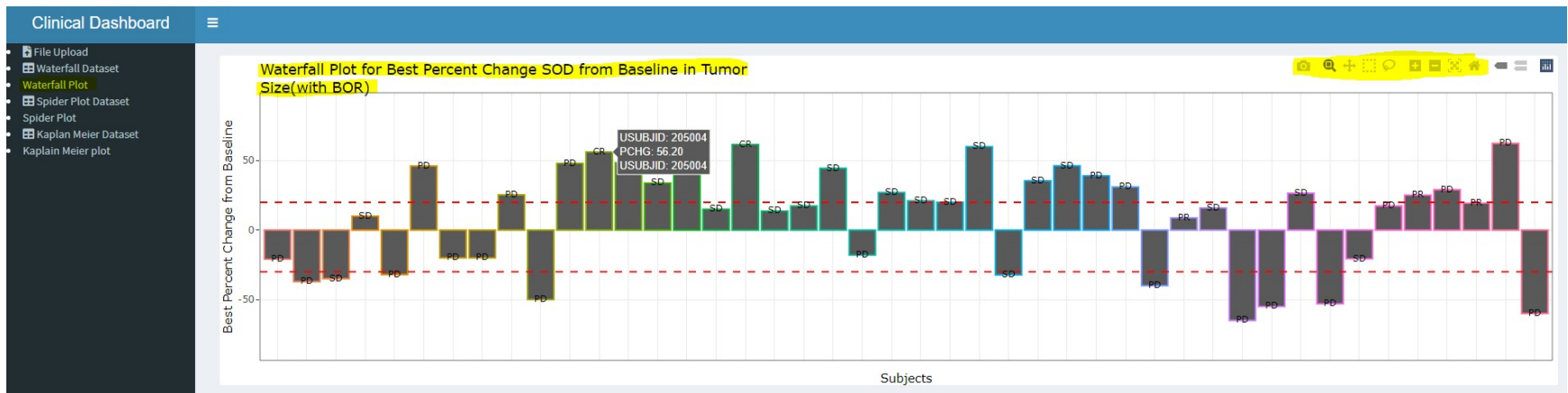
Search:

	USUBJID	PCHG	BEST_RESPONSE	N
1	213004	60	SD	1
2	203014	25.3	PD	2
3	222011	17.3	PD	3
4	207005	13.68	SD	4
5	218010	8.82	PR	5
6	207001	67.89	SD	6
7	223001	62.3	PD	7
8	207004	61.48	CR	8
9	205004	56.2	CR	9
10	206008	48.42	PD	10

Showing 1 to 10 of 44 entries

Previous 1 2 3 4 5 Next

Waterfall plot using ggplot2 and plotly



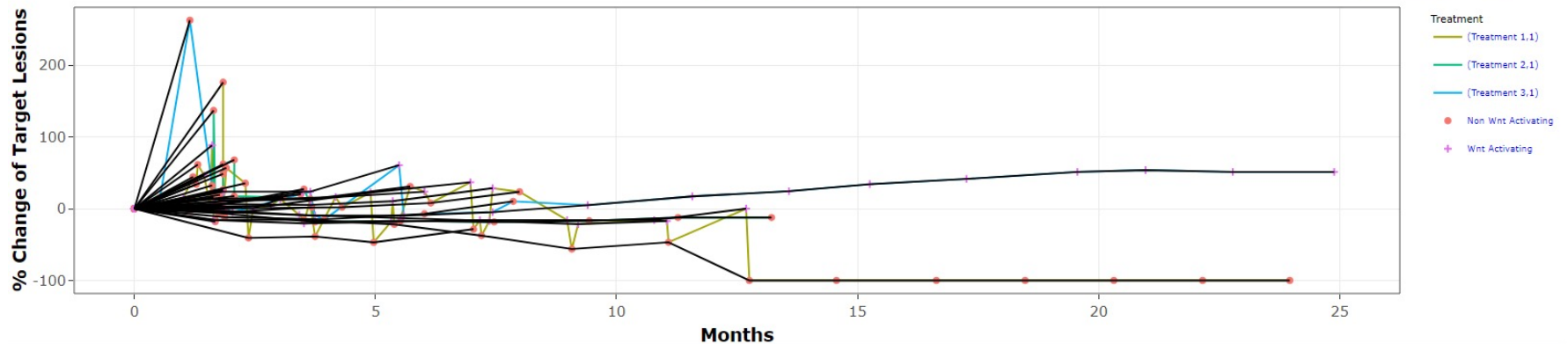
Spider plot

```
output$spplot <- renderPlotly({
  data2 <- data_sp()
  p <- ggplot(data2, aes(x=DUR, y=PCHG)) +
    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)) +
    labs(list(title = "Spider Plot", x = "Time (in months)", y = "Change from baseline (%)")) +
    geom_line(aes(color=ATRTGP)) +
    geom_point(aes(shape=PATT, color=PATT), show.legend=FALSE) +
    geom_path(aes(group=USUBJID)) +
    scale_colour_discrete(name="Treatment", labels=c("Treatment 1", "Treatment 2", "Treatment 3")) +
    scale_shape_manual(name = "PATT", values = c("wnt Activating"=3, "Non wnt Activating"=16)) +
    coord_cartesian(xlim=c(0,25)) +
    labs(y="% Change of Target Lesions", x="Months", title="Spider Plot of Percent Change in Target Lesion Measurements") +
    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'))

  ggplotly(p)
})
```

Spider Plot

Spider Plot of Percent Change in Target Lesion Measurements



Kaplan Meier dataset

Kaplan Meier dataset

Show 10 ▾ entries

Search:

	futime	fustat	age	resid.ds	rx	ecog-ps
1	59	1	72.3315	2	1	1
2	115	1	74.4932	2	1	1
3	156	1	66.4658	2	1	2
4	421	0	53.3644	2	2	1
5	431	1	50.3397	2	1	1
6	448	0	56.4301	1	1	2
7	464	1	56.937	2	2	2
8	475	1	59.8548	2	2	2
9	477	0	64.1753	2	1	1
10	563	1	55.1781	1	2	2

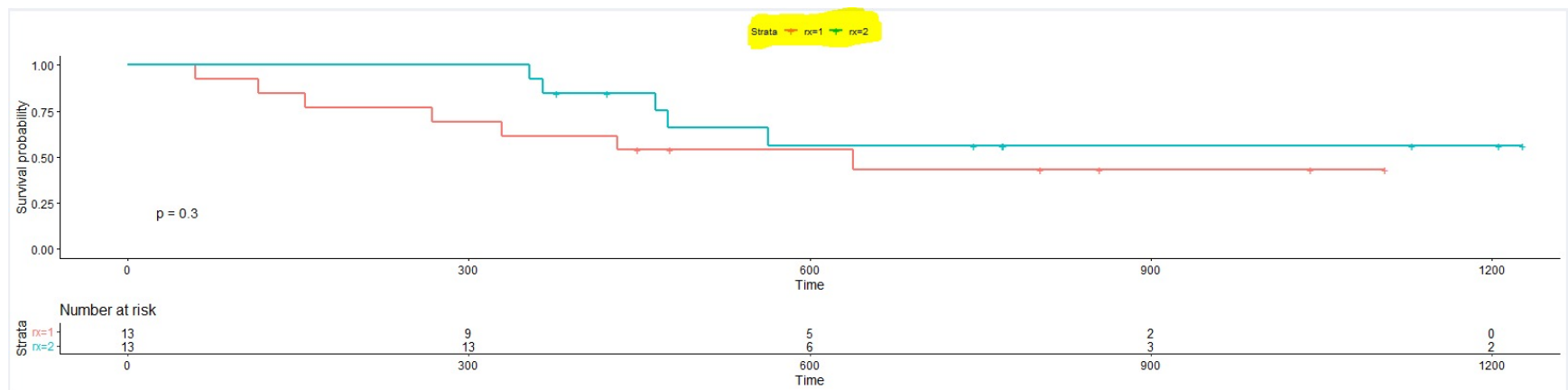
Showing 1 to 10 of 26 entries

Previous 1 2 3 Next

Kaplan Meier code

```
output$kpPlot=renderPlot({  
  datakp<- data_kp()  
  fit <- survfit(Surv(futime, fustat) ~ rx, data = datakp)  
  ggsurvplot(fit, data = datakp, pval=TRUE, risk.table = TRUE)  
})
```

Kaplan Meier plot



Conclusion

- R offers a good platform to create informative and interactive dashboards.
- ggplot2 is an effective and simple to use library in R to create interactive plots.
- Open-source programming languages such as R or Python are effective in clinical trial data analysis.
- Adapting and improvising are the key ingredients for growth.

References

- <https://www.rstudio.com/resources/cheatsheets/>
- <https://www.r-bloggers.com/2018/08/visualization-of-tumor-response-spider-plots/>
- Christine Teng (2020). Visualize Tumor Response Data using ggplot2 R Package through Examples. PHUSE US CONNECT 2020.
- Taniya Muliylil (2020). Oncology Graphs-Creation (Using SAS and R), Interpretation and QA. PHUSE US CONNECT 2020.

QUESTIONS