Part A

1. For a given set of training data examples stored in a .CSV file, compute the Mean, Median, Variance, Standard Deviation, Range and Quartiles of one of the attributes using R programming.

```
#Importing data set and printing the first few values
myData = read.csv("C:/Users/navbs/OneDrive/Desktop/CardioGoodFitness.csv")
print(head(myData))
#Mean
mean=mean(myData$Age)
print(mean)
#Median
median = median(myData$Age)
print(median)
# Variance
variance = var(myData$Age)
print(variance)
# Standard Deviation
standard = sd(myData$Age)
print(standard)
# Range
# Calculate the maximum
max = max(myData$Age)
# Calculate the minimum
min = min(myData$Age)
# Calculate the range
range = max - min
print(range)
# Calculating Quartiles
quartiles = quantile(myData$Age)
print(quartiles)
```

2. Write an R program to perform the following operations: Create a file, Writing into a file, Renaming a file, Reading a file, Listing all files, Copy a file.

```
#Create a file
file.create("Nav.txt")
#Writing into a file
write.table(x = iris[1:10, ], file = "Nav.txt")
# Reading a text file
myData = read.table(file = "Nav.txt ")
print(myData)
```

```
#Renaming a file
file.rename("Nav.txt", "Abc.txt")
#Lising the table
list.files()
# Copy a file
file.copy"C:/Users/navbs/OneDrive/Documents/Abc.txt", "D:/")
list.files("D:/")
```

3. Write an R program to perform the following operations on strings: Concatenate two strings, Compare two strings, Reverse the string and Check if a given string is a palindrome or not.

```
# Create two strings
string1 = "data"
string2 = "science"
# using paste() to concatenate two strings
result = paste(string1, string2)
print(result)
# Compare both strings
result = toupper(string1) == toupper(string2)
print(result)
# Reverse the string
text = "data science"
reversed_text = rev(strsplit(text, "")[[1]])
reversed_text = paste(reversed_text, collapse = "")
cat("Reversed String:", reversed_text, "\n")
# Check if a given string is a palindrome
text2 = "madam"
# Reverse the string
reversed_string = rev(strsplit(text2, "")[[1]])
reversed_string = paste(reversed_string, collapse = "")
# Check if the input string is equal to its reverse
if (text2 == reversed_string) {
cat("The string is a palindrome.\n")
} else {
cat("The string is not a palindrome.\n")
}
```

4. Write an R program to demonstrate the use of the following String manipulation functions in R: nchar, toupper, tolower, substr, grep, paste, strsplit, sprintf, cat and sub functions.

```
string = "Hello My Name Is TechVidvan"
```

```
# nchar function
nchar(string)
strvec = c(string,"HI", "hey", "haHa")
nchar(strvec)
# toupper function
toupper(string)
toupper(strvec)
# tolower function
tolower(string)
tolower(strvec)
# substr function
substr(string, 5, 20)
# grep function
grep("Tech", string)
# paste function
paste("hello", "techvidvan", string, sep = "-")
# strsplit function
strsplit(string,'e')
# sprintf function
count = 5L
name = "Bob"
place = "pocket"
sprintf("There are %d dollars in %s's %s", count, name, place)
# cat function
cat("hello","this","is","Techvidvan",sep = "-")
# sub function
sub("My Name Is", "I Am", string)
```

Part B

5. Write an R program to create the following basic plots: Scatter plot, Line graph, Bar plot and Histogram.

```
# Sample data
x = c(1, 2, 3, 4, 5)
y = c(2, 4, 6, 8, 10)
# Scatter Plot
plot(x, y, type = "p", col = "blue", pch = 16, xlab = "X-axis", ylab = "Y-axis", main = "Scatter Plot Example")
# Line Plot
plot(x, y, type = "l", col = "red", lwd = 2, xlab = "X-axis", ylab = "Y-axis", main = "Line Plot
```

```
Example")
# Bar Plot
barplot(y, names.arg = x, col = "green", xlab = "X-axis", ylab = "Y-axis", main = "Bar Plot
Example")
# Histogram
hist(y, col = "purple", xlab = "X-axis", ylab = "Frequency", main = "Histogram Example")
```

6. Write an R program to create a 2D and 3D Pie chart with slice percentage & legend.

```
# Create data for the graph
data = c(23, 56, 20, 63)
labels = c("Mumbai", "Pune", "Chennai", "Bangalore")
piepercent= round(100 * data / sum(data), 1)
# Plot the 2D pie chart
pie(data, labels = piepercent, main = "City pie chart", col = rainbow(length(data)))
legend("topright", c("Mumbai", "Pune", "Chennai", "Bangalore"), cex=0.5,
fill=rainbow(length(data)))
# Get the library
library(plotrix)
# Plot the 3D pie chart
pie3D(data, labels = piepercent, main = "City pie chart", col = rainbow(length(data)))
legend("topright", c("Mumbai", "Pune", "Chennai", "Bangalore"), cex = 0.5,
fill=rainbow(length(data)))
```

7. Using the in-build Iris dataset and ggplot2 package, write an R program to create Scatter plot, Line graph and Bar plot with chart titles and axis titles.

```
# install.packages("ggplot2")
library(ggplot2)
# Load the iris dataset (it's built-in)
data(iris)
# Bar Plot
bar_plot = ggplot(iris, aes(x = Species)) + geom_bar(fill = "skyblue") + labs(title = "Bar Plot
of Iris Species", x = "Species", y = "Count")
print(bar_plot)
# Line Plot
line_plot = ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
geom_line() + labs(title = "Line Plot of Sepal Length vs Sepal Width", x = "Sepal Length", y
= "Sepal Width")
print(line_plot)
# Scatter Plot
```

```
scatter_plot = ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species, shape = Species)) + geom_point(size = 3, alpha = 0.7) + labs(title = "Scatter Plot of Petal Length vs Petal Width", x = "Petal Length", y = "Petal Width") + scale_color_manual(values = c("setosa" = "blue", "versicolor" = "green", "virginica" = "red")) + scale_shape_manual(values = c("setosa" = 16, "versicolor" = 17, "virginica" = 18))
print(scatter_plot)
```

8. Write an R program to create Histogram and Box plots using ggplot2 package in R.

```
# Load required libraries
library(ggplot2)
# Histogram
# Sample data for Histogram
data = data.frame(values = rnorm(1000))
# Create a histogram
ggplot(data, aes(x = values)) +
geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
labs(title = "Histogram", x = "Values", y = "Frequency") + theme_minimal()
library(ggplot2)
#Box plot
# Sample data for box plot
data = data.frame(group = rep(c("A", "B", "C"), each = 50), value = rnorm(150))
# Create a box plot
ggplot(data, aes(x = group, y = value, fill = group)) +
geom_boxplot() + labs(title = "Box Plot", x = "Group", y = "Value") +
theme_minimal()
```

9. Using the in-build mtcars dataset and lattice package, write an R program to create Bar plot, Scatter plot, Histogram and Density plot.

```
# Load required libraries
library(lattice)
# Create a bar plot of average MPG by number of cylinders
avg_mpg_by_cyl = tapply(mtcars$mpg, mtcars$cyl, mean)
bar_plot = barchart(avg_mpg_by_cyl, main = "Average MPG by Number of Cylinders",
xlab = "Cylinders", ylab = "Average MPG", col = "orange")
print(bar_plot)
# Create a scatter plot of MPG vs Horsepower
scatter_plot = xyplot(mpg ~ hp, data = mtcars, pch = 16, col = "blue", main = "Scatter Plot of MPG vs. Horsepower", xlab = "Horsepower", ylab = "MPG")
```

```
print(scatter_plot)
# Create a histogram of MPG values
histogram_plot = histogram (~ mpg, data = mtcars, main = "Histogram of MPG", xlab = "MPG", ylab = "Frequency", col = "green")
print(histogram_plot)
# Create a density plot of MPG values
density_plot <- densityplot(~ mpg, data = mtcars, main = "Density Plot of MPG", xlab = "MPG", ylab = "Density", col = "purple")
print(density_plot)
```

10. Write an R program to create 3D Wireframe Plot and Level Plot using lattice package in R.

```
# Load the lattice package for advanced visualizations
library(lattice)
# Create numeric vectors 'a' and 'b'
a = 1:10
b = 1:15
# Generate a data frame with all combinations of 'a' and 'b'
eg =- expand.grid(x=a, y=b)
# Calculate a new variable 'z' based on the formula
eg$z = eg$x^2 + eg$x * eg$y
# Create a 3D wireframe plot to visualize 'z' vs 'x' and 'y'
wireframe(z ~ x+y, eg)
#Level plot
x = seq(-pi, pi, length.out = 100)
y = seq(-pi, pi, length.out = 100)
z = outer(x, y, function(x, y) sin(sqrt(x^2 + y^2)))
levelplot(z, xlab = "x", ylab = "y", main = "2D Sin Function")
```