|  |  |  |  |
| --- | --- | --- | --- |
| 1. A car travels 25 miles at 25 miles per hour (mi/h), 25 miles at 50 mph, and 25 miles at 75 mph. Write a program to find the arithmetic mean of the three velocities and the harmonic mean of the three velocities.Which is correct?   CODE:  # Given velocities  velocities <- c(25, 50, 75)  # Arithmetic Mean  arith\_mean <- mean(velocities)  # Harmonic Mean  harmonic\_mean <- length(velocities) / sum(1 / velocities)  # Print results  print(paste("Arithmetic Mean:", arith\_mean))  print(paste("Harmonic Mean:", harmonic\_mean))   |  |  |  | | --- | --- | --- | | 2. Enter the following details of wages of 65 employees at the ABC Ltd. In Excel: Wages  Number of Employees     |  | | --- | | 25000-25999 8 26000-26999 10  27000-27999 16  28000-28999 14  29000-29999 10  30000-30999 5  31000-31999 2  Total 65 |  |  | | --- | |  | | |

Import the data in R and find the mean, standard deviation and variance of wage and mode wage of the 65 employees.

CODE:

# Data: Wages and Number of Employees

wages <- c(25500, 26500, 27500, 28500, 29500, 30500, 31500) # Midpoints of ranges

employees <- c(8, 10, 16, 14, 10, 5, 2) # Frequencies

# Weighted Mean

mean\_wage <- sum(wages \* employees) / sum(employees)

# Standard Deviation

variance\_wage <- sum(employees \* (wages - mean\_wage)^2) / sum(employees)

std\_dev\_wage <- sqrt(variance\_wage)

# Mode (wage range with highest frequency)

mode\_wage <- wages[which.max(employees)]

# Print results

print(paste("Mean Wage:", mean\_wage))

print(paste("Standard Deviation:", std\_dev\_wage))

print(paste("Variance:", variance\_wage))

print(paste("Mode Wage:", mode\_wage))

1. Enter the following details of wages of 65 employees at the ABC Ltd. In Excel:

Wages Number of Employees

25000-25999 8

26000-26999 10

27000-27999 16

28000-28999 14

29000-29999 10

30000-30999 5

31000-31999

Total 65

Import the data in R and find the median wage and mode wage of the 65 employees.

CODE:

# Create a cumulative frequency table

cumulative\_freq <- cumsum(employees)

# Find the median class (n/2 rule)

n <- sum(employees)

median\_class\_index <- which(cumulative\_freq >= n / 2)[1]

# Find median wage

L <- wages[median\_class\_index] # Lower boundary

cf <- ifelse(median\_class\_index == 1, 0, cumulative\_freq[median\_class\_index - 1])

f <- employees[median\_class\_index]

h <- 1000 # Assuming class width

median\_wage <- L + ((n / 2 - cf) / f) \* h

# Print results

print(paste("Median Wage:", median\_wage))

print(paste("Mode Wage:", mode\_wage))

4. Enter the following data sets in Excel:

a) 12, 6, 7, 3, 15, 10, 18, 5 b) 9, 3, 8, 8, 9, 8, 9, 18.

Import the data in R and find standard deviation and variance of the data sets using R .

CODE:

# Data Set A

data\_a <- c(12, 6, 7, 3, 15, 10, 18, 5)

# Data Set B

data\_b <- c(9, 3, 8, 8, 9, 8, 9, 18)

# Standard Deviation and Variance

std\_dev\_a <- sd(data\_a)

var\_a <- var(data\_a)

std\_dev\_b <- sd(data\_b)

var\_b <- var(data\_b)

# Print results

print(paste("Standard Deviation (A):", std\_dev\_a))

print(paste("Variance (A):", var\_a))

print(paste("Standard Deviation (B):", std\_dev\_b))

print(paste("Variance (B):", var\_b))

1. Enter the following table of three distributions f1, f2 and f3 for the variable X in

EXCEL.

X f1 f2 f3

1. 10 1 1
2. 5 2 2
3. 2 14 2
4. 2 2 5
5. 1 1 10

Import the data in R and write and program to find Pearson’s first and second coefficients of skewness.

CODE:

# Data for X and frequencies

x <- c(0, 1, 2, 3, 4)

f1 <- c(10, 5, 2, 2, 1)

f2 <- c(1, 2, 14, 2, 1)

f3 <- c(1, 2, 2, 5, 10)

# Function to calculate skewness

skewness <- function(x, f) {

mean\_x <- sum(x \* f) / sum(f)

sd\_x <- sqrt(sum(f \* (x - mean\_x)^2) / sum(f))

mode\_x <- x[which.max(f)]

# Pearson’s First Coefficient of Skewness

skew1 <- (mean\_x - mode\_x) / sd\_x

# Pearson’s Second Coefficient of Skewness

median\_x <- median(rep(x, f)) # Weighted median

skew2 <- (3 \* (mean\_x - median\_x)) / sd\_x

return(list(Skewness1 = skew1, Skewness2 = skew2))

}

# Compute skewness for each distribution

skew\_f1 <- skewness(x, f1)

skew\_f2 <- skewness(x, f2)

skew\_f3 <- skewness(x, f3)

# Print results

print(skew\_f1)

print(skew\_f2)

print(skew\_f3)

1. Many casinos use card-dealing machines to deal cards at random. Occasionally, the machine is tested to ensure an equal likelihood of dealing for each suit. To conduct the test, 1,500 cards are dealt from the machine, while the number of cards in each suit is counted. Theoretically, 375 cards should be dealt from each suit. But this is not the case as shown in the following table:

Spades Diamonds Clubs Hearts

Observed 402 358 273 467

Expected 375 375 375 375

Enter the data in Excel. Import the date in R and write a program using chi-square test to determine if the discrepancies are significant. If the discrepancies are significant, then the game would not be fair.

CODE:

# Observed and Expected frequencies

observed <- c(402, 358, 273, 467)

expected <- c(375, 375, 375, 375)

# Chi-Square Test

chi\_test <- chisq.test(observed, p = expected / sum(expected))

# Print results

print(chi\_test)

7.A business owner had been working to improve employee relations in his company. He predicted that he met his goal of increasing employee satisfaction from 65% to 80%. Employees from four departments were asked if they were satisfied with the working conditions of the company. The results are shown in the following table:

Finance Sales HR Technology

Satisfied 12 38 5 8

Dissatisfied 7 19 3 1

Total 19 57 8 9

Enter the data in Excel. Import the date from Excel to R and write a program suing chi-square test to determine whether the results support or reject the business owner's prediction.

CODE:

# Create a contingency table

satisfaction\_table <- matrix(c(12, 38, 5, 8, 7, 19, 3, 1), nrow = 2, byrow = TRUE)

# Add row and column names

rownames(satisfaction\_table) <- c("Satisfied", "Dissatisfied")

colnames(satisfaction\_table) <- c("Finance", "Sales", "HR", "Technology")

# Chi-Square Test

chi\_test\_satisfaction <- chisq.test(satisfaction\_table)

# Print results

print(chi\_test\_satisfaction)

8.Suppose the number of games in which major league baseball players play during their careers is normally distributed with mean equal to 1500 games and standard deviation equal to 350 games.

Use R to solve the following problems.

(a) What percentage play in fewer than 750 games? (b) What percentage play in more than 2000 games?

(c) Find the 90th percentile for the number of games played during a career.

CODE:

# Given values

mean\_games <- 1500

sd\_games <- 350

# (a) Percentage playing in fewer than 750 games

prob\_750 <- pnorm(750, mean\_games, sd\_games) \* 100

# (b) Percentage playing in more than 2000 games

prob\_2000 <- (1 - pnorm(2000, mean\_games, sd\_games)) \* 100

# (c) 90th percentile of games played

percentile\_90 <- qnorm(0.90, mean\_games, sd\_games)

# Print results

print(paste("Percentage playing in fewer than 750 games:", prob\_750))

print(paste("Percentage playing in more than 2000 games:", prob\_2000))

print(paste("90th percentile for games played:", percentile\_90))

9. Enter the following table which shows the heights(H) to the nearest inch (in) and the weights(W) to the nearest pound (lb) of a sample of 12 male students drawn at random from the first-year students at College.

H 70 63 72 60 66 70 74 65 62 67 65 68

W 155 150 180 135 156 168 178 160 132 145 139 152

Import the data in R and write a program to fit a least squares line using a) H as the independent variable

b) H as dependent variable

CODE:

# Given data

H <- c(70, 63, 72, 60, 66, 70, 74, 65, 62, 67, 65, 68)

W <- c(155, 150, 180, 135, 156, 168, 178, 160, 132, 145, 139, 152)

# Perform linear regression

model <- lm(W ~ H)

# Summary of regression

summary(model)

# Plot data and regression line

plot(H, W, main="Height vs Weight Regression", xlab="Height (in)", ylab="Weight (lb)", pch=16, col="blue")

abline(model, col="red", lwd=2)

10. Enter the total agricultural exports in millions of dollars in Excel:

Year 2000 2001 2002 2003 2004 2005

Total 51246 53659 53115 59364 61383 62958

Value

Import the data in R and perform the following

(a) Graph the data and show the least-squares regression line. (b) Find and plot the trend line for the data.

(c) Estimate the value of total agricultural exports in the year 2006.

CODE:

# Given data

Year <- c(2000, 2001, 2002, 2003, 2004, 2005)

Exports <- c(51246, 53659, 53115, 59364, 61383, 62958)

# Perform linear regression

model\_exports <- lm(Exports ~ Year)

# Summary of model

summary(model\_exports)

# Plot data and regression line

plot(Year, Exports, main="Agricultural Exports Over Time", xlab="Year", ylab="Exports (Millions)", pch=16, col="blue")

abline(model\_exports, col="red", lwd=2)

# Estimate exports for the year 2006

prediction\_2006 <- predict(model\_exports, newdata = data.frame(Year = 2006))

# Print estimated exports for 2006

print(paste("Estimated total agricultural exports in 2006:", prediction\_2006))

11. Enter the following table in Excel which shows the first two grades (denoted by First Quiz X and Second Quiz Y, respectively) of 10 students on two short quizzes in biology.

1. 6 5 8 8 7 6 10 4 9 7
2. 8 7 7 10 5 10 8 6 8 6

Import the data in R and write programs for the following: (a) Find the least-squares regression line of Y on X.

(b) Find the least-squares regression line of X on Y.

CODE:

# Given data

X <- c(6, 5, 8, 8, 7, 6, 10, 4, 9, 7)

Y <- c(8, 7, 10, 5, 10, 8, 6, 8, 6, 6)

# Regression of Y on X

model\_Y\_on\_X <- lm(Y ~ X)

summary(model\_Y\_on\_X)

# Regression of X on Y

model\_X\_on\_Y <- lm(X ~ Y)

summary(model\_X\_on\_Y)

# Plot and regression line for Y on X

plot(X, Y, main="Regression of Y on X", xlab="X (Quiz 1)", ylab="Y (Quiz 2)", pch=16, col="blue")

abline(model\_Y\_on\_X, col="red", lwd=2)

# Plot and regression line for X on Y

plot(Y, X, main="Regression of X on Y", xlab="Y (Quiz 2)", ylab="X (Quiz 1)", pch=16, col="green")

abline(model\_X\_on\_Y, col="purple", lwd=2)

14. Write a program in R to create two matrices A and B of order 3 X 3 and perform the following operations:

1. Add matrices A and B
2. Multiply matrices A and B
3. Find the inverse of matrix A
4. Find the inverse of matrix B

Find the transpose of matrix B

CODE:

# Define matrices A and B (3x3)

A <- matrix(c(2, 4, 1, 3, 5, 7, 6, 8, 9), nrow=3, byrow=TRUE)

B <- matrix(c(1, 3, 5, 2, 4, 6, 7, 8, 9), nrow=3, byrow=TRUE)

# (a) Add matrices A and B

A\_plus\_B <- A + B

# (b) Multiply matrices A and B

A\_times\_B <- A %\*% B

# (c) Find the inverse of matrix A

inv\_A <- solve(A)

# (d) Find the inverse of matrix B

if (det(B) != 0) {

inv\_B <- solve(B)

print("Inverse of B:")

print(inv\_B)

} else {

print("Matrix B is singular; using pseudo-inverse instead:")

library(MASS)

inv\_B <- ginv(B)

print(inv\_B)

}

# (e) Find the transpose of matrix B

transpose\_B <- t(B)

# Print results

print("A + B:")

print(A\_plus\_B)

print("A \* B:")

print(A\_times\_B)

print("Inverse of A:")

print(inv\_A)

print("Inverse of B:")

print(inv\_B)

print("Transpose of B:")

print(transpose\_B)