



BARI ANKIT (56)

Experiment No.7
Data Visualization: Use different R libraries for data visualization
Date of Performance:
Date of Submission:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No - 7

Aim - Data Visualization: Data Visualization: Use different R libraries for data visualization.

Objective:- To understand and apply the R libraries for visualization using python.

Description:-

1. **ggplot2** ggplot2 is an R data visualization library that is based on The Grammar of Graphics. ggplot2 can create data visualizations such as bar charts, pie charts, histograms, scatterplots, error charts, etc. using highlevel API. It also allows you to add different types of data visualization components or layers in a single visualization. Once ggplot2 has been told which variables to map to which aesthetics in the plot, it does the rest of the work so that the user can focus on interpreting the visualizations and take less time in creating them. But this also means that it is not possible to create highly customized graphics in ggplot2. But there are a lot of resources in the RStudio community and Stack Overflow which can provide help in ggplot2 when needed. Just like dplyr, if you want to install ggplot2, you can install the tidyverse or you can just install ggplot2 using `install.packages("ggplot2")`

2. **Plotly**

Plotly is a free open-source graphing library that can be used to form data visualizations. Plotly is an R package that is built on top of the Plotly JavaScript library (plotly.js) and can be used to create web-based data visualizations that can be displayed in Jupyter notebooks or web applications using Dash or saved as individual HTML files. Plotly provides more than 40 unique chart types like scatter plots, histograms, line charts, bar charts, pie charts, error bars, box plots, multiple axes, sparklines, dendrograms, 3-D charts, etc. Plotly also provides contour plots, which are not that common in other data visualization libraries. In addition to all this, Plotly can be used offline with no internet connection. You can install Plotly from CRAN using `install.packages('plotly')` or install the latest development version from GitHub using `devtools::install_github("ropensci/plotly")`.



3. Esquisse

Esquisse is a data visualization tool in R that allows you to create detailed data visualizations using the ggplot2 package. You can create all sorts of scatter plots, histograms, line charts, bar charts, pie charts, error bars, box plots, multiple axes, sparklines, dendrograms, 3-D charts, etc. using Esquisse and also export these graphs or access the code for creating these graphs. Esquisse is such a famous and easily used data visualization tool because of its drag and drops ability that makes it popular even among beginners. You can install Esquisse from CRAN using `install.packages("esquisse")` or install the development version from GitHub using `remotes::install_github("dreamRs/esquisse")`.

4. Lattice

Lattice is a data visualization tool that is primarily used to implement Trellis graphs in R. These Trellis graphs are used to view many complicated and multi-variable data sets at the same time so they can be compared. Since all these different plots end up looking like a Trellis, this is called a Trellis graph. Since Lattice is a high-level data visualization library, it can handle many of the typical graphics without needing many customizations. In case you want to extend the capabilities of Lattice, they can download the LatticeExtra package which is an extended version. You can install Lattice from CRAN using `install.packages("lattice")` or install the development version from GitHub using `remotes::install_github("deepayan/lattice")`.

5. RGL

The RGL package in R is created specifically for making 3-D data visualizations and data plots. It has many graphics commands that work in 3 dimensions but is modeled loosely after the classic 2-D graphics in R. RGL is also inspired by the grid package in R but it is incompatible with it. However, seasoned R coders can easily use RGL because of an existing familiarity with the grid. And RGL is very cool! It has a lot of options for 3-D shapes, various lighting effects, creating new shapes, and also animations. You can install



RGL from CRAN using `install.packages("rgl")`.

The **dygraphs** package is an **R** interface to the JavaScript charting library **dygraphs** that are used to provide various charts for visualizing data sets. This package can be used for creating various interactive visualizations with zooming, and panning options along with default mouse-over labels. **dygraphs** also provides support for various graph overlays such as point annotations, shaded regions, event lines, etc. You can also plot the **xts** time series objects automatically. However, all of these features do not come at the expense of speed in **dygraph**. Rather, it can provide maximal interactivity even with millions of data points in the visualization. You can install **RGL** from CRAN using `install.packages("dygraphs")`.

7. Leaflet

Just like **dygraphs**, the **Leaflet** package is an **R** interface to the JavaScript **Leaflet** library that is extremely popular. **Leaflet** is very useful in creating interactive but lightweight maps that are seen on various websites such as the Washington Post, the New York Times, etc. There are many useful features in this package such as interactive panning and zooming in the charts, the option to combine Polygons, Lines, Popups, etc. to create charts, embed maps in **knitr**, create maps in mercator projections that are nonspherical and so on. The **Leaflet** package can be used at the **R** console after installing it from CRAN using the command `install.packages("leaflet")`.

All of these **R** Libraries for Data Visualization are excellent options if you want to create data visualizations. Each of these libraries has its strengths and you can choose the best one depending on the type of visualization or data science project you want to create. Now that you know these libraries, go on and create beautiful yet informative data visualizations using them!



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Program

```
# Load necessary libraries
```

```
library(ggplot2)
```

```
# Create a sample dataset
```

```
data <- data.frame(
```

```
  category = c("A", "B", "C", "D"),
```

```
  values = c(4, 3, 5, 7)
```

```
)
```

```
# Bar plot
```

```
ggplot(data, aes(x = category, y = values, fill = category)) +
```

```
  geom_bar(stat = "identity") +
```

```
  labs(title = "Bar Plot Example", x = "Category", y = "Values") +
```

```
  theme_minimal()
```

```
# Create a scatter plot with another dataset
```

```
mpg_data <- ggplot2::mpg
```

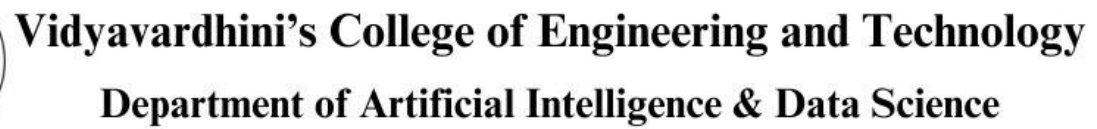
```
# Scatter plot
```

```
ggplot(mpg_data, aes(x = displ, y = hwy, color = class)) +
```

```
  geom_point() +
```

```
  labs(title = "Displacement vs. Highway MPG", x = "Displacement (L)", y = "Highway MPG") +
```

```
  theme_minimal()
```



```
plot_ly(data = mpg_data, x = ~displ, y = ~hwy, color = ~class,  
        type = "scatter", mode = "markers",  
        marker = list(size = 10)) %>%  
  layout(title = "Interactive Scatter Plot of Displacement vs. Highway MPG",  
         xaxis = list(title = "Displacement (L)"),  
         yaxis = list(title = "Highway MPG"))
```



```
# Load necessary package

library(shiny)

library(ggplot2)


# Define the UI

ui <- fluidPage(

  titlePanel("Simple Shiny App"),

  sidebarLayout(

    sidebarPanel(

      selectInput("class", "Select Vehicle Class:",

                  choices = unique(mpg_data$class)),

      actionButton("update", "Update Plot")

    ),

    mainPanel(

      plotOutput("scatterPlot")

    )

  )

)


# Define the server

server <- function(input, output) {

  observeEvent(input$update, {

    output$scatterPlot <- renderPlot({

      filteredData <- subset(mpg_data, class == input$class)

      ggplot(filteredData, aes(x = displ, y = hwy)) +
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

```
geom_point() +  
  
labs(title = paste("MPG for class:", input$class),  
      x = "Displacement (L)", y = "Highway MPG") +  
  
theme_minimal()  
  
})  
  
})  
  
}  
  
  
# Run the app  
  
shinyApp(ui = ui, server = server)
```

Conclusion-

R provides a rich array of libraries for data visualization, each with its own strengths. The choice depends on the type of visualization needed, the interactivity required, and personal preference.

ggplot2 is ideal for static and dynamic visualizations, especially suited for publication-quality graphics.

lattice is great for conditioned plots and complex layouts.

plotly enhances user engagement with interactive plots.

shiny lets you create comprehensive web applications for data exploration.

By understanding and leveraging these libraries, you can create informative and visually appealing data presentations that facilitate better insights and decision-making.