Practical No. 10

Title: Study and Implementation of Data Visualization with `ggplot2` in R

Introduction:

Data visualization is one of the most powerful tools in data analysis. It helps to uncover patterns, trends, and insights by presenting data graphically. In R, the `ggplot2` package is a widely-used and highly flexible system for creating elegant and informative visualizations. Based on the Grammar of Graphics, `ggplot2` allows users to build complex plots layer by layer with consistent syntax and customization capabilities.

Objective:

- To understand the grammar of graphics and the structure of `ggplot2`.
- To learn how to create various types of visualizations using `ggplot2`.
- To implement basic and advanced visualizations such as bar charts, line plots, scatter plots, histograms, and boxplots.
- To explore aesthetics, themes, and customizations in `ggplot2`.

Tools Required:

- **Software:** R (version 4.0 or later)
- **IDE:** RStudio (recommended)
- **Libraries:** `ggplot2`, `tidyverse`

Theory:

1. What is `ggplot2`?

`ggplot2` is an R package based on the **Grammar of Graphics**, which provides a framework for building plots layer by layer. Each plot in `ggplot2` starts with:

```
ggplot(data, aes(x = ..., y = ...)) + geometry + additional layers
```

The idea is to define the data and aesthetic mappings (`aes()`), then add geometric objects (like `geom_bar()`, `geom_point()`, etc.) to represent the data.

Structure of a `ggplot2` Plot:

Component	Description
`ggplot()`	Initializes the plot with data and aesthetics
`aes()`	Aesthetic mapping (e.g., x, y, color, size)
`geom_` functions	Defines the type of plot (e.g., `geom_point`, `geom_bar`, `geom_line`)
`facet_`	For multi-panel plots

Component	Description
`theme()`	Customizes the appearance
`labs()`	Adds labels and titles
`scale_`	Customize scales for colors, axes, etc.

Implementation: Basic Visualizations in `ggplot2`

Sample Dataset:

```
library(ggplot2)

# Sample data
df <- data.frame(
  Category = c("A", "B", "C", "D"),
  Value = c(30, 70, 45, 85)
)
```

1. Bar Chart

```
ggplot(df, aes(x = Category, y = Value)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Bar Chart Example", x = "Category", y = "Value")
```

2. Scatter Plot

```
# Using built-in dataset
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point(color = "darkred") +
  labs(title = "Scatter Plot: MPG vs Weight", x = "Weight", y = "Miles per
Gallon")
```

3. Line Plot

```
df_line <- data.frame(
  Year = 2015:2020,
  Sales = c(100, 150, 170, 120, 180, 210)
)

ggplot(df_line, aes(x = Year, y = Sales)) +
  geom_line(color = "blue", size = 1.2) +
  geom_point(color = "black") +
  labs(title = "Sales Trend Over Years", x = "Year", y = "Sales")
```

4. Histogram

```
ggplot(mtcars, aes(x = mpg)) +
  geom_histogram(bins = 10, fill = "green", color = "black") +
  labs(title = "Distribution of MPG", x = "Miles per Gallon", y = "Count")
```

5. Boxplot

```
ggplot(mtcars, aes(x = factor(cyl), y = mpg)) +
  geom_boxplot(fill = "orange") +
  labs(title = "MPG by Number of Cylinders", x = "Cylinders", y = "Miles per
Gallon")
```

6. Faceted Plot

```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  facet_wrap(~cyl) +
  labs(title = "Faceted Scatter Plot by Cylinder")
```

Customization with `theme()` and `labs()`

`theme()` allows you to control the appearance of text, axes, backgrounds, and more.

```
ggplot(mtcars, aes(wt, mpg)) +
  geom_point(color = "purple") +
  labs(title = "Customized Plot", x = "Weight", y = "MPG") +
  theme_minimal()
```

Conclusion:

The `ggplot2` package in R is a versatile and powerful tool for creating high-quality data visualizations. By understanding its layered structure, you can create a wide variety of plots, from simple bar charts to complex multi-faceted plots. `ggplot2` not only enhances data storytelling but also makes your visualizations publication-ready. Mastery of this package is essential for effective data exploration and presentation in data science.