Bharati Vidyapeeth
Deemed to be University

BVUDET

Department of Engineering and Technology

Plot no. KC-1, Sector 3, Kharghar, Navi Mumbai-410210

Subject: Computing Lab - III | Experiment No - 04 (3rd YEAR CSE-AIML 2023-2024)

Roll No: 11	Name: Kamran Khan
Class: CSE-AIML	Batch: B1
PRN: 2143110133	Date of Experiment: 09 / 02 / 2024
Marks (Out of 25):	Date of Submission: __ / 02 / 2024

Aim: Data Visualization using R Programming Language.

Theory:

In data visualization, we use graphical representations of data to summarize and communicate information effectively. R programming language offers powerful tools and libraries for creating various types of visualizations.

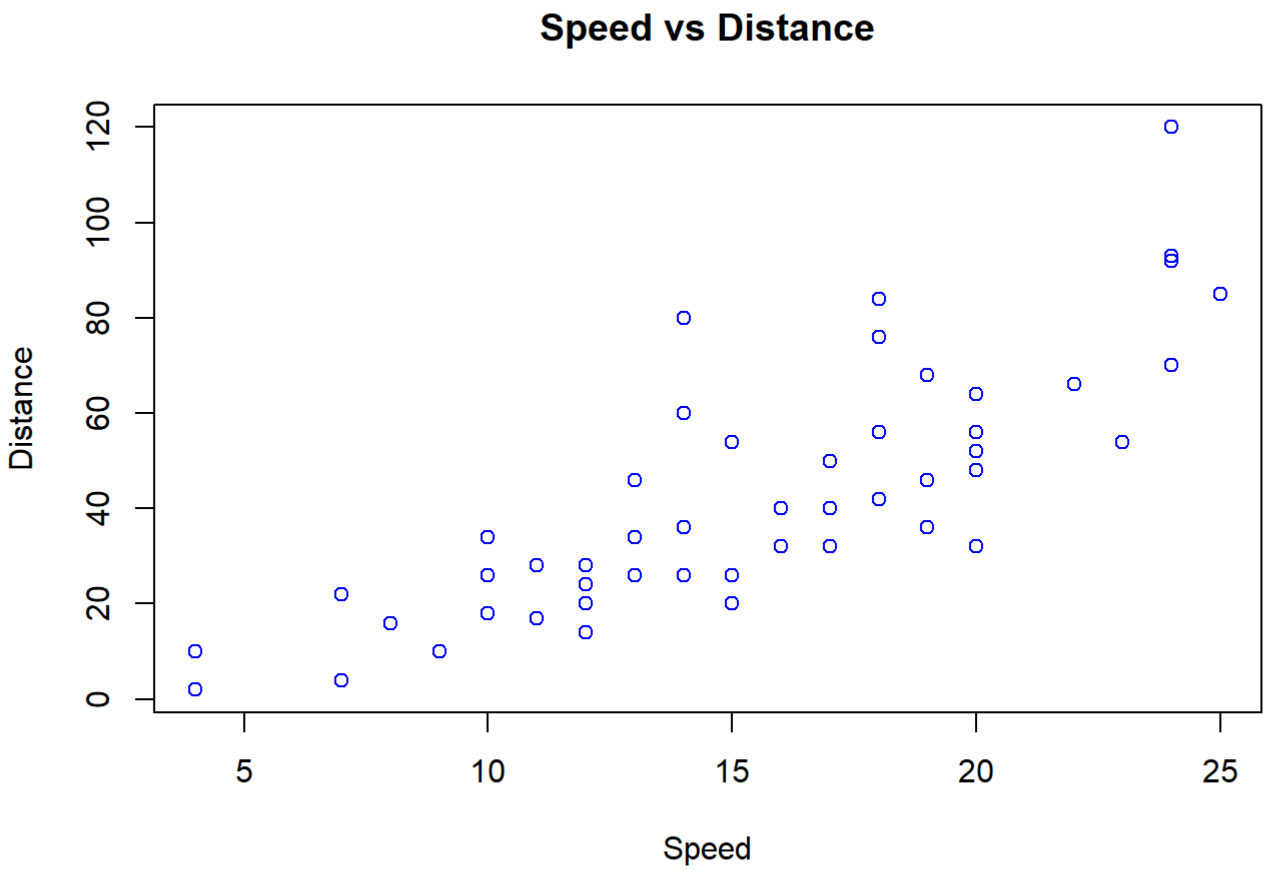
```
d <- cars
head(d)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

Plot Function:

The `plot()` function in R is a versatile tool for creating scatter plots, line plots, histograms, and more. It’s a basic function that can be customized with various arguments to tailor the plot according to the data and visualization needs.

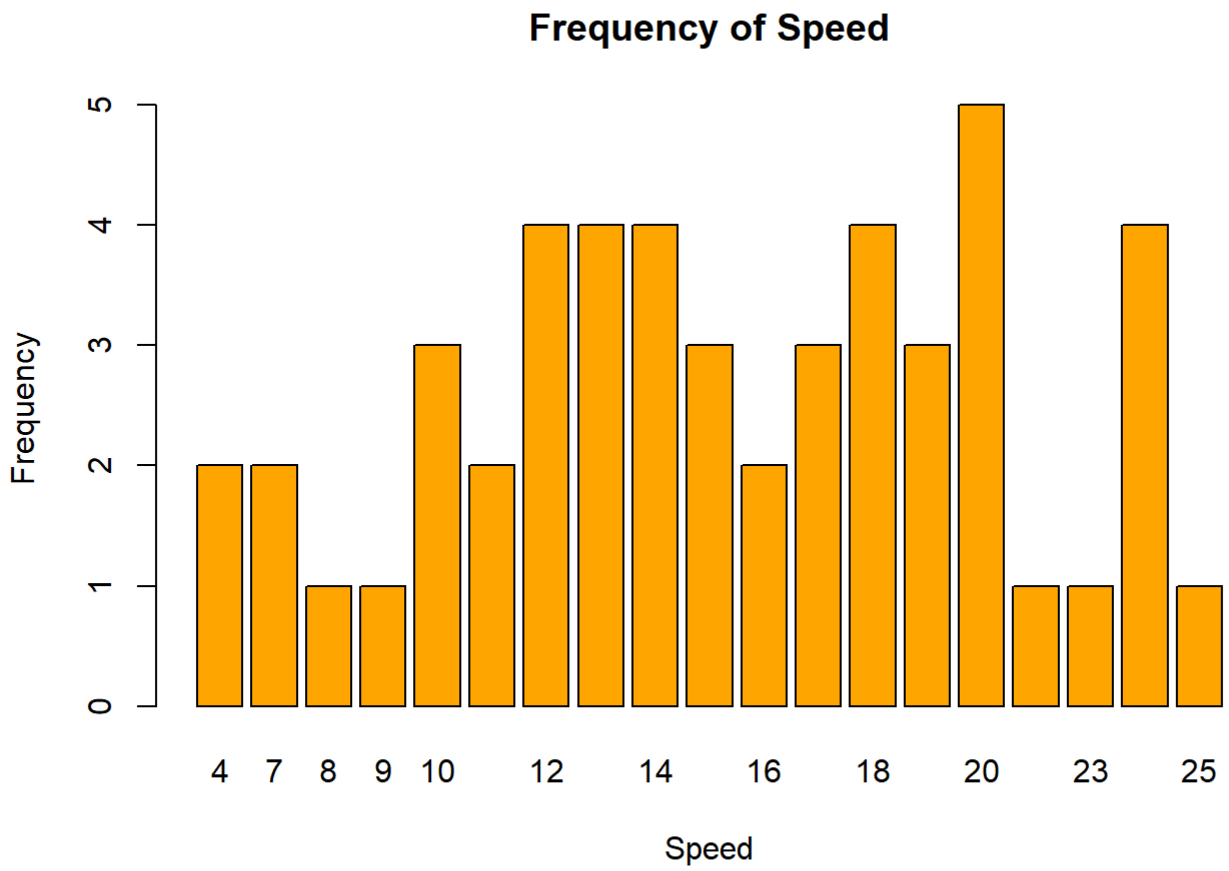
```
plot(x = d$speed, y = d$dist, main = "Speed vs Distance", xlab = "Speed", ylab = "Distance",
col = "blue")
```



Bar Chart:

Bar charts are useful for comparing the frequency, count, or other summary statistics of different categories. In R, you can create bar charts using the `barplot()` function.

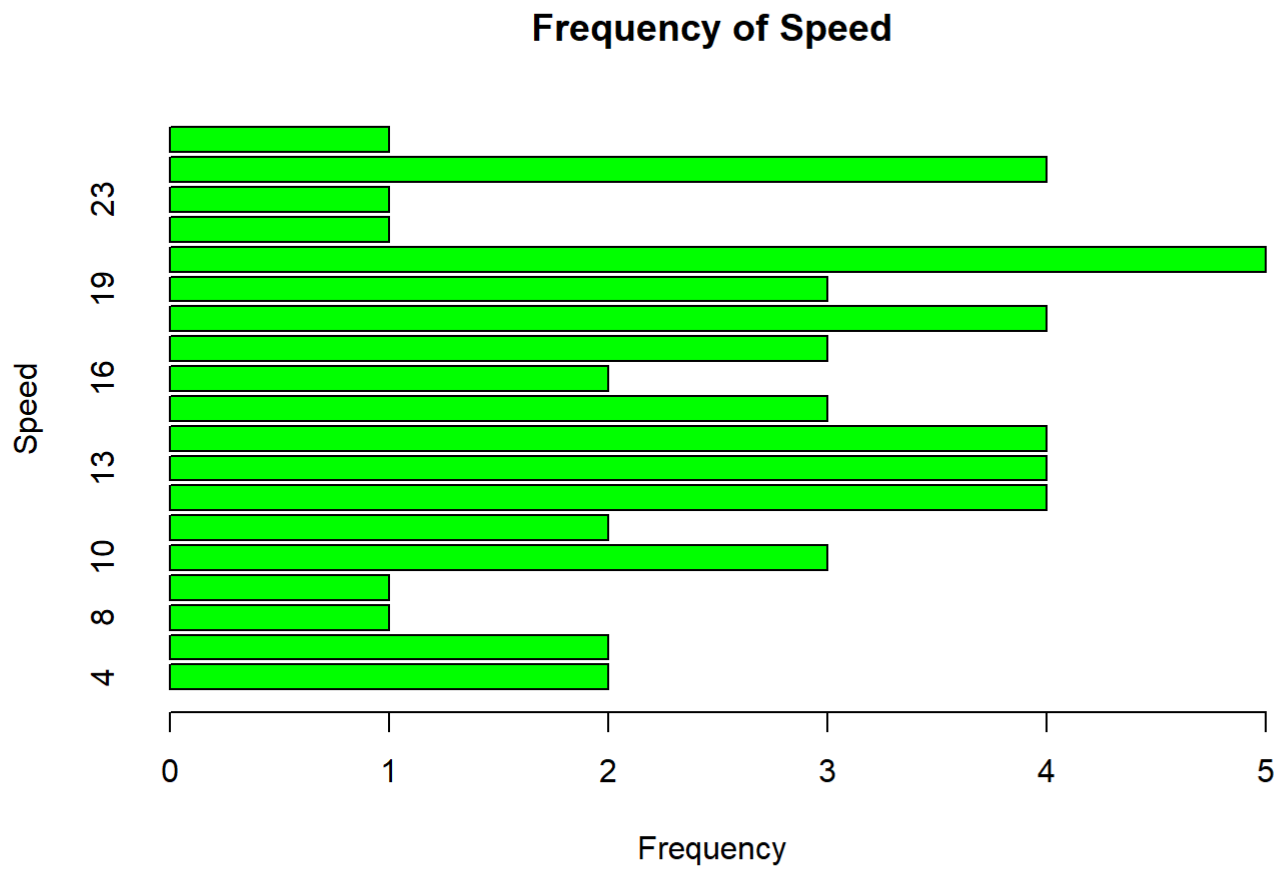
```
barplot(table(d$speed), main = "Frequency of Speed", xlab = "Speed", ylab = "Frequency", col
= "orange")
```



Horizontal Bar Graph:

Horizontal bar graphs are similar to bar charts but with the bars displayed horizontally. You can achieve this in R using the barplot() function with the horiz = TRUE argument.

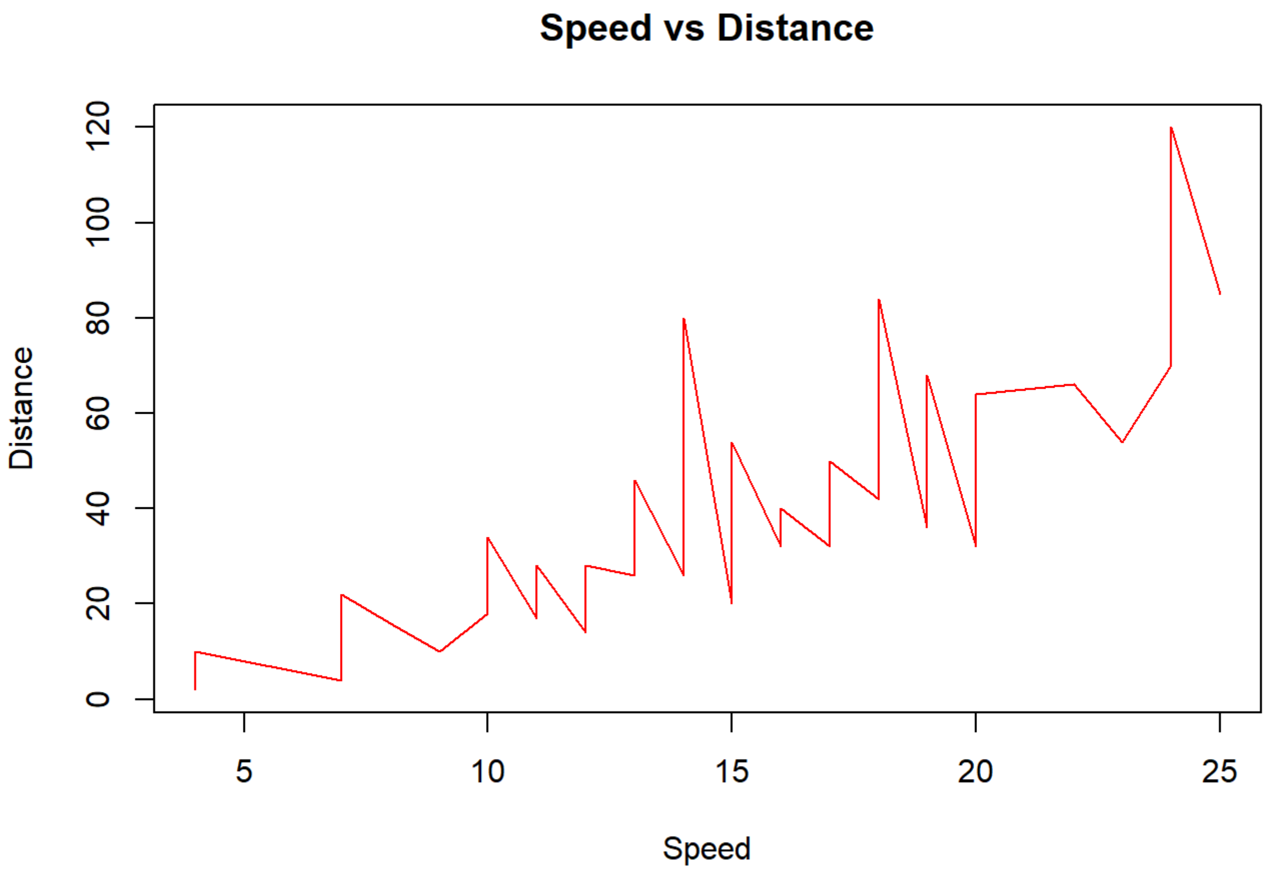
```
barplot(table(d$speed), main = "Frequency of Speed", xlab = "Frequency", ylab = "Speed", horiz = TRUE, col = "green")
```



Line Graph:

Line graphs are effective for showing trends or changes in data over time or other continuous variables. In R, you can create line graphs using the `plot()` function with the type = “l” argument.

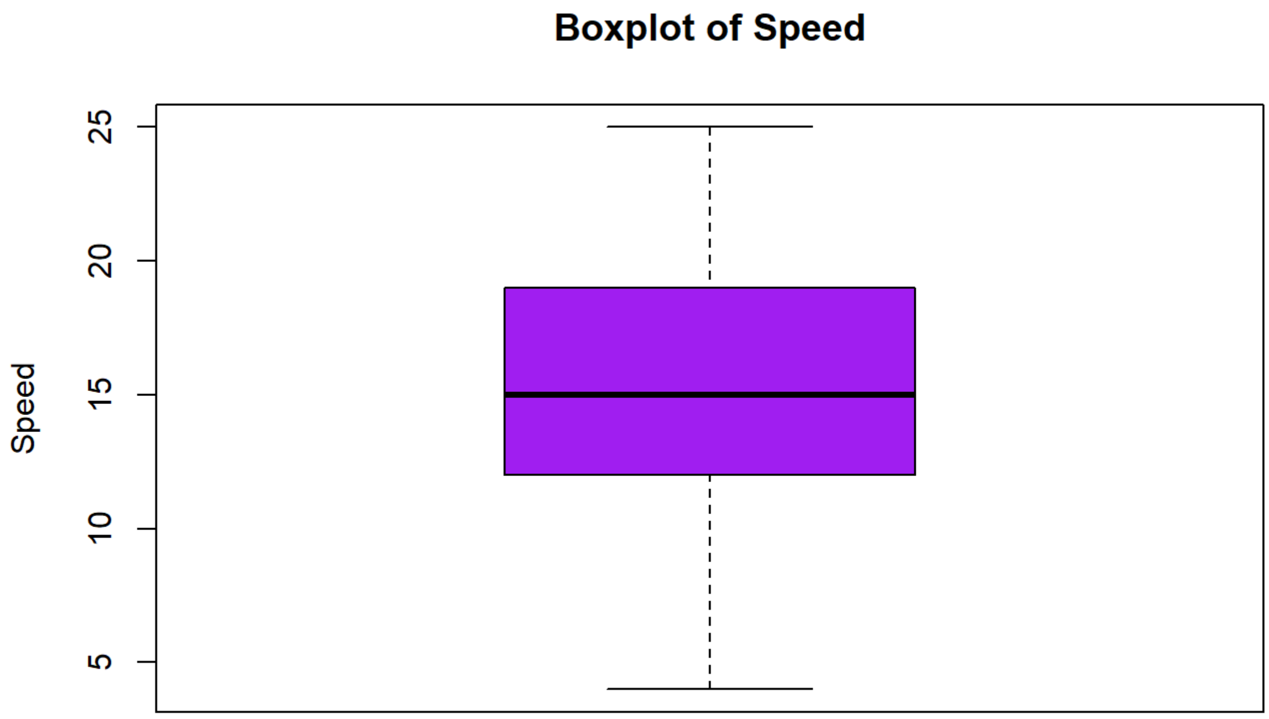
```
plot(x = d$speed, y = d$dist, type = "l", main = "Speed vs Distance", xlab = "Speed", ylab =
"Distance", col = "red")
```



Box Graph:

Box plots (box-and-whisker plots) are useful for displaying the distribution of a dataset and identifying outliers. In R, you can create box plots using the `boxplot()` function.

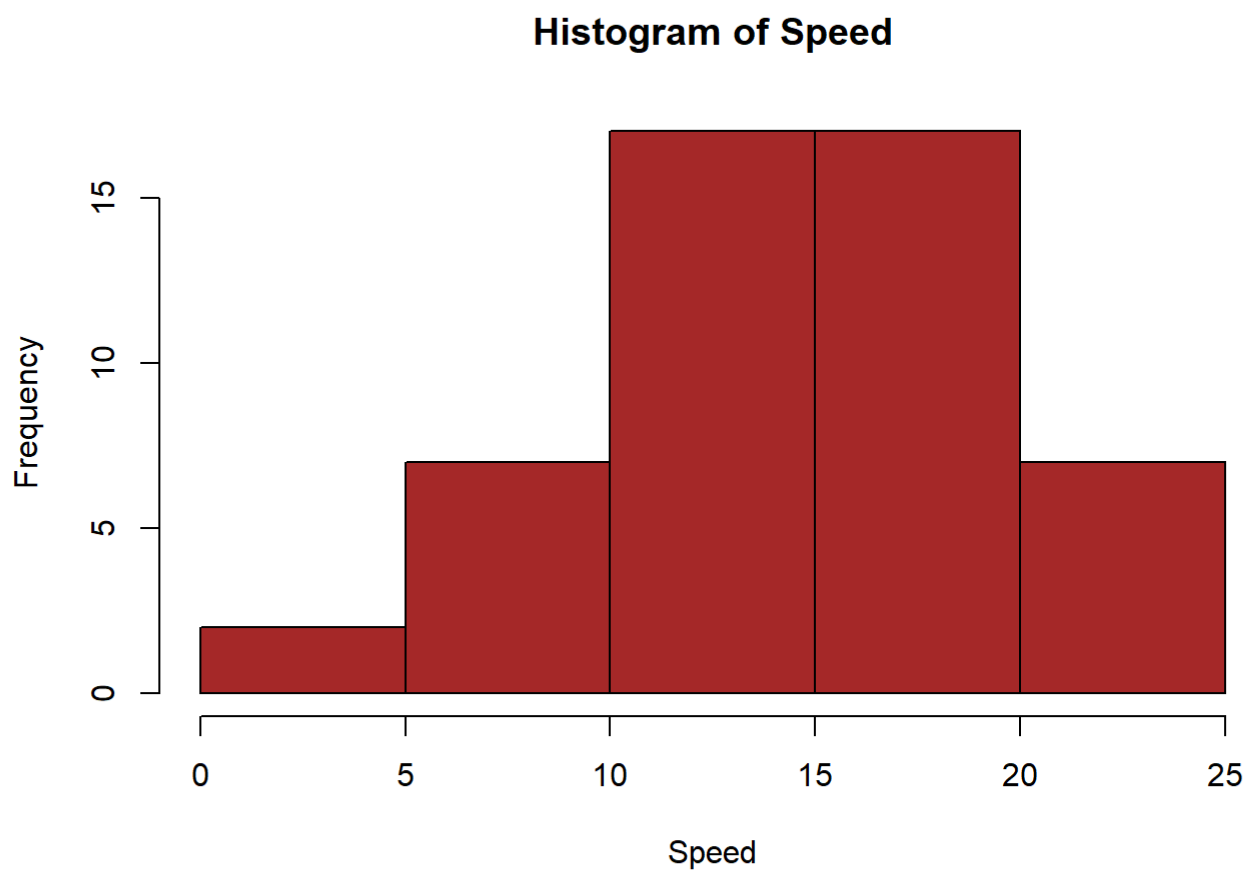
```
boxplot(d$speed, main = "Boxplot of Speed", ylab = "Speed", col = "purple")
```



Histogram:

Histograms are used to visualize the distribution of a continuous variable. In R, you can create histograms using the `hist()` function.

```
hist(d$speed, main = "Histogram of Speed", xlab = "Speed", ylab = "Frequency", col = "brown")
```



GGPlot2 Library:

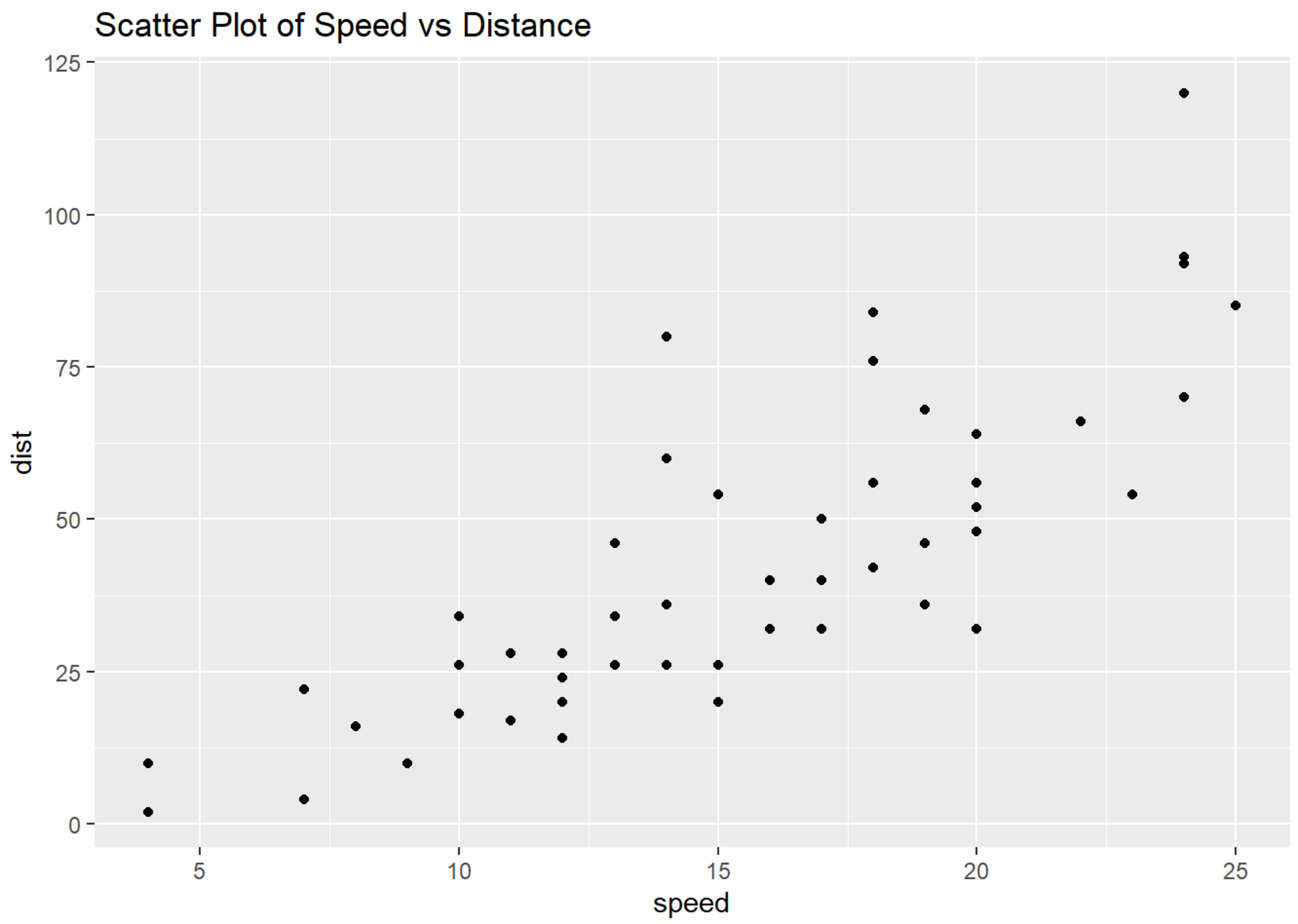
`ggplot2` is a popular package in R for creating elegant and sophisticated visualizations. It provides a flexible and powerful syntax for building various types of plots.

```
library(ggplot2)
```

Geom Function:

In `ggplot2` , the `geom` functions are used to specify the type of geometric object (points, lines, bars, etc.) to be plotted. Each geom function adds a layer to the plot.

```
ggplot(data = d, aes(x = speed, y = dist)) + geom_point() + labs(title = "Scatter Plot of Spe
ed vs Distance")
```



Conclusion:

This experiment familiarized us with data visualization techniques in R. By creating diverse plots such as scatter plots, bar charts, and histograms, we gained practical insights into representing and analyzing data effectively.

Signature of Lab Incharge

(Prof. Supriya Khaitan)