BDH LAB-Practical 09

|  |
| --- |
| **Name: Himanshu Dhomane** |
| **Roll no: 43** |
| **Registration No: 21070297** |

**Aim: Analyzing Various Data Visualization Methods Using R.**

**Theory:**

The objective of this practical journal is to explore and analyze various data visualization techniques in R. Data visualization is essential for understanding data patterns, relationships, and insights. R, with its rich ecosystem of libraries, offers powerful tools for effective data visualization.

Tools and Libraries:

* R Programming Language
* Libraries: ggplot2, plotly, lattice, dplyr

**Implementation:**

**Step 1: Install R and RStudio**

1. **Download R:**

Visit the CRAN website and download R for your operating system (Windows, Mac, or Linux). Install it following the on-screen instructions.

1. **Download RStudio:**

Visit the RStudio website and download the free version of RStudio Desktop. Install it after downloading.

**Step 2: Launch RStudio and Verify Setup**

1. **Open RStudio: Double-click the RStudio icon to launch it.**
2. **Verify R Installation: In the Console (bottom left), type the following and press Enter: version**

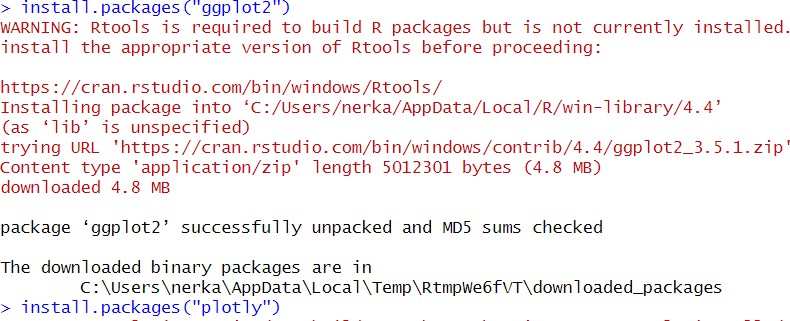
You should see details about your R version and environment. This confirms that R and RStudio are properly linked.

**Step 3: Install Required Packages for Data Visualization**

To visualize data, you’ll need libraries like ggplot2, plotly, and dplyr. Use the following commands to install them in the Console.

**# Install ggplot2 for visualizations install.packages("ggplot2")**

**# Install plotly for interactive plots install.packages("plotly")**

**# Install dplyr for data manipulation (optional) install.packages("dplyr")**

**Step 4: Load Libraries in Your Script**

Add the following code to your R script to load the libraries you installed.

**# Load libraries library(ggplot2) library(plotly) library(dplyr)**

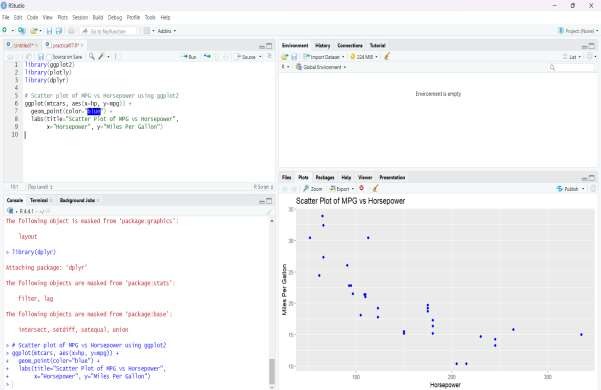
**# Optional for data manipulation**

**Step 6: Write and Run Your First Visualization Code**

Here’s an example code for a scatter plot using the built-in mtcars dataset.

1. **Scatter plot of MPG vs Horsepower using ggplot2 ggplot(mtcars, aes(x=hp, y=mpg)) + geom\_point(color="blue") +**

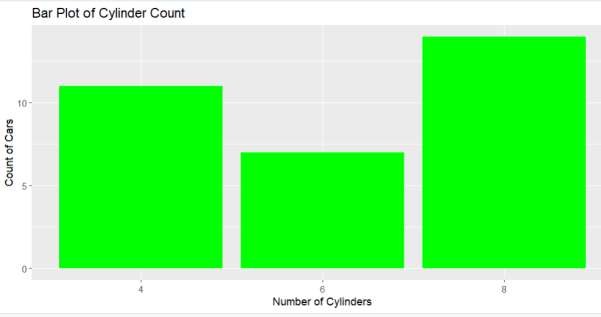
**labs(title="Scatter Plot of MPG vs Horsepower", x="Horsepower", y="Miles Per Gallon")**



1. **Bar Plot:**

Bar plots are useful for visualizing categorical data. # Bar plot: Number of cars by cylinder ggplot(mtcars, aes(x=factor(cyl))) + geom\_bar(fill="green") +

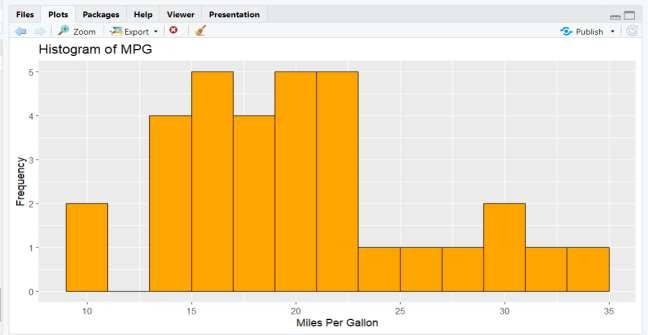
labs(title="Bar Plot of Cylinder Count", x="Number of Cylinders", y="Count of Cars")



1. **Histogram:** A histogram helps to visualize the distribution of a single continuous variable.

# Histogram: Distribution of MPG ggplot(mtcars, aes(x=mpg)) +

geom\_histogram(binwidth=2, fill="orange", color="black") + labs(title="Histogram of MPG", x="Miles Per Gallon", y="Frequency")

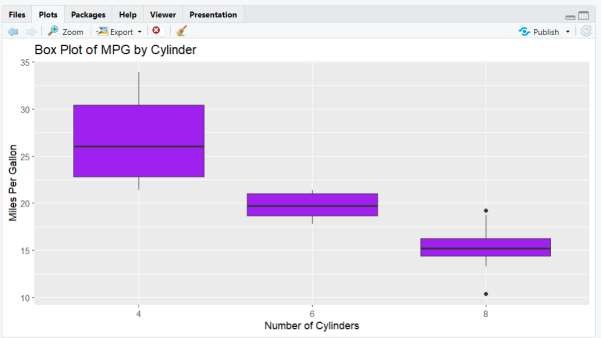


1. **Box Plot:**

Box plots are used to visualize the distribution of data based on a summary of minimum, first quartile, median, third quartile, and maximum values.

# Box plot: MPG by Cylinder ggplot(mtcars, aes(x=factor(cyl), y=mpg)) + geom\_boxplot(fill="purple") +

labs(title="Box Plot of MPG by Cylinder", x="Number of Cylinders", y="Miles Per Gallon")

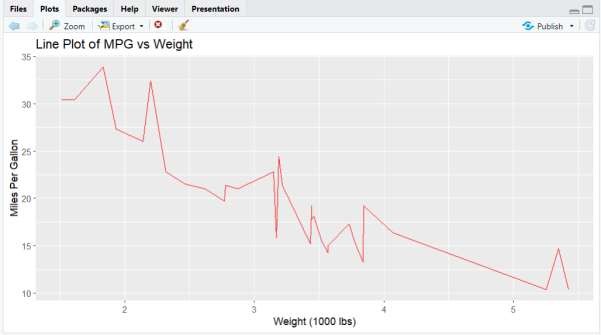


1. **Line Plot:**

Line plots are effective for time series data or trends over a continuous variable. # Line plot: MPG vs Weight

ggplot(mtcars, aes(x=wt, y=mpg)) + geom\_line(color="red") +

labs(title="Line Plot of MPG vs Weight", x="Weight (1000 lbs)", y="Miles Per Gallon")

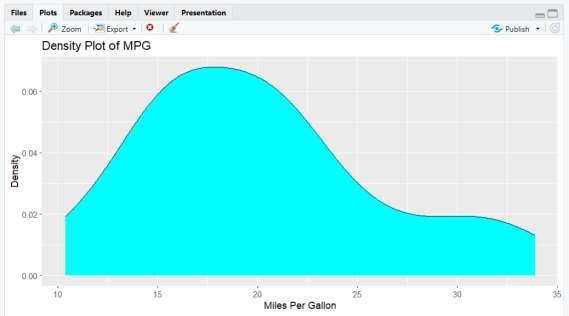


1. **Density Plot:**

A density plot visualizes the distribution of a variable and shows the probability density of the data.

# Density plot: Distribution of MPG ggplot(mtcars, aes(x=mpg)) + geom\_density(fill="cyan") +

labs(title="Density Plot of MPG", x="Miles Per Gallon", y="Density")

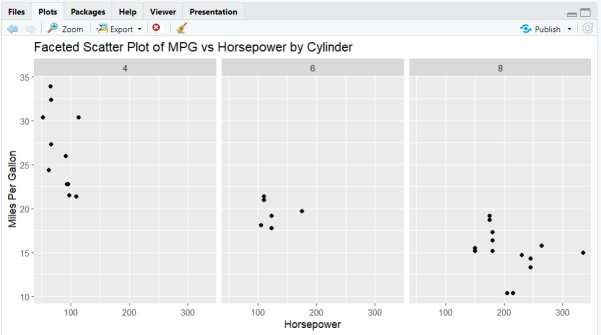


1. **Faceted Plot:**

Faceted plots are useful for visualizing data with multiple subsets. # Faceted plot: MPG vs Horsepower, faceted by Cylinder ggplot(mtcars, aes(x=hp, y=mpg)) +

geom\_point() + facet\_wrap(~cyl) +

labs(title="Faceted Scatter Plot of MPG vs Horsepower by Cylinder", x="Horsepower", y="Miles Per Gallon")



1. **Interactive Plot with plotly:** To make interactive plots, the plotly library can be used.

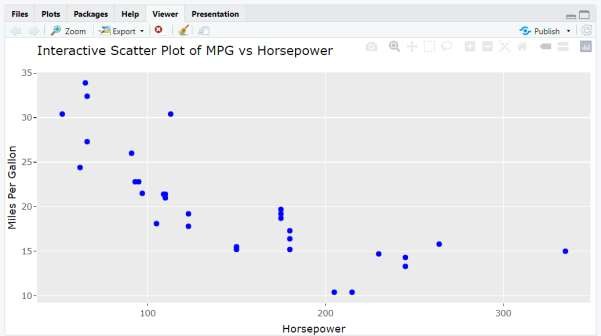
# Load plotly library library(plotly)

# Interactive scatter plot: MPG vs Horsepower

p <- ggplot(mtcars, aes(x=hp, y=mpg)) + geom\_point(color="blue") +

labs(title="Interactive Scatter Plot of MPG vs Horsepower", x="Horsepower", y="Miles Per Gallon")

# Convert ggplot to interactive plot ggplotly(p)



**Conclusion:**

In this journal, we demonstrated various data visualization techniques using R. Different plots serve different purposes depending on the type of data and the story that needs to be conveyed. The ggplot2 package provides great flexibility for creating static visualizations, while plotly enables interactivity. Using these tools, one can effectively communicate insights derived from data.