PART-A

Program 1 (.csv mean, median)

```
# Importing the example data
examples <- read.csv("D:/example.csv")

# Taking the mean
mean_value <- mean(examples[, 1])
cat("Mean =", mean_value)

# Taking the median
median_value <- median(examples[, 1])
cat("Median =", median_value)

# Taking the variance
variance_value <- var(examples[, 1])
cat("Variance =", variance_value)

# Taking the standard deviation
sd_value <- variance_value^0.5
cat("Standard Deviation =", sd_value)

# Taking the range
range_value <- range(examples[, 1])
cat("Range =", range_value)

# Taking the quartile
quartile_1 <- quantile(examples[, 1], 0.25)
cat("1st Quartile =",quartile_1)

quartile_2 <- quantile(examples[, 1], 0.5)
cat("2nd Quartile =",quartile_2)

quartile_3 <- quantile(examples[, 1], 0.75)
cat("3rd Quartile =",quartile_3)
```

Program 2 (file operations)

```
# Create a file
file.create("D:/AEC_File1.txt")
# Writing into a file
write.table(x = iris[1:10, ], file = "AEC_File1.txt")
```

```
# Reading a text file
myData = read.table(file = "AEC_File1.txt ")
print(myData)
# Renaming a file
file.rename("AEC_File1.txt", "AEC_renamed.txt")
# Listing the table
list.files()
# Copy a file
file.copy("D:/AEC_File1.txt", "D:/programs")
list.files("D:/programs")
```

Program 3 (string operations, reverse, palindrome)

```
# Create two strings
str1 = "Artificial"
str2 = "Intelligence"
# using paste() to concatenate two strings
result = paste(str1, str2)
print(result)
# Compare both strings
result = toupper(str1) == toupper(str2)
print(result)
# Reverse the string
text = "Maths is fun"
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 = "mom"
# 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")
}
```

Program 4 (string manipulation)

```
string = "Hello World"
# nchar function
nchar(string)
# toupper function
```

```
toupper(string)
# tolower function
tolower(string)
# substr function
substr(string, 5, 20)
# grep function
grep("wor", string)
# paste function
paste("hello", "world", string, sep = "-")
# strsplit function
strsplit(string,'e')
# sprintf function
age = 40L
name = "Ram"
sprintf("%s is %d years old", name, age)
# cat function
cat("hello","world",sep = "-")
# sub function
sub("World", "there!", string)
```

PART-B

Program 5 (scatter, line, bar, histo)

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

Program 6 (2D, 3D pie chart)

```
marks<-c(100,65,89,90,85)
subjects<-c("Math","DAA","R","OS","ARM")

percentage<-round(100 * marks/sum(marks),1)

pie(marks, labels=marks, main="Marks scored by a student", col=rainbow(length(marks)))
legend("topleft",subjects,cex=1,fill =rainbow(length(marks)))

library(plotrix)

pie3D(marks, labels=marks, main="Marks scored by a person",
col=rainbow(length(marks)),explode=0.2)
legend("bottom",subjects,cex=1,fill =rainbow(length(marks)))
```

Program 7 (iris dataset)

```
# 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)
```

Program 8 (histo and box using ggplot2)

```
# 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()
```

Program 9 (mtcars and lattice)

```
# 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)
```

Program 10 (3D wireframe)

```
# 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")
```