Exercise 6: Simple Linear Regression Analysis in R

Aim:

To perform a simple linear regression analysis in R, predicting the dependent variable based on the independent variable.

Procedure:

- **Create Dataset**: Use a dataset with two variables: one dependent and one independent.
- **Fit Linear Model**: Fit a linear model using the lm() function.
- **View Model Summary**: Analyze the summary of the regression model.
- **Plot Regression Line**: Visualize the regression line on a scatter plot.

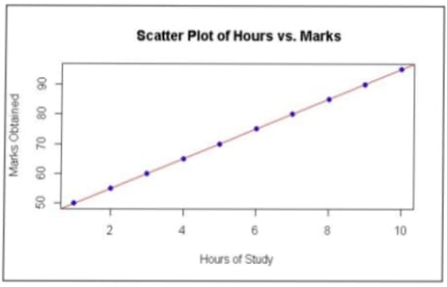
Program:

```
# 1. Create Dataset
# Independent variable: Hours of study
# Dependent variable: Marks obtained
hours <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
marks <- c(50, 55, 60, 65, 70, 75, 80, 85, 90, 95)

# Plot the scatter plot
plot(hours, marks,
  main = "Scatter Plot of Hours vs. Marks",
  xlab = "Hours of Study",
  ylab = "Marks Obtained",
  pch = 16,
  col = "blue")

# 2. Perform Simple Linear Regression
model <- lm(marks ~ hours)

# 3. View Model Summary
print(summary(model))

# 4. Plot the Regression Line
abline(model, col = "red")

# 5. Predict Marks for 6.5 hours of study
predicted_marks <- predict(model, data.frame(hours = 6.5))
cat("Predicted Marks for 6.5 hours of study:", predicted_marks, "\n")
```


Output:

```
Call:
lm(formula = marks ~ hours)

Residuals:
     Min       1Q   Median       3Q      Max
-4.155e-15 -1.582e-15  9.575e-16  1.032e-15  4.688e-15

Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 4.500e+01  1.848e-15 2.434e+16   <2e-16 ***
hours       5.000e+00  2.979e-16 1.678e+16   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.706e-15 on 8 degrees of freedom
Multiple R-squared:     1,     Adjusted R-squared:     1
F-statistic: 2.817e+32 on 1 and 8 DF,  p-value: < 2.2e-16
```

Predicted Marks for 6.5 hours of study: 77.5



Result:

The exercise demonstrates how to fit a simple linear regression model, interpret the results, and make predictions based on the model.

SIMPLE LINEAR REGRESSION ANALYSIS IN R

```
height <- c(150, 160, 170, 180, 190)
weight <- c(50, 60, 65, 75, 85)
data <- data.frame(height, weight)
model <- lm(weight ~ height, data = data)
summary(model)
plot(data$height, data$weight,
     main = "Height vs Weight",
     xlab = "Height (cm)",
     ylab = "Weight (kg)",
     pch = 19,
     col = "blue")
abline(model, col = "red")
new_height <- data.frame(height = 172.5)
predicted_weight <- predict(model, newdata = new_height)
predicted_weight
```

Exercise 7: Chi-Square Test for Independence in R

Aim:

To perform a chi-square test on a contingency table to determine if two categorical variables are independent.

Procedure:

- **Create a Contingency Table**: Create a table showing the frequency of occurrences of two categorical variables.
- **Perform Chi-Square Test**: Use the chisq.test() function to perform the test.
- **Analyze the Result**: Interpret the test result by analyzing the p-value and the test statistic.

Program

```
# 1. Create Contingency Table
# Let's consider a study about the preference of two types of sports (Football and Basketball)
# among males and females in a sample of 100 individuals.

# Contingency Table: Rows = Gender, Columns = Sport Preference
gender_sport <- matrix(c(30, 10, 20, 40), nrow = 2, byrow = TRUE,
                       dimnames = list("Gender" = c("Male", "Female"),
                                       "Sport" = c("Football", "Basketball")))

# Display the contingency table
cat("Contingency Table:\n")
print(gender_sport)

# 2. Perform Chi-Square Test for Independence
chi_square_test <- chisq.test(gender_sport)

# 3. View Test Results
cat("\nChi-Square Test Results:\n")
print(chi_square_test)

# 4. Interpretation
if (chi_square_test$p.value < 0.05) {
  cat("\nConclusion: The variables are not independent (reject the null hypothesis).\n")
} else {
  cat("\nConclusion: The variables are independent (fail to reject the null hypothesis).\n")
}
```


Output:

```
Contingency Table:
                Sport
Gender          Football        Basketball
  Male          30              10
  Female        20              40

Chi-Square Test Results:

        Pearson's Chi-squared test with Yates' continuity correction

data:  gender_sport
X-squared = 15.042, df = 1, p-value = 0.0001052

Conclusion: The variables are not independent (reject the null hypothesis).
```

CHI SQUARE TEST FOR INDEPENDENCE IN R

```
cuisine_table <- matrix(c(25, 30, 20, 35), nrow = 2, byrow = TRUE)
rownames(cuisine_table) <- c("Young", "Middle-aged")
colnames(cuisine_table) <- c("Italian", "Chinese")
cuisine_table
chi_square_result <- chisq.test(cuisine_table)
chi_square_result
if (chi_square_result$p.value < 0.05) {
  print("There is a significant association between age group and cuisine preference.")
} else {
  print("There is no significant association between age group and cuisine preference. The variables are
independent.")
}
```

Exercise 8: One-Way ANOVA in R

Aim:

To perform a one-way ANOVA on a given dataset to test if there are statistically significant differences between the means of multiple groups.

Procedure:

- **Create a Dataset**: Define the dataset with multiple groups.
- **Perform One-Way ANOVA**: Use the aov() function to conduct the analysis.
- **Interpret the Results**: Analyze the p-value and F-statistic to determine whether the group means are significantly different.

Code:

```
# 1. Create a Dataset
# Example: Test scores of students from three different groups (Group A, Group B, and Group C)
scores <- c(88, 90, 85, 92, 95, 78, 82, 87, 91, 86, 83, 89, 94, 80, 77)
groups <- factor(c(rep("Group A", 5), rep("Group B", 5), rep("Group C", 5)))

# Combine into a data frame
data <- data.frame(scores, groups)

# Display the dataset
cat("Dataset:\n")
print(data)

# 2. Perform One-Way ANOVA
anova_result <- aov(scores ~ groups, data = data)

# 3. View the ANOVA Summary
cat("\nOne-Way ANOVA Results:\n")
print(summary(anova_result))

# 4. Interpretation
# Check p-value from the ANOVA summary
p_value <- summary(anova_result)[[1]]["Pr(>F)"][1]
if (p_value < 0.05) {
  cat("\nConclusion: There is a significant difference between the group means (reject the null hypothesis).\n")
} else {
  cat("\nConclusion: There is no significant difference between the group means (fail to reject the null hypothesis).\n")
}
```


Output:

Dataset:

	scores	groups
1	88	Group A
2	90	Group A
3	85	Group A
4	92	Group A
5	95	Group A
6	78	Group B
7	82	Group B
8	87	Group B
9	91	Group B
10	86	Group B
11	83	Group C
12	89	Group C
13	94	Group C
14	80	Group C
15	77	Group C

One-Way ANOVA Results:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
groups	2	93.7	46.87	1.625	0.237
Residuals	12	346.0	28.83		

Conclusion: There is no significant difference between the group means (fail to reject the null hypothesis).

Result:

This program demonstrates how to perform a one-way ANOVA in R, allowing you to analyze whether the means of different groups are statistically different.

ONE –WAY ANOVA IN R

```
weights <- c(68, 72, 65, 70, 74, 60, 63, 67, 69, 64, 76, 78, 71, 73, 75)
diets <- factor(c(rep("Diet A", 5), rep("Diet B", 5), rep("Diet C", 5)))
data <- data.frame(weights, diets)
data
anova_result <- aov(weights ~ diets, data = data)
summary(anova_result)
if (summary(anova_result)[[1]][["Pr(>F)"]][1] < 0.05) {
  print("There is a significant difference in the mean weights of individuals on different diets.")
} else {
  print("There is no significant difference in the mean weights of individuals on different diets.")
}
```

Exercise 9: Two-Sample T-Test in R

Aim:

To perform a two-sample t-test to compare the means of two independent samples.

Procedure:

- **Create a Dataset**: Define two independent samples (e.g., weights of individuals from two different groups).
- **Perform Two-Sample T-Test**: Use the t.test() function to compare the means of the two samples.
- **Interpret the Results**: Analyze the p-value and t-statistic to determine whether the means are significantly different.

Program:

```
# 1. Create a Dataset
# Example: Weights of individuals in Group 1 and Group 2
group1 <- c(68, 72, 65, 70, 74)  # Weights in Group 1
group2 <- c(60, 63, 67, 69, 64)  # Weights in Group 2

# Display the samples
cat("Group 1 Weights:\n")
print(group1)
cat("Group 2 Weights:\n")
print(group2)

# 2. Perform Two-Sample T-Test
t_test_result <- t.test(group1, group2)

# 3. View the T-Test Results
cat("\nTwo-Sample T-Test Results:\n")
print(t_test_result)

# 4. Interpretation
# Check p-value from the t-test result
p_value <- t_test_result$p.value
if (p_value < 0.05) {
 cat("\nConclusion: There is a significant difference between the means of the two groups (reject the null hypothesis).\n")
} else {
 cat("\nConclusion: There is no significant difference between the means of the two groups (fail to reject the null hypothesis).\n")
}
```


Output:

```
Group 1 Weights:
[1] 68 72 65 70 74
Group 2 Weights:
[1] 60 63 67 69 64

Two-Sample T-Test Results:
        Welch Two Sample t-test

data:  group1 and group2
t = 2.3491, df = 7.9999, p-value = 0.04675
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.09542744 10.30457256
sample estimates:
mean of x mean of y
   69.8      64.6
```

Conclusion: There is a significant difference between the means of the two groups (reject the null hypothesis).

TWO-SAMPLE t-TEST IN R

```
group_A <- c(85, 88, 90, 78, 95, 80, 85)
group_B <- c(78, 82, 84, 75, 89, 83, 81)
data <- data.frame(group_A, group_B)
t_test_result <- t.test(group_A, group_B, var.equal = FALSE)
t_test_result
if (t_test_result$p.value < 0.05) {
  print("There is a significant difference between the means of the two groups.")
} else {
  print("There is no significant difference between the means of the two groups.")
}
```

Exercise 10: Plotting Various Probability Distributions in R

Aim:

To plot different types of probability distributions using R, including Normal, Binomial, Poisson, and Exponential distributions.

Procedure:

- **Create Data for Distributions**: Generate data for different probability distributions.
- **Plot Distributions**: Use the plot(), hist(), and curve() functions to visualize the distributions.
- **Interpret the Graphs**: Analyze the shape and behavior of each distribution.

Program:

```
# 1. Plot Normal Distribution
cat("Plotting Normal Distribution:\n")
x_normal <- seq(-10, 10, length = 100)
y_normal <- dnorm(x_normal, mean = 0, sd = 1)

# Plot Normal Distribution
plot(x_normal, y_normal, type = "l", col = "blue", lwd = 2,
   main = "Normal Distribution", xlab = "x", ylab = "Density")

# 2. Plot Binomial Distribution
cat("Plotting Binomial Distribution:\n")
x_binom <- 0:10
y_binom <- dbinom(x_binom, size = 10, prob = 0.5)

# Plot Binomial Distribution
barplot(y_binom, names.arg = x_binom, col = "green",
     main = "Binomial Distribution", xlab = "Number of Successes", ylab = "Probability")

# 3. Plot Poisson Distribution
cat("Plotting Poisson Distribution:\n")
x_pois <- 0:15
y_pois <- dpois(x_pois, lambda = 4)

# Plot Poisson Distribution
barplot(y_pois, names.arg = x_pois, col = "red",
     main = "Poisson Distribution", xlab = "Number of Events", ylab = "Probability")

# 4. Plot Exponential Distribution
cat("Plotting Exponential Distribution:\n")
x_exp <- seq(0, 5, length = 100)
```

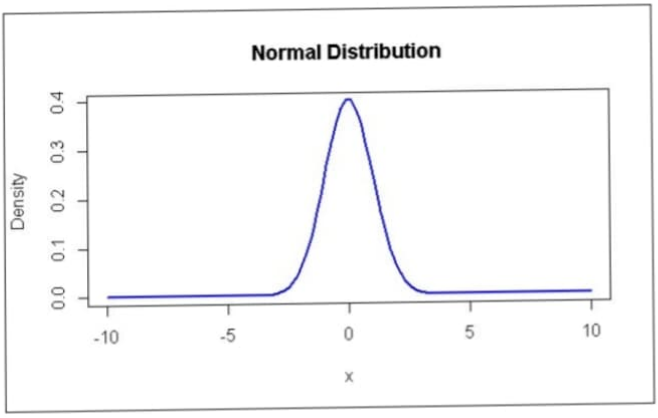


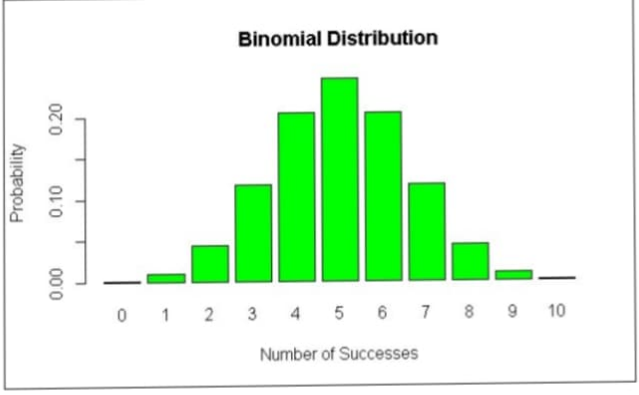
```
y_exp <- dexp(x_exp, rate = 1)

# Plot Exponential Distribution
plot(x_exp, y_exp, type = "l", col = "purple", lwd = 2,
   main = "Exponential Distribution", xlab = "x", ylab = "Density")
```

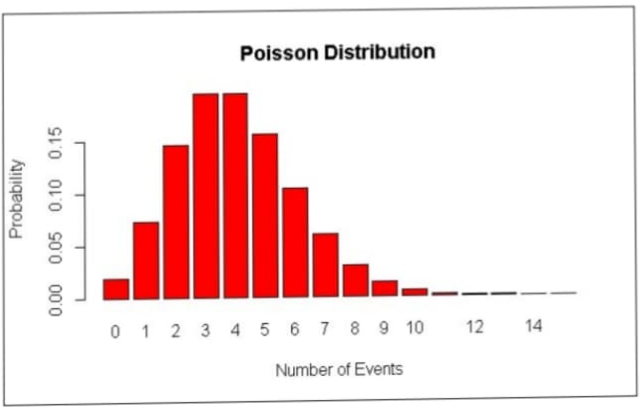
Explanation of Distributions:

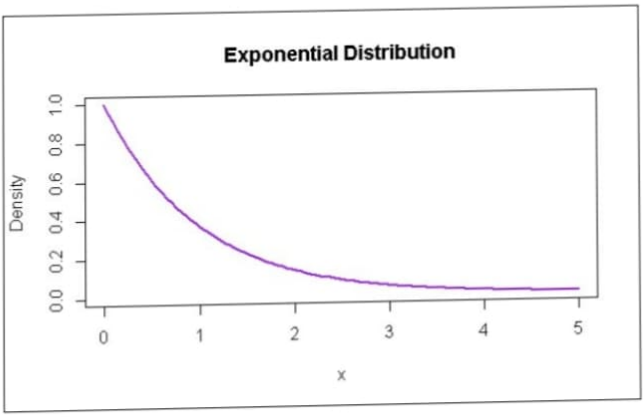
1. **Normal Distribution**:
   - The normal distribution is bell-shaped, symmetric, and describes continuous data. The plot uses the dnorm() function.
2. **Binomial Distribution**:
   - The binomial distribution represents the probability of a given number of successes in a fixed number of independent trials. The plot uses the dbinom() function.
3. **Poisson Distribution**:
   - The Poisson distribution represents the probability of a given number of events happening in a fixed interval of time or space. The plot uses the dpois() function.
4. **Exponential Distribution**:
   - The exponential distribution describes the time between events in a Poisson process. The plot uses the dexp() function.

Output:









Result:

This program successfully plots various probability distributions, demonstrating the characteristics and behavior of each distribution type.

PLOTTING GAMMA DISTRIBUTION IN R

```
x <- seq(0, 20, length.out = 100)
gamma_shape1 <- dgamma(x, shape = 2, scale = 1)  # Shape = 2, Scale = 1
gamma_shape2 <- dgamma(x, shape = 5, scale = 1)  # Shape = 5, Scale = 1
gamma_shape3 <- dgamma(x, shape = 9, scale = 1)  # Shape = 9, Scale = 1
plot(x, gamma_shape1, type = "l", col = "blue", lwd = 2,
    main = "Gamma Distribution with Different Shape Parameters",
    xlab = "Time",
    ylab = "Density")
lines(x, gamma_shape2, col = "red", lwd = 2)
lines(x, gamma_shape3, col = "green", lwd = 2)
legend("topright", legend = c("Shape = 2", "Shape = 5", "Shape = 9"),
     col = c("blue", "red", "green"), lwd = 2)
```