```
#1
print("Hello, Welcome to R Programming")

# Assign variables
x <- 25
y <- 5

# Arithmetic operations
sum_val <- x + y
diff_val <- x - y
prod_val <- x * y
div_val <- x / y

sum_val
diff_val
prod_val
div_val

# Relational operations
greater_check <- x > y
equal_check <- x == y

# Load and explore iris dataset
data("iris")     # load dataset
head(iris)       # first 6 rows
str(iris)        # structure of dataset
summary(iris)    # descriptive statistics
```

```
#Performed By Niranjan Muley(A-18)2
library(dplyr)

# 1. Import inbuilt dataset: iris
data("iris")
head(iris)
str(iris)
summary(iris)
```

```
# 2. Import another dataset: mtcars
data("mtcars")
head(mtcars)
str(mtcars)

# 3. Titanic dataset
data("Titanic")
Titanic

# 4. Check for missing values
sum(is.na(iris))         # total missing values
colSums(is.na(mtcars))    # missing values per column

# 5. Summarize distribution by group
iris %>%
  group_by(Species) %>%
  summarise(
    Avg_Sepal_Length = mean(Sepal.Length),
    Avg_Petal_Length = mean(Petal.Length)
  )
```

```
#Performed By Niranjan Muley(A-18)3
library(dplyr)

# Load dataset
data("airquality")
head(airquality)

# 1. Check for missing values
sum(is.na(airquality))
colSums(is.na(airquality))

# 2. Handle missing values (mean, median)
airquality$Ozone[is.na(airquality$Ozone)] <- mean(airquality$Ozone, na.rm = TRUE)
airquality$Solar.R[is.na(airquality$Solar.R)] <- median(airquality$Solar.R, na.rm = TRUE)
```

```
# 3. Remove duplicates (iris example)
data("iris")
iris_with_duplicates <- rbind(iris, iris[1:5, ])
nrow(iris_with_duplicates)

iris_clean <- distinct(iris_with_duplicates)
nrow(iris_clean)

# 4. Standardize formats
iris_clean$Species <- tolower(as.character(iris_clean$Species))
iris_clean$Species <- as.factor(iris_clean$Species)

# View cleaned dataset
head(iris_clean)
```

```
#Performed By Niranjan Muley(A-18)4
install.packages("modeest")
library(modeest)
library(dplyr)
library(ggplot2)
library(modeest)

# Load dataset
data("iris")

# Descriptive statistics
mean(iris$Sepal.Length)
median(iris$Sepal.Length)
mlv(iris$Sepal.Length, method = "mfv")   # mode
sd(iris$Sepal.Length)
range(iris$Sepal.Length)
summary(iris$Sepal.Length)

# Histogram
hist(iris$Sepal.Length,
    main = "Histogram of Sepal Length",
    xlab = "Sepal Length",
    col = "lightblue",
    border = "black")

# Scatterplot
plot(iris$Sepal.Length, iris$Petal.Length,
    main = "Scatterplot of Sepal vs Petal Length",
    xlab = "Sepal Length", ylab = "Petal Length",
    col = "blue", pch = 19)

# Boxplot
boxplot(Sepal.Length ~ Species, data = iris,
      main = "Boxplot of Sepal Length by Species",
      xlab = "Species", ylab = "Sepal Length",
      col = c("lightgreen","lightblue","pink"))
```

```
#Performed By Niranjan Muley(A-18)5
# Install required packages (run only once)
install.packages("factoextra")
install.packages("ggplot2")
install.packages("dplyr")

# Load libraries
library(dplyr)
library(ggplot2)
library(factoextra)

# Load dataset
data("iris")
iris_data <- iris[, 1:4]   # only numeric features

# 1. Elbow Method to find optimal K
fviz_nbclust(iris_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal K")

# 2. Apply K-means clustering with K = 3
set.seed(123)
kmeans_model <- kmeans(iris_data, centers = 3, nstart = 20)
```

```
# 3. Cluster assignments
kmeans_model$cluster[1:10]
table(kmeans_model$cluster, iris$Species)

# 4. Visualize clusters
fviz_cluster(kmeans_model,
             data = iris_data,
             ellipse.type = "norm",
             palette = "jco",
             ggtheme = theme_minimal()
)
```

```
#Performed By Niranjan Muley(A-18)6
library(dplyr)
library(ggplot2)
library(GGally)
library(ggcorrplot)

data("iris")
head(iris)

# 1. Summary
summary(iris)

# 2. Histogram
ggplot(iris, aes(x = Sepal.Length)) +
  geom_histogram(bins = 20, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Sepal Length", x = "Sepal Length", y = "Frequency")

# 3. Scatterplot
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, color = Species)) +
  geom_point(size = 3) +
  labs(title = "Sepal Length vs Petal Length")

# 4. Boxplot
ggplot(iris, aes(x = Species, y = Sepal.Width, fill = Species)) +
  geom_boxplot() +
  labs(title = "Boxplot of Sepal Width by Species")

# 5. Correlation matrix
corr_matrix <- cor(iris[,1:4])
corr_matrix

ggcorrplot(corr_matrix, lab = TRUE, title = "Correlation Matrix Heatmap")

# 6. Pair plot
ggpairs(iris[,1:4])
```

```
#Performed By Niranjan Muley(A-18)7
library(dplyr)
data("iris")

# 1. t-test (setosa vs versicolor)
t_test_result <- t.test(Sepal.Length ~ Species,
                        data = iris %>% filter(Species %in% c("setosa","versicolor")))
t_test_result

# 2. ANOVA
anova_model <- aov(Sepal.Length ~ Species, data = iris)
summary(anova_model)

# 3. Correlation
correlation <- cor(iris$Sepal.Length, iris$Petal.Length)
correlation

# 4. Correlation test
cor_test <- cor.test(iris$Sepal.Length, iris$Petal.Length)
cor_test
```

```
#Performed By Niranjan Muley(A-18)8
library(dplyr)
library(ggplot2)

data("mtcars")
```

```
head(mtcars)

# 1. Simple regression
model_simple <- lm(mpg ~ wt, data = mtcars)
summary(model_simple)

# 2. Plot regression
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(title = "Simple Linear Regression: MPG vs Weight",
       x = "Weight (1000 lbs)", y = "Miles per Gallon")

# 3. Multiple regression
model_multiple <- lm(mpg ~ wt + hp, data = mtcars)
summary(model_multiple)

# 4. Residual diagnostics
plot(model_multiple, which = 1)   # Residuals vs Fitted
plot(model_multiple, which = 2)   # Q–Q plot
```

```
#Performed By Niranjan Muley(A-18)9
install.packages("caret")
install.packages("rpart")
install.packages("rpart.plot")
install.packages("pROC")

library(dplyr)
library(caret)
library(rpart)
library(rpart.plot)
library(pROC)
library(ggplot2)

data("iris")

# Convert to binary: setosa vs non_setosa
```

```
iris_bin <- iris %>%
  mutate(Species = factor(ifelse(Species == "setosa", "setosa", "non_setosa"),
                         levels = c("non_setosa","setosa")))

# Train-test split
set.seed(123)
idx <- createDataPartition(iris_bin$Species, p = 0.7, list = FALSE)

trainData <- iris_bin[idx, ]
testData  <- iris_bin[-idx, ]

# 1. Logistic Regression
log_model <- glm(Species ~ Sepal.Length + Petal.Length,
                 data = trainData, family = binomial())

log_prob <- predict(log_model, testData, type = "response")

log_class <- ifelse(log_prob > 0.5, "setosa", "non_setosa")

confusionMatrix(factor(log_class), testData$Species)

# ROC Curve
roc_obj <- roc(testData$Species, log_prob, levels = c("non_setosa","setosa"))
plot(roc_obj, col="blue", main="ROC Curve - Logistic Regression")

# 2. Decision Tree
tree_model <- rpart(Species ~ Sepal.Length + Petal.Length,
                    data = trainData, method="class")

rpart.plot(tree_model)

# Confusion matrix for tree model
tree_pred <- predict(tree_model, testData, type = "class")

confusionMatrix(tree_pred, testData$Species)
```

```
library(dplyr)
library(ggplot2)
library(factoextra)

data("iris")
iris_data <- iris[,1:4]

# 1. Elbow method
fviz_nbclust(iris_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal K")

# 2. K-means clustering K=3
set.seed(123)
kmeans_model <- kmeans(iris_data, centers = 3, nstart = 20)

# 3. Cluster assignments
kmeans_model$cluster[1:10]
table(kmeans_model$cluster, iris$Species)

# 4. Visualize clusters
fviz_cluster(kmeans_model, data = iris_data,
             ellipse.type = "norm",
             palette = "jco",
             ggtheme = theme_minimal())
```