Experiment No. 7

Aim - Data Visualization: Data Visualization: Use different R libraries for data visualization.
Objective:- Tounderstand andapplytheRlibrariesforvisualization usingpython.

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 high-level 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
canjust 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 ofthe Plotly JavaScript library (plotly.js) and can be used to create web-based data visualizations that can be displayed in Jupyter notebooks orweb applications using Dash orsaved 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 offiine 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 fromGitHubusing 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 usmg 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 pointsinthe visualization. Youcan install RGL from CRANusing 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 non-

spherical and so on. The Leaflet package can be used atthe R console afterinstalling 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!

Program

1. Write a R program to demonstrate use of the ggplot library.

2. Write a R program to demonstrate use of plotly library.

```r
# Load necessary libraries
library(ggplot2)
library(plotly)
library(lattice)

# Generate example data
data <- data.frame(
  x = 1:10,
  y = rnorm(10),
  z = sample(letters[1:2], 10, replace = TRUE)
)

# Create ggplot2 plot
ggplot(data, aes(x = x, y = y, color = z)) +
  geom_point() +
  ggtitle("GGPlot2 Scatter Plot")

# Create plotly interactive plot
plot_ly(data, x = ~x, y = ~y, color = ~z, type = 'scatter', mode = 'markers') %>%
  layout(title = "Plotly Scatter Plot", xaxis = list(title = "X-axis"), yaxis = list(title = "Y-axis"))

# Create lattice plot
xyplot(y ~ x | z, data = data, type = "p", layout = c(1, 2), main = "Lattice Scatter Plot")
```

Conclusion-

Features of plotly and ggplot2 libraries are

Plotly and ggplot2 are powerful visualization libraries in Python and R, respectively, offering rich features for creating interactive and publication-quality plots. Plotly provides interactive plots that can be easily customized and shared online, with support for various chart types, animations, and dashboards. It also offers integration with web technologies like JavaScript for creating dynamic visualizations. On the other hand, ggplot2 follows a grammar of graphics approach, allowing users to construct complex plots through a layered system of aesthetics and geometric objects. It offers extensive customization options, elegant syntax, and seamless integration with the tidyverse ecosystem, making it popular for exploratory data analysis and data visualization in R. Both libraries support a wide range of statistical graphics and facilitate the creation of visually appealing and informative plots for data analysis and presentation purposes.

What is Plotly?

Plotly is an open-source library in Python and R for creating interactive and publication-quality visualizations. It offers a wide range of chart types, including scatter plots, line charts, bar charts, and heatmaps. Plotly enables users to create interactive plots with features such as zooming, panning, tooltips, and animations, making it suitable for both exploratory data analysis and presentation purposes. Additionally, it provides tools for building dashboards and sharing visualizations online.

Which programming languages can be used with Plotly?

Plotly can be used with several programming languages, including Python, R, JavaScript, and Julia. It provides libraries and APIs for each of these languages, allowing users to create interactive visualizations and dashboards. Plotly's versatility across multiple languages makes it a popular choice for data visualization and analysis in various domains. Additionally, Plotly offers seamless integration with web technologies for creating dynamic and interactive plots.