Dr. SNS RAJALAKSHMI COLLEGE OF ARTS & SCIENCE (AUTONOMOUS)

Affiliated to Bharathiar University
Accredited with “A+” Grade by NAAC (Cycle – III)
Coimbatore-641049

DEPARTMENT OF COMPUTER APPLICATIONS

III BCA

21UCA558 : DATA ANALYTICS USING R

PRACTICAL RECORD

NAME :

REG. No:

Dr. SNS RAJALAKSHMI COLLEGE OF ARTS & SCIENCE
(AUTONOMOUS)

Affiliated to Bharathiar University
Accredited with “A+” Grade by NAAC (Cycle – III)
Coimbatore-641049

DEPARTMENT OF COMPUTER APPLICATIONS

III BCA

21UCA558 : DATA ANALYTICS USING R

Certified that this Bonafide record work done by

NAME :

REG.NO :

BATCH : 2021 -2024

STAFF INCHARGE **HOD**

Submitted for BCA Degree VI Semester Practical Examination on______________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

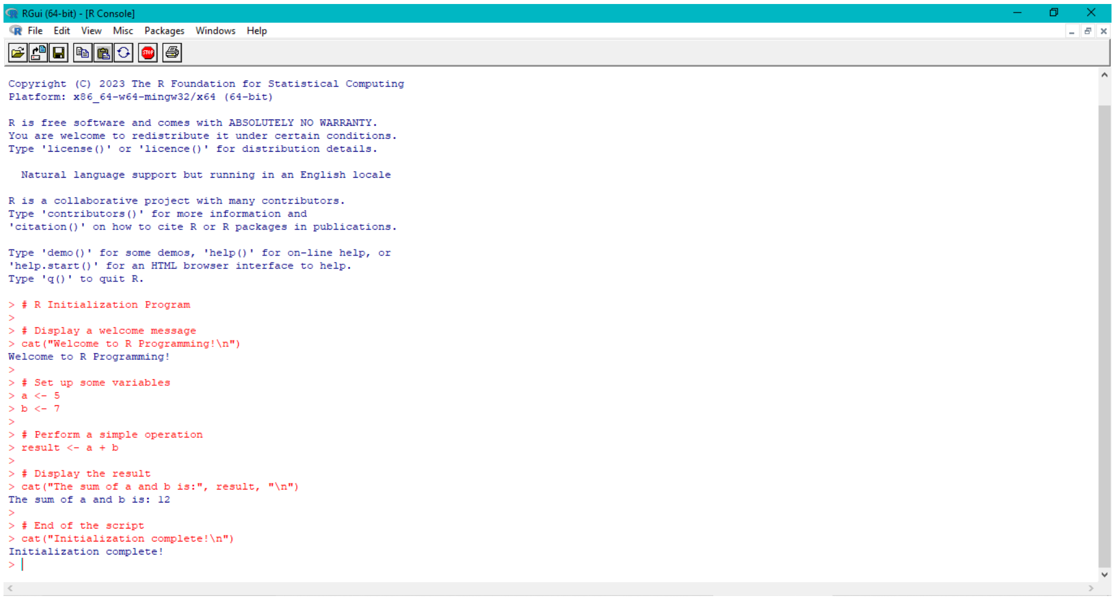
INDEX

Ex.No:1	**INITIALIZATION OF R PROGRAMMING**
Date:	

SOURCE CODE

```
# R Initialization Program

# Display a welcome message
cat("Welcome to R Programming!\n")

# Set up some variables
a <- 5
b <- 7

 # Perform a simple operation
 result <- a + b

 # Display the result
cat("The sum of a and b is:", result, "\n")

# End of the script
cat("Initialization complete!\n")
```

OUTPUT



RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

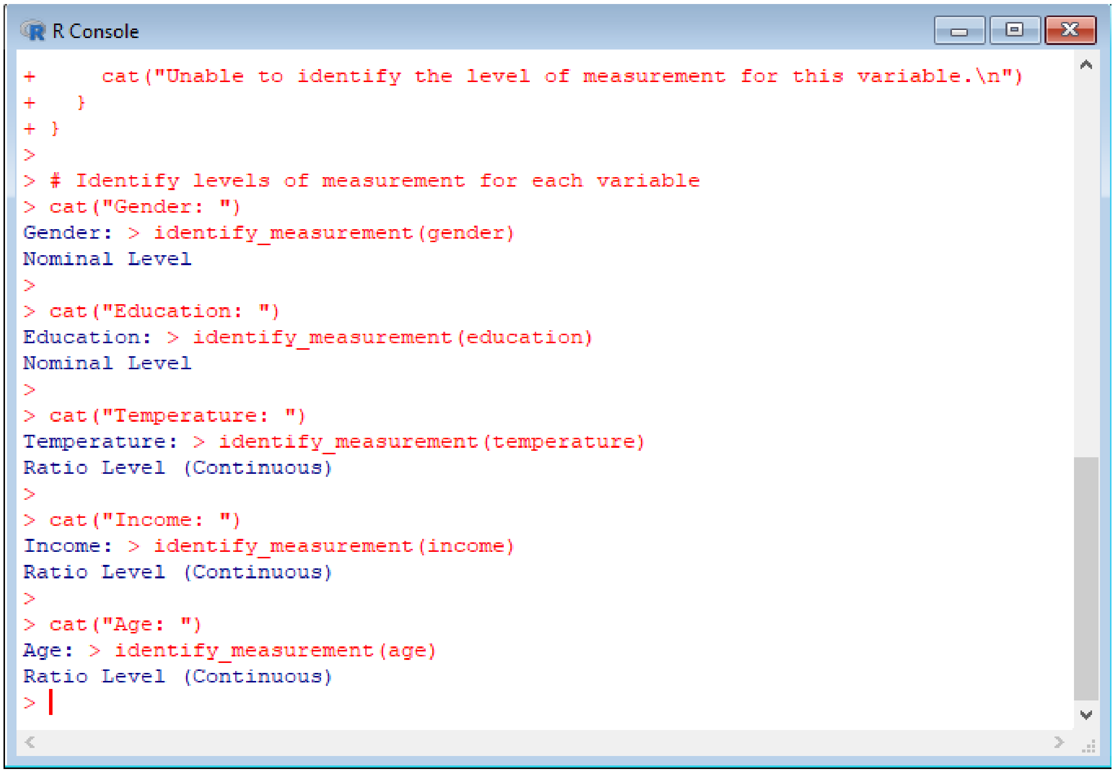
```
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> # R Initialization Program
>
> # Display a welcome message
> cat("Welcome to R Programming!\n")
Welcome to R Programming!
>
> # Set up some variables
> a <- 5
> b <- 7
>
> # Perform a simple operation
> result <- a + b
>
> # Display the result
> cat("The sum of a and b is:", result, "\n")
The sum of a and b is: 12
>
> # End of the script
> cat("Initialization complete!\n")
Initialization complete!
> |
```

EX.NO :2	**IDENTIFYING TYPES OF VARIABLES : LEVELS OF MEASUREMENTS**
Date:	

SOURCE CODE

```
# Sample data
gender <- c("Male", "Female", "Male", "Male", "Female")
education <- c("High School", "Bachelor's", "Master's", "High School", "PhD")
temperature <- c(23.5, 28.2, 25.0, 21.8, 30.5)
income <- c(45000, 60000, 75000, 55000, 90000)
age <- c(28, 35, 42, 24, 50)

# Function to identify level of measurement
identify_measurement <- function(variable) {
  if (is.factor(variable) || is.character(variable)) {
    cat("Nominal Level\n")
  } else if (is.ordered(variable)) {
    cat("Ordinal Level\n")
  } else if (is.numeric(variable)) {
    if (is.integer(variable)) {
      cat("Ratio Level (Integer)\n")
    } else {
      cat("Ratio Level (Continuous)\n")
    }
  } else {
    cat("Unable to identify the level of measurement for this variable.\n")
  }
}

# Identify levels of measurement for each variable
cat("Gender: ")
identify_measurement(gender)

cat("Education: ")
identify_measurement(education)

cat("Temperature: ")
identify_measurement(temperature)

cat("Income: ")
identify_measurement(income)

cat("Age: ")
identify_measurement(age)
```

OUTPUT



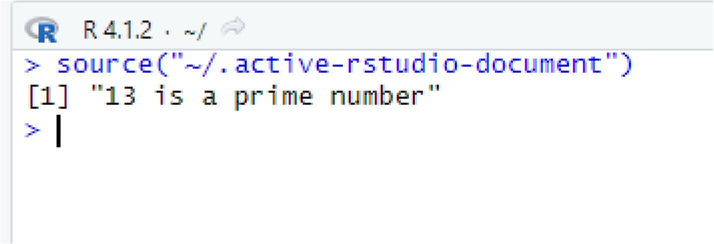
```
+     cat("Unable to identify the level of measurement for this variable.\n")
+   }
+ }
>
> # Identify levels of measurement for each variable
> cat("Gender: ")
Gender: > identify_measurement(gender)
Nominal Level
>
> cat("Education: ")
Education: > identify_measurement(education)
Nominal Level
>
> cat("Temperature: ")
Temperature: > identify_measurement(temperature)
Ratio Level (Continuous)
>
> cat("Income: ")
Income: > identify_measurement(income)
Ratio Level (Continuous)
>
> cat("Age: ")
Age: > identify_measurement(age)
Ratio Level (Continuous)
> |
```

Ex.No:3	**FIND THE PRIME NUMBER USING R PROGRAMMING**
Date:	

SOURCE CODE

```
  Find_Prime_No<-function(n1)
{
 if(n1==2)
 {
  return(TRUE)
 }
 if(n1<=1)
 {
  return(FALSE)
 }
 for(i in 2:(n1-1))
 {
  if(n1%%i==0)
  {
   return(FALSE)
  }
 }
 return(TRUE)
}
numb_1<-13
if(Find_Prime_No(numb_1))
{
 print(paste(numb_1,"is a prime number"))
}else
{
 print("It is not a prime number")
}
```

OUTPUT



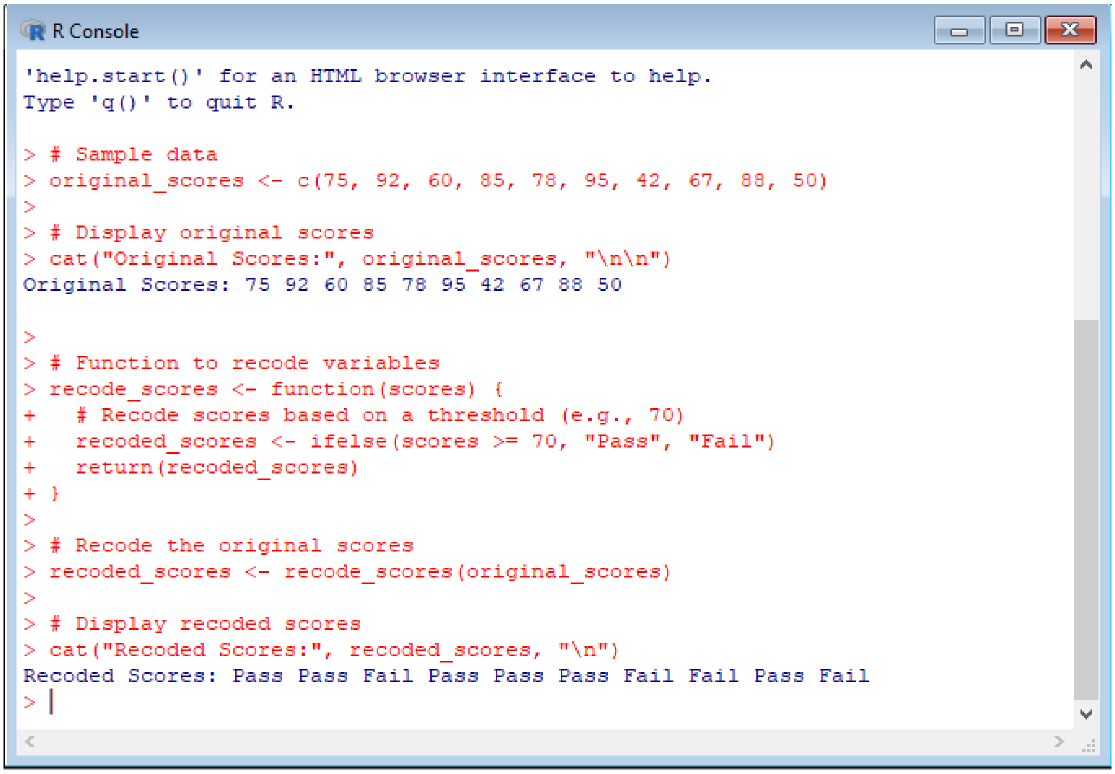
```
> source("~/.active-rstudio-document")
[1] "13 is a prime number"
> 
```

Ex.No:4	**INTRODUCTION TO PROBABILITY - RECODING VARIABLES**
Date:	

SOURCE CODE

```
# Sample data
original_scores <- c(75, 92, 60, 85, 78, 95, 42, 67, 88, 50)

# Display original scores
cat("Original Scores:", original_scores, "\n\n")

# Function to recode variables
recode_scores <- function(scores) {
  # Recode scores based on a threshold (e.g., 70)
  recoded_scores <- ifelse(scores >= 70, "Pass", "Fail")
  return(recoded_scores)
}

# Recode the original scores
recoded_scores <- recode_scores(original_scores)

# Display recoded scores
cat("Recoded Scores:", recoded_scores, "\n")
```

OUTPUT



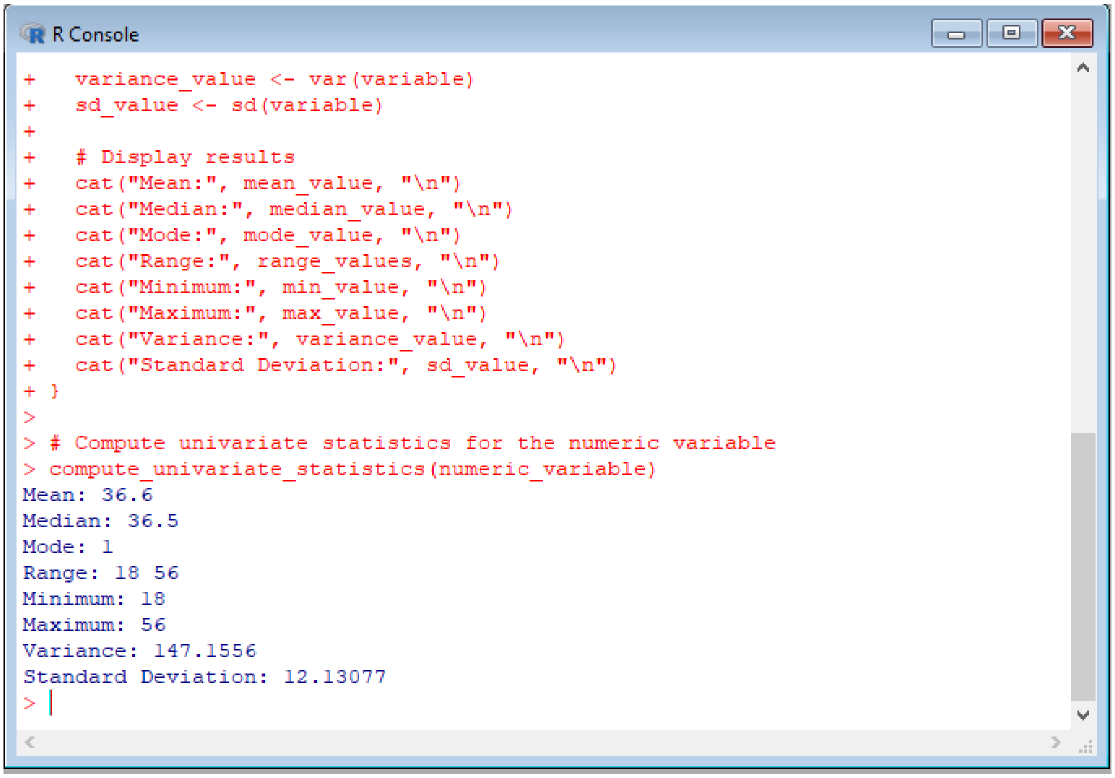
```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> # Sample data
> original_scores <- c(75, 92, 60, 85, 78, 95, 42, 67, 88, 50)
>
> # Display original scores
> cat("Original Scores:", original_scores, "\n\n")
Original Scores: 75 92 60 85 78 95 42 67 88 50

>
> # Function to recode variables
> recode_scores <- function(scores) {
+   # Recode scores based on a threshold (e.g., 70)
+   recoded_scores <- ifelse(scores >= 70, "Pass", "Fail")
+   return(recoded_scores)
+ }
>
> # Recode the original scores
> recoded_scores <- recode_scores(original_scores)
>
> # Display recoded scores
> cat("Recoded Scores:", recoded_scores, "\n")
Recoded Scores: Pass Pass Fail Pass Pass Pass Fail Fail Pass Fail
> 
```

Ex.No:5	**UNIVARIATE STATISTICS**
Date:	

SOURCE CODE

```
# Sample data
numeric_variable <- c(23, 45, 32, 18, 56, 40, 27, 33, 50, 42)

# Function to compute univariate statistics
compute_univariate_statistics <- function(variable) {
  # Measures of central tendency
  mean_value <- mean(variable)
  median_value <- median(variable)
  mode_value <- table(variable)[which.max(table(variable))]

  # Measures of dispersion
  range_values <- range(variable)
  min_value <- min(variable)
  max_value <- max(variable)
  variance_value <- var(variable)
  sd_value <- sd(variable)

  # Display results
  cat("Mean:", mean_value, "\n")
  cat("Median:", median_value, "\n")
  cat("Mode:", mode_value, "\n")
  cat("Range:", range_values, "\n")
  cat("Minimum:", min_value, "\n")
  cat("Maximum:", max_value, "\n")
  cat("Variance:", variance_value, "\n")
  cat("Standard Deviation:", sd_value, "\n")
}

# Compute univariate statistics for the numeric variable
compute_univariate_statistics(numeric_variable)
```

OUTPUT



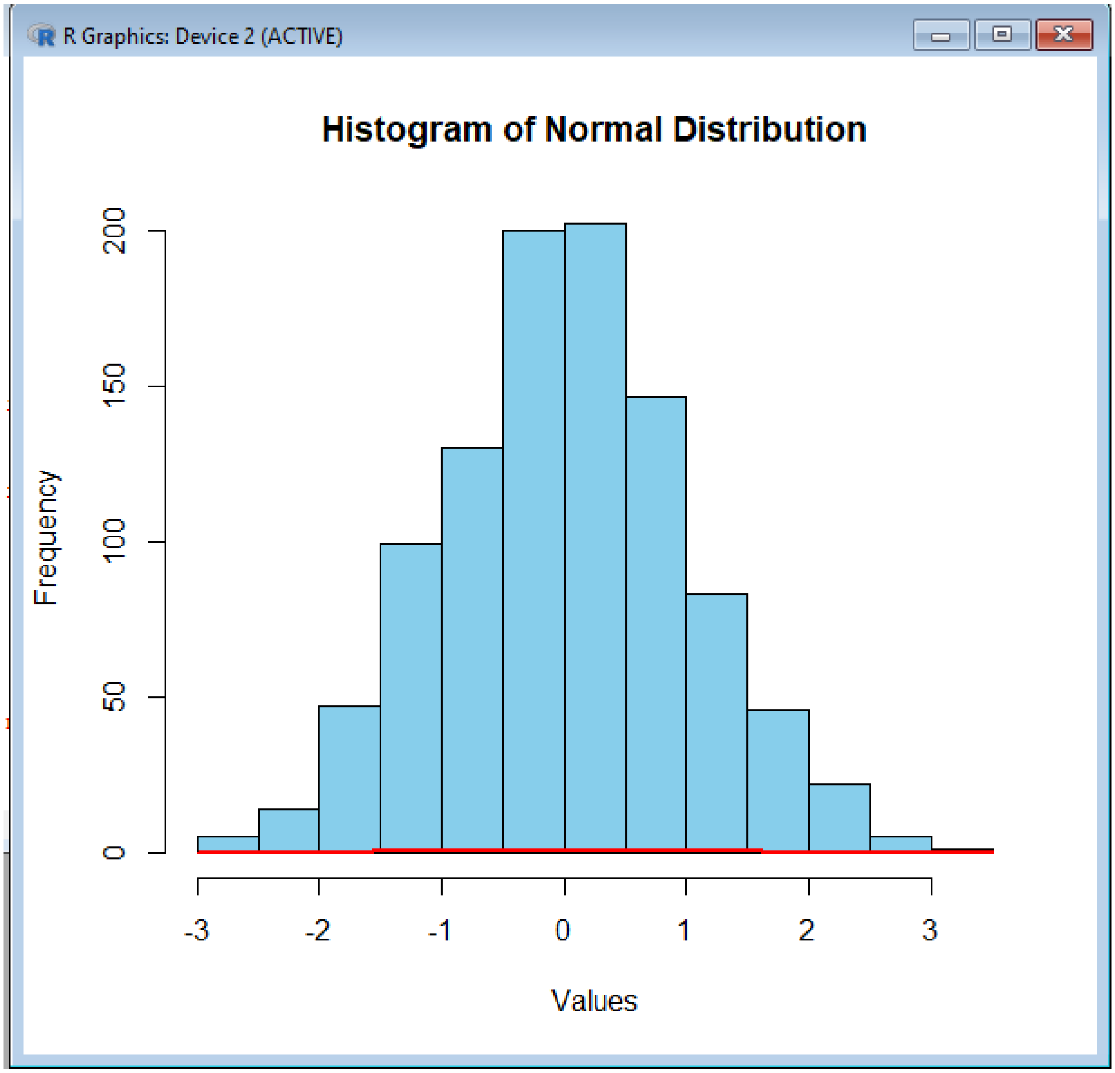
```
+   variance_value <- var(variable)
+   sd_value <- sd(variable)
+
+   # Display results
+   cat("Mean:", mean_value, "\n")
+   cat("Median:", median_value, "\n")
+   cat("Mode:", mode_value, "\n")
+   cat("Range:", range_values, "\n")
+   cat("Minimum:", min_value, "\n")
+   cat("Maximum:", max_value, "\n")
+   cat("Variance:", variance_value, "\n")
+   cat("Standard Deviation:", sd_value, "\n")
+ }
>
> # Compute univariate statistics for the numeric variable
> compute_univariate_statistics(numeric_variable)
Mean: 36.6
Median: 36.5
Mode: 1
Range: 18 56
Minimum: 18
Maximum: 56
Variance: 147.1556
Standard Deviation: 12.13077
> |
```

Ex.No: 6	**THE NORMAL CURVE - CREATING A HISTOGRAM IN R**
Date:	

SOURCE CODE

```
# Set seed for reproducibility
set.seed(123)

# Generate random data from a normal distribution
data <- rnorm(1000, mean = 0, sd = 1)

# Create a histogram
hist(data,
     col = "skyblue",            # Color of the bars
     main = "Histogram of Normal Distribution",  # Main title
     xlab = "Values",            # X-axis label
     ylab = "Frequency",         # Y-axis label
     border = "black",           # Color of the bar borders
     breaks = 20                 # Number of bins
)

# Overlay a normal curve on the histogram
curve(dnorm(x, mean = mean(data), sd = sd(data)),
      col = "red",               # Color of the curve
      lwd = 2,                   # Line width
      add = TRUE                 # Overlay on the existing plot
)
```

OUTPUT

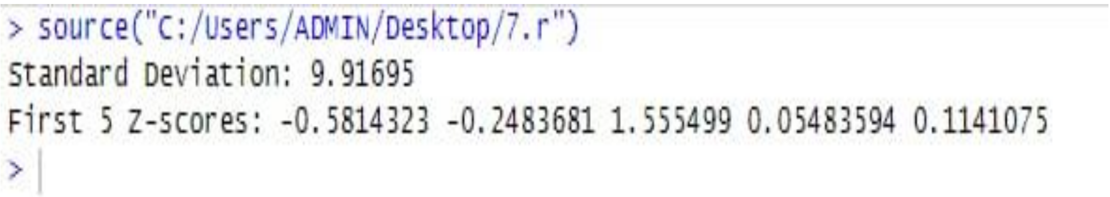

R Graphics: Device 2 (ACTIVE)
Histogram of Normal Distribution
Frequency
0
50
100
150
200
-3
-2
-1
0
1
2
3
Values

Ex.No: 7	**STANDARD DEVIATIONS , STANDARD SCORES AND THE NORMAL DISTRIBUTION**
Date:	

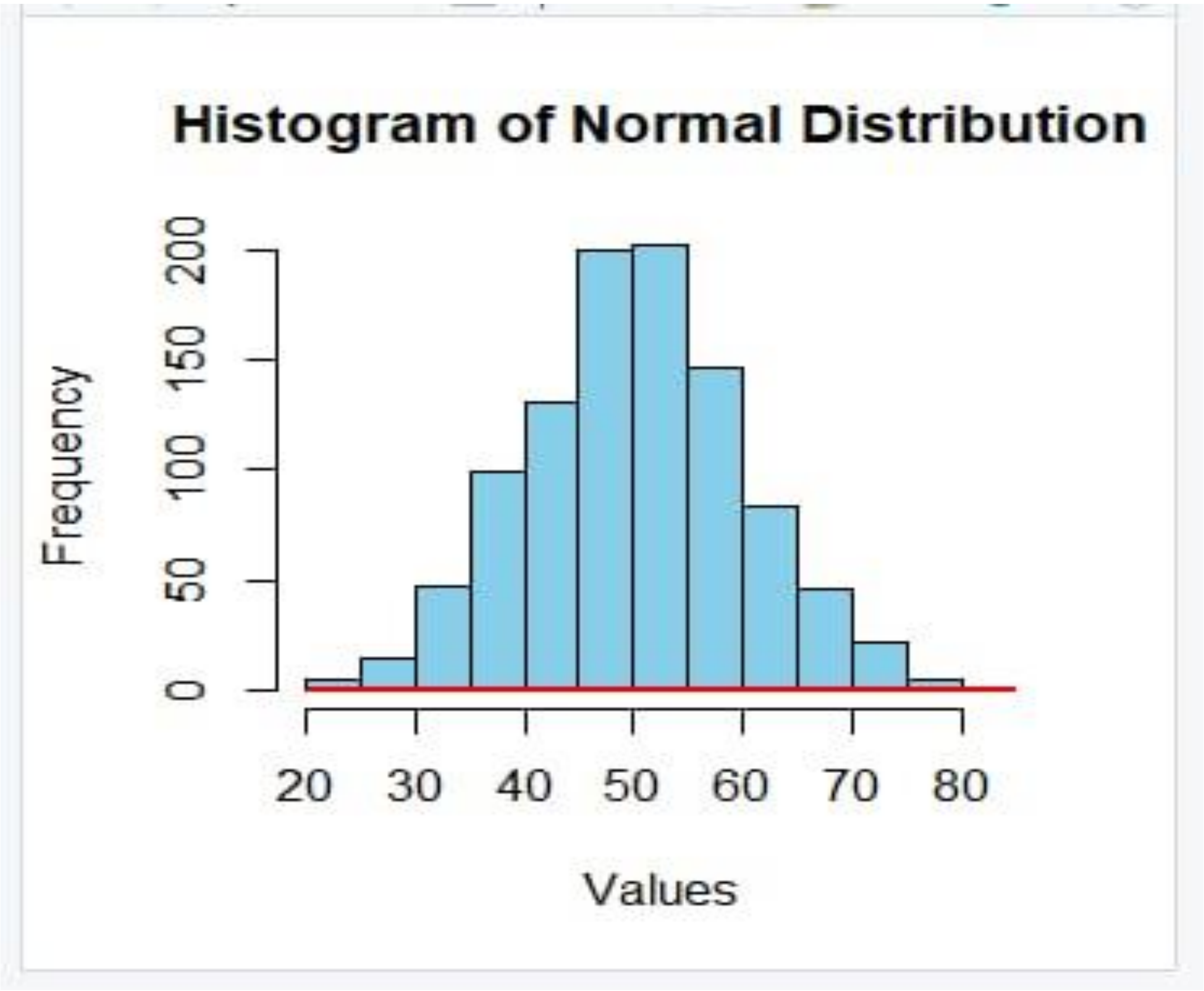
SOURCE CODE

```
# Set seed for reproducibility
set.seed(123)

# Generate random data from a normal distribution
data <- rnorm(1000, mean = 50, sd = 10)

# Calculate standard deviation
std_dev <- sd(data)
cat("Standard Deviation:", std_dev, "\n")

# Calculate standard scores (z-scores)
z_scores <- scale(data)
cat("First 5 Z-scores:", head(z_scores, 5), "\n")

# Create a histogram
hist(data,
     col = "skyblue",            # Color of the bars
     main = "Histogram of Normal Distribution",  # Main title
     xlab = "Values",            # X-axis label
     ylab = "Frequency",         # Y-axis label
     border = "black",           # Color of the bar borders
     breaks = 20                 # Number of bins
)

# Overlay a standard normal curve on the histogram
curve(dnorm(x, mean = mean(data), sd = std_dev),
      col = "red",               # Color of the curve
      lwd = 2,                   # Line width
      add = TRUE                 # Overlay on the existing plot
)
```

OUTPUT



```
> source("C:/Users/ADMIN/Desktop/7.r")
Standard Deviation: 9.91695
First 5 Z-scores: -0.5814323 -0.2483681 1.555499 0.05483594 0.1141075
> 
```

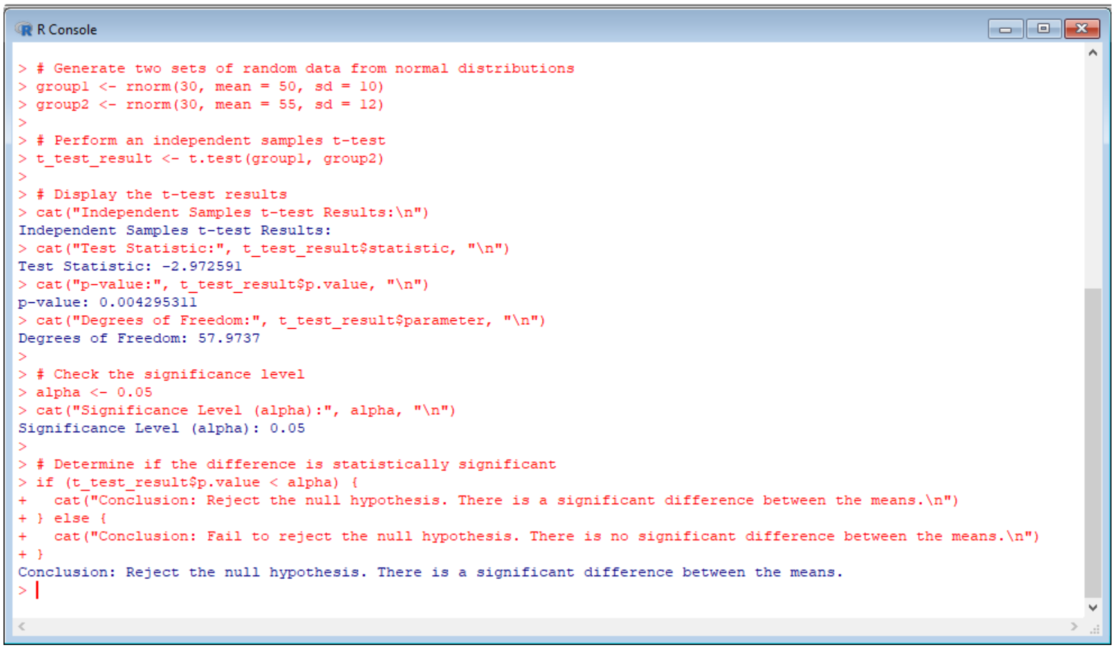


Ex.No: 8	**TESTING THE SIGNIFICANCE OF THE DIFFERENCE BETWEEN TWO MEANS**
Date:	

SOURCE CODE

```
Set seed for reproducibility
set.seed(123)

# Generate two sets of random data from normal distributions
group1 <- rnorm(30, mean = 50, sd = 10)
group2 <- rnorm(30, mean = 55, sd = 12)

# Perform an independent samples t-test
t_test_result <- t.test(group1, group2)

# Display the t-test results
cat("Independent Samples t-test Results:\n")
cat("Test Statistic:", t_test_result$statistic, "\n")
cat("p-value:", t_test_result$p.value, "\n")
cat("Degrees of Freedom:", t_test_result$parameter, "\n")

# Check the significance level
alpha <- 0.05
cat("Significance Level (alpha):", alpha, "\n")

# Determine if the difference is statistically significant
if (t_test_result$p.value < alpha) {
  cat("Conclusion: Reject the null hypothesis. There is a significant difference between the means.\n")
} else {
  cat("Conclusion: Fail to reject the null hypothesis. There is no significant difference between the means.\n")
}
```

OUTPUT



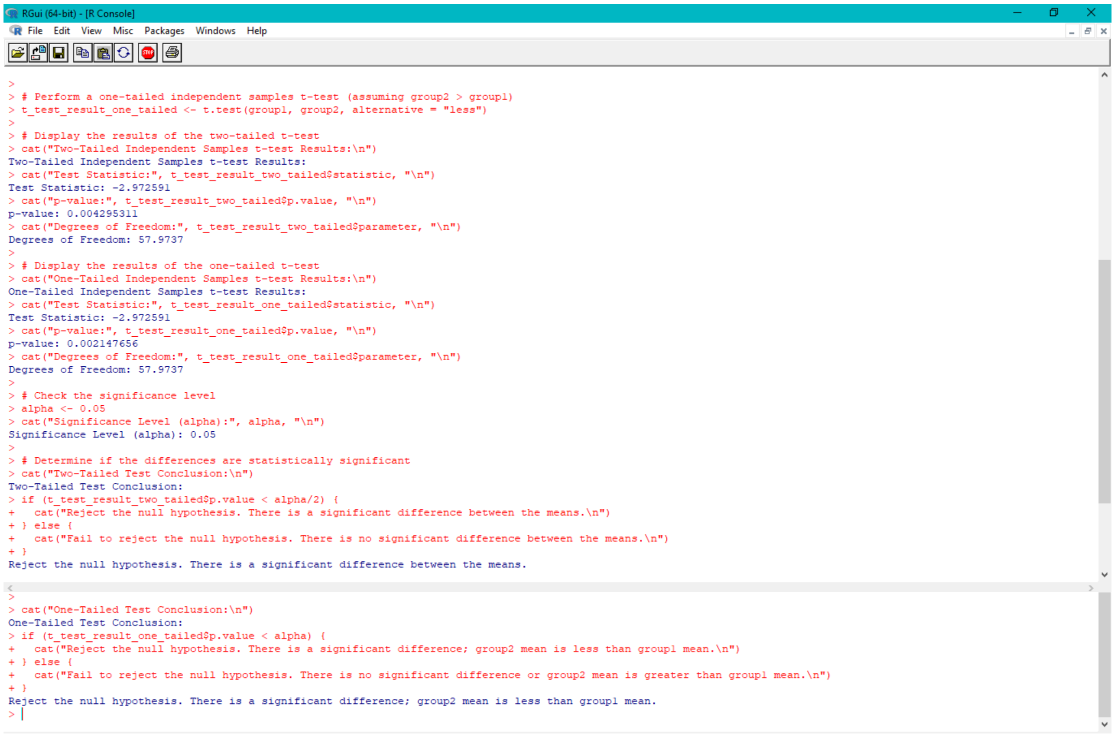
```
> # Generate two sets of random data from normal distributions
> group1 <- rnorm(30, mean = 50, sd = 10)
> group2 <- rnorm(30, mean = 55, sd = 12)
>
> # Perform an independent samples t-test
> t_test_result <- t.test(group1, group2)
>
> # Display the t-test results
> cat("Independent Samples t-test Results:\n")
Independent Samples t-test Results:
> cat("Test Statistic:", t_test_result$statistic, "\n")
Test Statistic: -2.972591
> cat("p-value:", t_test_result$p.value, "\n")
p-value: 0.004295311
> cat("Degrees of Freedom:", t_test_result$parameter, "\n")
Degrees of Freedom: 57.9737
>
> # Check the significance level
> alpha <- 0.05
> cat("Significance Level (alpha):", alpha, "\n")
Significance Level (alpha): 0.05
>
> # Determine if the difference is statistically significant
> if (t_test_result$p.value < alpha) {
+   cat("Conclusion: Reject the null hypothesis. There is a significant difference between the means.\n")
+ } else {
+   cat("Conclusion: Fail to reject the null hypothesis. There is no significant difference between the means.\n")
+ }
Conclusion: Reject the null hypothesis. There is a significant difference between the means.
> |
```

Ex.No: 9	**ONE AND TWO TAILED TESTS**
Date:	

SOURCE CODE

```
# Set seed for reproducibility
set.seed(123)

# Generate two sets of random data from normal distributions
group1 <- rnorm(30, mean = 50, sd = 10)
group2 <- rnorm(30, mean = 55, sd = 12)

# Perform a two-tailed independent samples t-test
t_test_result_two_tailed <- t.test(group1, group2)

# Perform a one-tailed independent samples t-test (assuming group2 > group1)
t_test_result_one_tailed <- t.test(group1, group2, alternative = "less")

# Display the results of the two-tailed t-test
cat("Two-Tailed Independent Samples t-test Results:\n")
cat("Test Statistic:", t_test_result_two_tailed$statistic, "\n")
cat("p-value:", t_test_result_two_tailed$p.value, "\n")
cat("Degrees of Freedom:", t_test_result_two_tailed$parameter, "\n")

# Display the results of the one-tailed t-test
cat("One-Tailed Independent Samples t-test Results:\n")
cat("Test Statistic:", t_test_result_one_tailed$statistic, "\n")
cat("p-value:", t_test_result_one_tailed$p.value, "\n")
cat("Degrees of Freedom:", t_test_result_one_tailed$parameter, "\n")

# Check the significance level
alpha <- 0.05
cat("Significance Level (alpha):", alpha, "\n")

# Determine if the differences are statistically significant
cat("Two-Tailed Test Conclusion:\n")
if (t_test_result_two_tailed$p.value < alpha/2) {
  cat("Reject the null hypothesis. There is a significant difference between the means.\n")
} else {
  cat("Fail to reject the null hypothesis. There is no significant difference between the means.\n")
}

cat("One-Tailed Test Conclusion:\n")
if (t_test_result_one_tailed$p.value < alpha) {
  cat("Reject the null hypothesis. There is a significant difference; group2 mean is less than group1
mean.\n")
} else {
  cat("Fail to reject the null hypothesis. There is no significant difference or group2 mean is greater than
group1 mean.\n")
}
```

OUTPUT



RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
>
> # Perform a one-tailed independent samples t-test (assuming group2 > group1)
> t_test_result_one_tailed <- t.test(group1, group2, alternative = "less")
>
> # Display the results of the two-tailed t-test
> cat("Two-Tailed Independent Samples t-test Results:\n")
Two-Tailed Independent Samples t-test Results:
> cat("Test Statistic:", t_test_result_two_tailed$statistic, "\n")
Test Statistic: -2.972591
> cat("p-value:", t_test_result_two_tailed$p.value, "\n")
p-value: 0.004295311
> cat("Degrees of Freedom:", t_test_result_two_tailed$parameter, "\n")
Degrees of Freedom: 57.9737
>
> # Display the results of the one-tailed t-test
> cat("One-Tailed Independent Samples t-test Results:\n")
One-Tailed Independent Samples t-test Results:
> cat("Test Statistic:", t_test_result_one_tailed$statistic, "\n")
Test Statistic: -2.972591
> cat("p-value:", t_test_result_one_tailed$p.value, "\n")
p-value: 0.002147656
> cat("Degrees of Freedom:", t_test_result_one_tailed$parameter, "\n")
Degrees of Freedom: 57.9737
>
> # Check the significance level
> alpha <- 0.05
> cat("Significance Level (alpha):", alpha, "\n")
Significance Level (alpha): 0.05
>
> # Determine if the differences are statistically significant
> cat("Two-Tailed Test Conclusion:\n")
Two-Tailed Test Conclusion:
> if (t_test_result_two_tailed$p.value < alpha/2) {
+   cat("Reject the null hypothesis. There is a significant difference between the means.\n")
+ } else {
+   cat("Fail to reject the null hypothesis. There is no significant difference between the means.\n")
+ }
Reject the null hypothesis. There is a significant difference between the means.
```

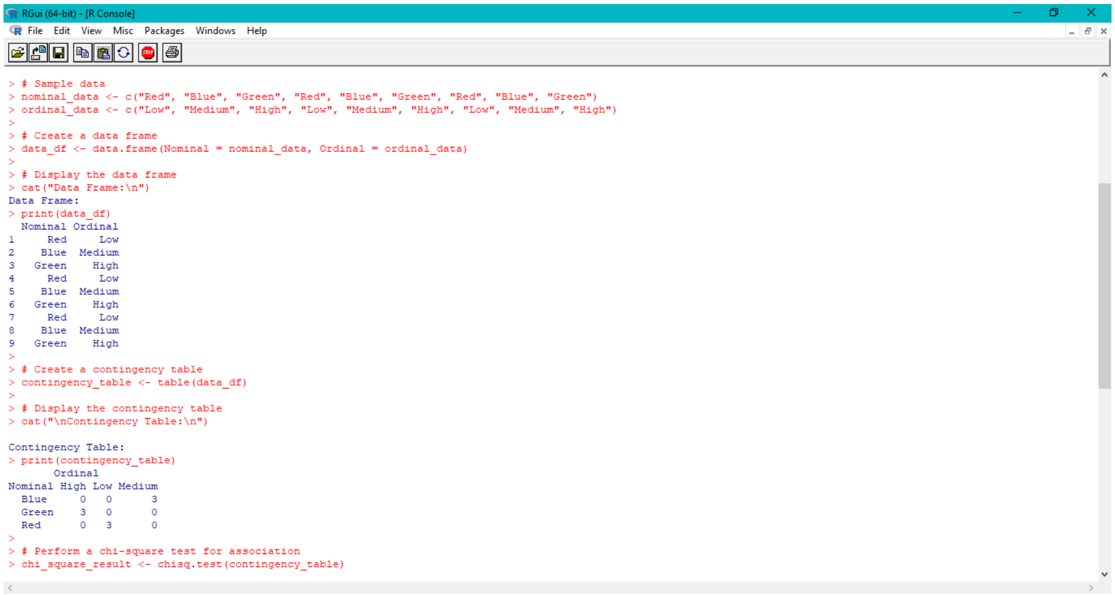
```
>
> cat("One-Tailed Test Conclusion:\n")
One-Tailed Test Conclusion:
> if (t_test_result_one_tailed$p.value < alpha) {
+   cat("Reject the null hypothesis. There is a significant difference; group2 mean is less than group1 mean.\n")
+ } else {
+   cat("Fail to reject the null hypothesis. There is no significant difference or group2 mean is greater than group1 mean.\n")
+ }
Reject the null hypothesis. There is a significant difference; group2 mean is less than group1 mean.
> |
```

Ex.No:10	**BIVARIATE STATICS FOR NOMINAL DATE AND ORDINAL DATE**
Date:	

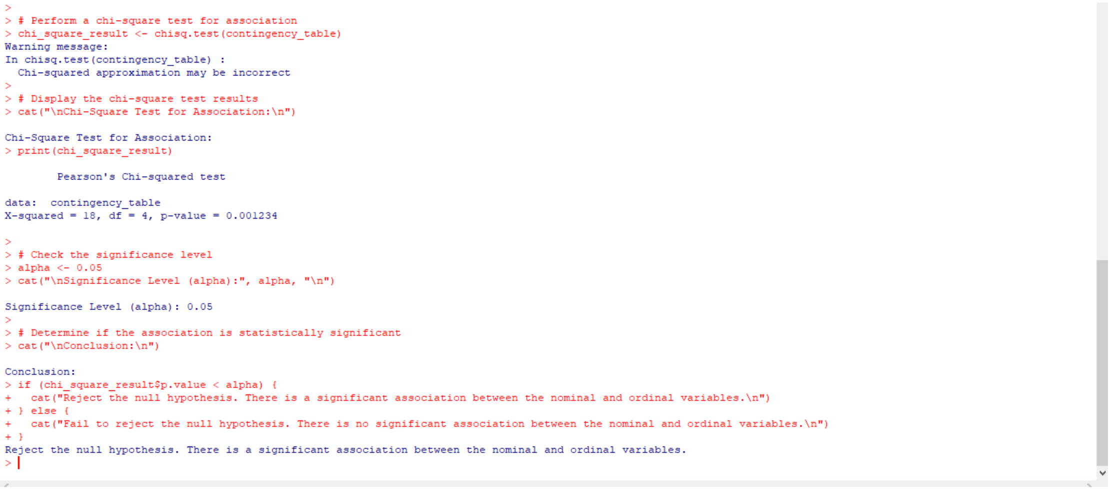
SOURCE CODE

```
# Sample data
nominal_data <- c("Red", "Blue", "Green", "Red", "Blue", "Green", "Red", "Blue", "Green")
ordinal_data <- c("Low", "Medium", "High", "Low", "Medium", "High", "Low", "Medium", "High")

# Create a data frame
data_df <- data.frame(Nominal = nominal_data, Ordinal = ordinal_data)

# Display the data frame
cat("Data Frame:\n")
print(data_df)

# Create a contingency table
contingency_table <- table(data_df)

# Display the contingency table
cat("\nContingency Table:\n")
print(contingency_table)

# Perform a chi-square test for association
chi_square_result <- chisq.test(contingency_table)

# Display the chi-square test results
cat("\nChi-Square Test for Association:\n")
print(chi_square_result)

# Check the significance level
alpha <- 0.05
cat("\nSignificance Level (alpha):", alpha, "\n")

# Determine if the association is statistically significant
cat("\nConclusion:\n")
if (chi_square_result$p.value < alpha) {
  cat("Reject the null hypothesis. There is a significant association between the nominal and ordinal variables.\n")
} else {
  cat("Fail to reject the null hypothesis. There is no significant association between the nominal and ordinal variables.\n")
}
```

OUTPUT


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
> # Sample data
> nominal_data <- c("Red", "Blue", "Green", "Red", "Blue", "Green", "Red", "Blue", "Green")
> ordinal_data <- c("Low", "Medium", "High", "Low", "Medium", "High", "Low", "Medium", "High")
>
> # Create a data frame
> data_df <- data.frame(Nominal = nominal_data, Ordinal = ordinal_data)
>
> # Display the data frame
> cat("Data Frame:\n")
Data Frame:
> print(data_df)
  Nominal Ordinal
1     Red     Low
2    Blue  Medium
3   Green    High
4     Red     Low
5    Blue  Medium
6   Green    High
7     Red     Low
8    Blue  Medium
9   Green    High
>
> # Create a contingency table
> contingency_table <- table(data_df)
>
> # Display the contingency table
> cat("\nContingency Table:\n")

Contingency Table:
> print(contingency_table)
       Ordinal
Nominal High Low Medium
  Blue     0   0      3
  Green    3   0      0
  Red      0   3      0
>
> # Perform a chi-square test for association
> chi_square_result <- chisq.test(contingency_table)
```



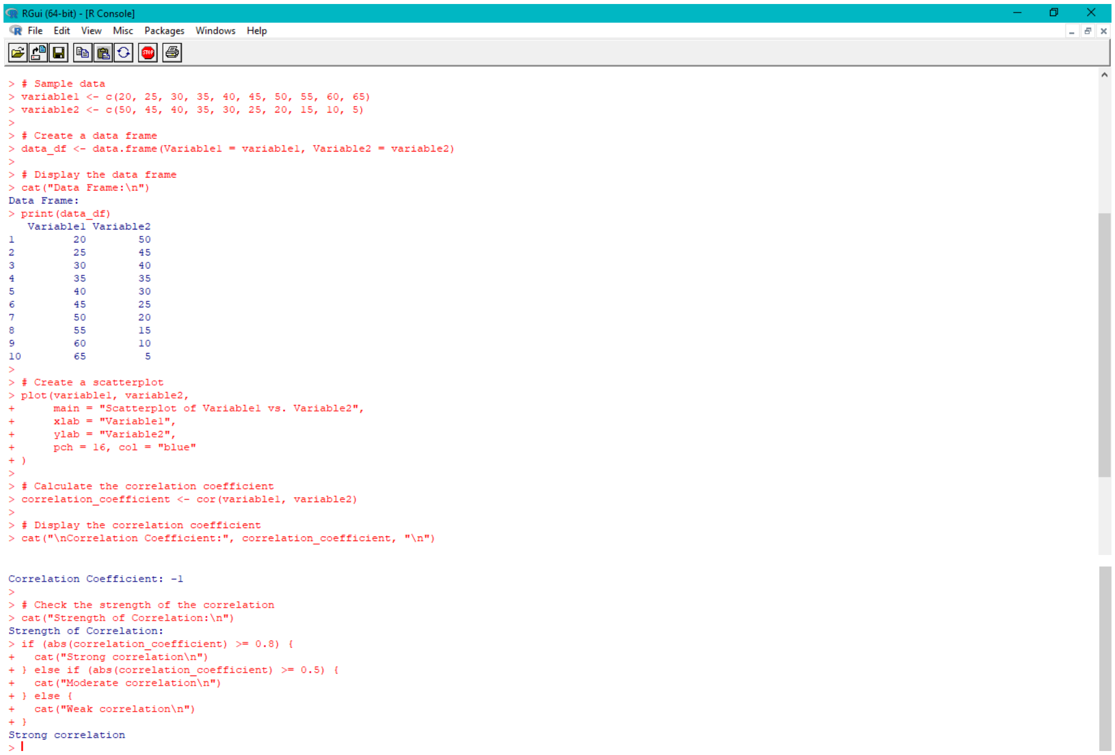
```
>
> # Perform a chi-square test for association
> chi_square_result <- chisq.test(contingency_table)
Warning message:
In chisq.test(contingency_table) :
  Chi-squared approximation may be incorrect
>
> # Display the chi-square test results
> cat("\nChi-Square Test for Association:\n")

Chi-Square Test for Association:
> print(chi_square_result)

        Pearson's Chi-squared test

data:  contingency_table
X-squared = 18, df = 4, p-value = 0.001234

>
> # Check the significance level
> alpha <- 0.05
> cat("\nSignificance Level (alpha):", alpha, "\n")

Significance Level (alpha): 0.05
>
> # Determine if the association is statistically significant
> cat("\nConclusion:\n")

Conclusion:
> if (chi_square_result$p.value < alpha) {
+   cat("Reject the null hypothesis. There is a significant association between the nominal and ordinal variables.\n")
+ } else {
+   cat("Fail to reject the null hypothesis. There is no significant association between the nominal and ordinal variables.\n")
+ }
Reject the null hypothesis. There is a significant association between the nominal and ordinal variables.
>
```

Ex.No:11	**BIVARIATE STATICS FOR INTERVAL/RATIO DATA**
Date:	

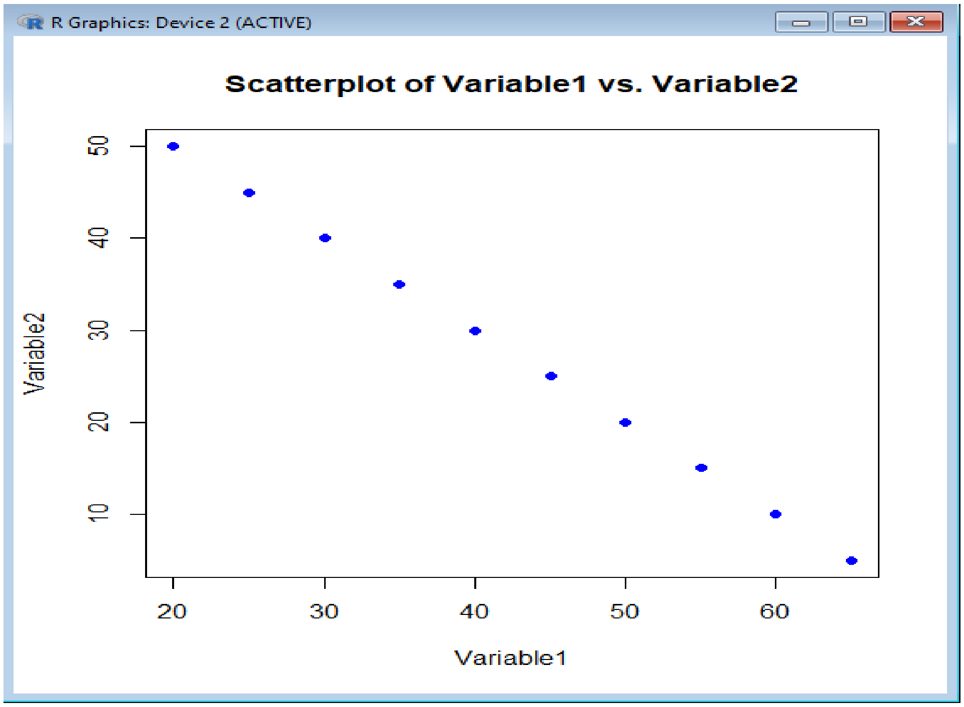
SOURCE CODE

```
#Sample data
variable1 <- c(20, 25, 30, 35, 40, 45, 50, 55, 60, 65)
variable2 <- c(50, 45, 40, 35, 30, 25, 20, 15, 10, 5)

# Create a data frame
data_df <- data.frame(Variable1 = variable1, Variable2 = variable2)

# Display the data frame
cat("Data Frame:\n")
print(data_df)

# Create a scatterplot
plot(variable1, variable2,
     main = "Scatterplot of Variable1 vs. Variable2",
     xlab = "Variable1",
     ylab = "Variable2",
     pch = 16, col = "blue"
)

# Calculate the correlation coefficient
correlation_coefficient <- cor(variable1, variable2)

# Display the correlation coefficient
cat("\nCorrelation Coefficient:", correlation_coefficient, "\n")

# Check the strength of the correlation
cat("Strength of Correlation:\n")
if (abs(correlation_coefficient) >= 0.8) {
  cat("Strong correlation\n")
} else if (abs(correlation_coefficient) >= 0.5) {
  cat("Moderate correlation\n")
} else {
  cat("Weak correlation\n")
}
```

OUTPUT



```
> # Sample data
> variable1 <- c(20, 25, 30, 35, 40, 45, 50, 55, 60, 65)
> variable2 <- c(50, 45, 40, 35, 30, 25, 20, 15, 10, 5)
>
> # Create a data frame
> data_df <- data.frame(Variable1 = variable1, Variable2 = variable2)
>
> # Display the data frame
> cat("Data Frame:\n")
Data Frame:
> print(data_df)
   Variable1 Variable2
1         20        50
2         25        45
3         30        40
4         35        35
5         40        30
6         45        25
7         50        20
8         55        15
9         60        10
10        65         5
>
> # Create a scatterplot
> plot(variable1, variable2,
+      main = "Scatterplot of Variable1 vs. Variable2",
+      xlab = "Variable1",
+      ylab = "Variable2",
+      pch = 16, col = "blue"
+ )
>
> # Calculate the correlation coefficient
> correlation_coefficient <- cor(variable1, variable2)
>
> # Display the correlation coefficient
> cat("\nCorrelation Coefficient:", correlation_coefficient, "\n")


Correlation Coefficient: -1
>
> # Check the strength of the correlation
> cat("Strength of Correlation:\n")
Strength of Correlation:
> if (abs(correlation_coefficient) >= 0.8) {
+   cat("Strong correlation\n")
+ } else if (abs(correlation_coefficient) >= 0.5) {
+   cat("Moderate correlation\n")
+ } else {
+   cat("Weak correlation\n")
+ }
Strong correlation
>
```



R Graphics: Device 2 (ACTIVE)
Scatterplot of Variable1 vs. Variable2
Variable2
Variable1