

**School of Computing Science & Engineering**

**Laboratory Record**

**Foundations of Data Science –CDS3006**

|  |  |
| --- | --- |
| Slot | A11+D11+A12+D12+A13 |
| Class ID | 10017 |
| Semester | Interim Semester 2023-2024 |
| Name of the student | ABINASHA DASH |
| Register Number | 22BAI10097 |

**Index**

|  |  |  |
| --- | --- | --- |
| **Exp. No.** | **Name of the Experiment** | **Page Number** |
|  | Descriptive Statistics  Case Study; Party Dataset | 1 |
|  | Descriptive Statistics  Case Study: NSE Dataset ( HCLTECH, TCS,WIPRO, AXIS, HDFC,SBIN) | 5 |
|  | One Sample T test  Case Study1: Reliance Mart  Case Study 2: Hypothetical Data (Score) | 22 |
|  | Two Sample T test (Paired)  Case Study1: Pre and Post Test Score  Case Study 2: Crocin Tablet Temperature dataset |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| Exp No: 1 | Descriptive Statistics: Case Study; Party Dataset |
| Date: 21/11/2023 |

**Aim:** Analyze and interpret key descriptive statistical summaries for a Birthday party dataset.

**Objective:** Compute and analyse measures of central tendency, such as mean, median, and mode, to understand the typical or central values of the dataset. Calculate measures of dispersion, including range and standard deviation, to assess the spread and variability of the data points.

**Program Code:**

1. **Program:**

library("xlsx")

> library("e1071")

> PartyDataset <- read.xlsx("C:/users/admin/Downloads/PartyDataset.xlsx", sheetName = "Sheet1")

> data <- PartyDataset$Data

> mean\_value <- mean(data)

> median\_value <- median(data)

> mode\_value <- names(table(data))[table(data) == max(table(data))]

> std\_deviation <- sd(data)

> variance\_value <- var(data)

> coefficient\_of\_variation <- (std\_deviation / mean\_value) \* 100

> skewness\_value <- skewness(data)

> kurtosis\_value <- kurtosis(data)

print\_statistics <- function(mean\_value, median\_value, mode\_value, std\_deviation, variance\_value, coefficient\_of\_variation, skewness\_value, kurtosis\_value) {

+ cat(paste("Mean:", mean\_value, "\n"))

+ cat(paste("Median:", median\_value, "\n"))

+ cat(paste("Mode:", mode\_value, "\n"))

+ cat(paste("Standard Deviation:", std\_deviation, "\n"))

+ cat(paste("Variance:", variance\_value, "\n"))

+ cat(paste("Coefficient of Variation:", coefficient\_of\_variation, "%\n"))

+ cat(paste("Skewness:", skewness\_value, "\n"))

+ cat(paste("Kurtosis:", kurtosis\_value, "\n"))

+ }

> print\_statistics(mean\_value, median\_value, mode\_value, std\_deviation, variance\_value, coefficient\_of\_variation, skewness\_value, kurtosis\_value)

**OUTPUT:**

Mean: 62.9333333333333

Median: 65

Mode: 67

Standard Deviation: 22.9205667676954

Variance: 525.352380952381

Coefficient of Variation: 36.4203921096854 %

Skewness: -0.287209790135492

Kurtosis: 3.11782625791679

**Python-Program:**

def calculate\_statistics(data):

# Mean

mean\_value = data.mean()

# Median

median\_value = data.median()

# Mode

mode\_value = data.mode().iloc[0] # Use iloc[0] to get the first mode if there are multiple modes

# Standard Deviation

std\_deviation = data.std()

# Variance

variance\_value = data.var()

# Coefficient of Variation (CV)

coefficient\_of\_variation = (std\_deviation / mean\_value) \* 100

return mean\_value, median\_value, mode\_value, std\_deviation, variance\_value, coefficient\_of\_variation

def read\_csv(file\_path):

# Read excel file

df = pd.read\_excel(file\_path)

# Data is in a column named 'Data'

data = df['Data']

# Call function to calculate

mean, median, mode, std\_dev, variance, cv = calculate\_statistics(data)

# Print

print(f"Mean: {mean}")

print(f"Median: {median}")

print(f"Mode: {mode}")

print(f"Standard Deviation: {std\_dev}")

print(f"Variance: {variance}")

print(f"Coefficient of Variation: {cv}%")

def main():

# Read 1st excel file

read\_csv('PartyDataset.xlsx')

main()

**Output:**

Mean: 62.93333333333334

Median: 64.0

Mode: 67.0

Standard Deviation: 22.143371217790865

Variance: 490.3288888888889

Coefficient of Variation: 35.18544155369311%

**Inference:**

The descriptive statistics for the variable 'Data' in the Party Dataset reveal a roughly symmetric distribution with a mean of approximately 62.93 and a median of 65, suggesting a central tendency around these values. The moderate negative skewness (-0.29) indicates a slight tail to the left, while the positive kurtosis (3.12) suggests heavier tails and potentially some outliers. The standard deviation of 22.92 reflects a considerable degree of variability around the mean, with a coefficient of variation of 36.42%, indicating a relatively high level of relative variability in the data.

**Tool/ Language:**

R-Studio

Jupyter notebook

|  |  |
| --- | --- |
| Exp No: 2 | Descriptive Statistics: Case Study; NSE Dataset ( HCLTECH, TCS,WIPRO, AXIS, HDFC,SBIN) |
| Date: 23/11/2023 |

**Aim:** Analyze and present statistical summaries in stock prices, volumes, and fluctuations for informed investment strategies and comparisons between the selected companies.

**Objective:** Describe the historical performance, volatility, correlations, and key statistical metrics (such as Central tendency and Dispersion) of stock market data for SBIN, AXIS, HDFC, HCLTECH, WIPRO, and TCS to support comprehensive analysis and decision-making in investment portfolios.

**Program Code:**

**R-Program:**

library("xlsx")

library("e1071")

# Read excel file

sbi <- read.xlsx("C:/Users/admin/Downloads/SBI.xlsx", sheetName = "Sheet1")

axis <- read.xlsx("C:/Users/admin/Downloads/AXISBANK.xlsx", sheetName = "Sheet1")

hdfc <- read.xlsx("C:/Users/admin/Downloads/HDFCBANK.xlsx", sheetName = "Sheet1")

tcs <- read.xlsx("C:/Users/admin/Downloads/TCS.xlsx", sheetName = "Sheet1")

wipro <- read.xlsx("C:/Users/admin/Downloads/WIPRO.xlsx", sheetName = "Sheet1")

hcl <- read.xlsx("C:/Users/admin/Downloads/HCLTECH.xlsx", sheetName = "Sheet1")

# Data is in a column named 'Data'

sbiData <- sbi$close

axisData <- axis$close

hdfcData <- hdfc$close

tcsData <- tcs$close

wiproData <- wipro$close

hclData <- hcl$close

calculate\_statistics <- function(data) {

# Calculate statistics

mean\_value <- mean(data)

median\_value <- median(data)

mode\_value <- names(table(data))[table(data) == max(table(data))]

std\_deviation <- sd(data)

variance\_value <- var(data)

coefficient\_of\_variation <- (std\_deviation / mean\_value) \* 100

skewness\_value <- skewness(data)

kurtosis\_value <- kurtosis(data)

# Return the calculated statistics as a list

return(list(mean\_value = mean\_value,

median\_value = median\_value,

mode\_value = mode\_value,

std\_deviation = std\_deviation,

variance\_value = variance\_value,

coefficient\_of\_variation = coefficient\_of\_variation,

skewness\_value = skewness\_value,

kurtosis\_value = kurtosis\_value))

}

# Call the function with the data

sbiStatistics <- calculate\_statistics(sbiData)

axisStatistics <- calculate\_statistics(axisData)

hdfcStatistics <- calculate\_statistics(hdfcData)

tcsStatistics <- calculate\_statistics(tcsData)

wiproStatistics <- calculate\_statistics(wiproData)

hclStatistics <- calculate\_statistics(hclData)

# Access the calculated statistics

print\_statistics <- function(statistics) {

mean\_value <- statistics$mean\_value

median\_value <- statistics$median\_value

mode\_value <- statistics$mode\_value

std\_deviation <- statistics$std\_deviation

variance\_value <- statistics$variance\_value

coefficient\_of\_variation <- statistics$coefficient\_of\_variation

skewness\_value <- statistics$skewness\_value

kurtosis\_value <- statistics$kurtosis\_value

cat(paste("Mean:", mean\_value, "\n"))

cat(paste("Median:", median\_value, "\n"))

cat(paste("Mode:", mode\_value, "\n"))

cat(paste("Standard Deviation:", std\_deviation, "\n"))

cat(paste("Variance:", variance\_value, "\n"))

cat(paste("Coefficient of Variation:", coefficient\_of\_variation, "%\n"))

cat(paste("Skewness:", skewness\_value, "\n"))

cat(paste("Kurtosis:", kurtosis\_value, "\n"))

}

# Call the function with the statistics

cat(paste("\n","Sbi","\n"))

print\_statistics(sbiStatistics)

cat(paste("\n","Axis","\n"))

print\_statistics(axisStatistics)

cat(paste("\n","Hdfc","\n"))

print\_statistics(hdfcStatistics)

cat(paste("\n","Tcs","\n"))

print\_statistics(tcsStatistics)

cat(paste("\n","Wipro","\n"))

print\_statistics(wiproStatistics)

cat(paste("\n","Hcl","\n"))

print\_statistics(hclStatistics)

**OUTPUT:**

Sbi

Mean: 569.827272727273

Median: 569.25

Mode: 547

Mode: 552.95

Mode: 556.2

Mode: 558.95

Mode: 561.15

Mode: 561.5

Mode: 563.05

Mode: 563.75

Mode: 565.2

Mode: 565.55

Mode: 566.4

Mode: 572.1

Mode: 574.35

Mode: 578.15

Mode: 578.35

Mode: 579.5

Mode: 579.75

Mode: 580.3

Mode: 581.3

Mode: 581.35

Mode: 584.65

Mode: 584.7

Standard Deviation: 11.048450950168

Variance: 122.068268398268

Coefficient of Variation: 1.93891227727458 %

Skewness: -0.289410678792737

Kurtosis: -1.20269290371236

Axis

Mean: 1000.375

Median: 996.225

Mode: 955.45

Mode: 964.3

Mode: 971.8

Mode: 972.05

Mode: 981.85

Mode: 982.95

Mode: 988.7

Mode: 988.85

Mode: 989.2

Mode: 991.8

Mode: 994.35

Mode: 998.1

Mode: 1002.75

Mode: 1009.4

Mode: 1020.9

Mode: 1021.7

Mode: 1021.9

Mode: 1025.35

Mode: 1026.35

Mode: 1029.2

Mode: 1029.95

Mode: 1041.35

Standard Deviation: 24.1066417937345

Variance: 581.130178571428

Coefficient of Variation: 2.40976051917876 %

Skewness: -0.0646319784806038

Kurtosis: -1.2408155212342

Hdfc

Mean: 1492.02272727273

Median: 1491.525

Mode: 1463.4

Mode: 1469.15

Mode: 1474.5

Mode: 1476.5

Mode: 1476.75

Mode: 1483.75

Mode: 1485.1

Mode: 1485.65

Mode: 1487.25

Mode: 1488.8

Mode: 1491.5

Mode: 1491.55

Mode: 1494.5

Mode: 1496.5

Mode: 1500

Mode: 1504.4

Mode: 1505.1

Mode: 1505.2

Mode: 1506.05

Mode: 1508.35

Mode: 1512.55

Mode: 1517.95

Standard Deviation: 14.5715372205439

Variance: 212.329696969696

Coefficient of Variation: 0.976629709064772 %

Skewness: -0.134770317667062

Kurtosis: -0.996200211210819

Tcs

Mean: 3397.42954545455

Median: 3374.5

Mode: 3330.65

Mode: 3331.55

Mode: 3333.45

Mode: 3336.75

Mode: 3347.45

Mode: 3350.9

Mode: 3350.95

Mode: 3357.3

Mode: 3360.1

Mode: 3368.75

Mode: 3370.45

Mode: 3378.55

Mode: 3380.25

Mode: 3381.55

Mode: 3390.1

Mode: 3404.3

Mode: 3410.15

Mode: 3497.85

Mode: 3502.45

Mode: 3510.2

Mode: 3519.6

Mode: 3530.15

Standard Deviation: 67.4325582996164

Variance: 4547.14991883116

Coefficient of Variation: 1.98481108724786 %

Skewness: 0.918748848022892

Kurtosis: -0.782298495045613

Wipro

Mean: 386.236363636364

Median: 382.8

Mode: 378.25

Mode: 379.3

Mode: 380.85

Mode: 380.95

Mode: 381.65

Mode: 381.8

Mode: 381.85

Mode: 382

Mode: 382.2

Mode: 382.35

Mode: 382.45

Mode: 383.15

Mode: 383.3

Mode: 383.55

Mode: 383.65

Mode: 384.65

Mode: 391.65

Mode: 395.4

Mode: 397.1

Mode: 400.2

Mode: 400.25

Mode: 400.65

Standard Deviation: 7.42414307734572

Variance: 55.1179004329004

Coefficient of Variation: 1.92217610155823 %

Skewness: 0.99016877098657

Kurtosis: -0.724575908101103

Hcl

Mean: 1274.52272727273

Median: 1268.175

Mode: 1229

Mode: 1231.8

Mode: 1236.65

Mode: 1255.45

Mode: 1258.95

Mode: 1260.2

Mode: 1261

Mode: 1264.75

Mode: 1266.25

Mode: 1267.5

Mode: 1267.65

Mode: 1268.7

Mode: 1269.7

Mode: 1273.4

Mode: 1273.95

Mode: 1276

Mode: 1276.65

Mode: 1309.15

Mode: 1311.05

Mode: 1325.4

Mode: 1326.6

Mode: 1329.7

Standard Deviation: 28.8926532401476

Variance: 834.785411255411

Coefficient of Variation: 2.26693903701295 %

Skewness: 0.538882918538928

Kurtosis: -0.618382286287375

**PYTHON PROGRAM:**

import scipy.stats as stats

import pandas as pd

def calculate\_statistics(data):

# Mean

mean\_value = data.mean()

# Median

median\_value = data.median()

# Mode

mode\_value = data.mode().iloc[0] # Use iloc[0] to get the first mode if there are multiple modes

# Hmean

harmonic\_mean = stats.hmean(data)

# Geometric Mean

geometric\_mean = stats.gmean(data)

# Standard Deviation

standard\_deviation = data.std()

# Coefficient of Variation

coefficient\_of\_variation = standard\_deviation / mean\_value

# Skewness

skewness = data.skew()

# Kurtosis

kurtosis = data.kurtosis()

# Data Quality

dq = harmonic\_mean / mean\_value

return mean\_value, median\_value, mode\_value, harmonic\_mean, geometric\_mean, standard\_deviation, coefficient\_of\_variation, skewness, kurtosis, dq

def read\_excel(file\_path):

# Read excel file

df = pd.read\_excel(file\_path)

# Data is in a column named 'close'

data = df['close']

return data

sbiData = read\_excel('SBI.xlsx')

sbiMean, sbiMode, sbiMedian, sbiHmean, sbiGmean, sbiStd, sbiCv, sbiSkew, sbiKurt, sbidq = calculate\_statistics(sbiData)

axisbankData = read\_excel('AXISBANK.xlsx')

axisMean, axisMode, axisMedian, axisHmean, axisGmean, axisStd, axisCv, axisSkew, axisKurt, axisdq = calculate\_statistics(axisbankData)

hdfcData = read\_excel('HDFCBANK.xlsx')

hdfcMean, hdfcMode, hdfcMedian, hdfcHmean, hdfcGmean, hdfcStd, hdfcCv, hdfcSkew, hdfcKurt, hdfcdq = calculate\_statistics(hdfcData)

tcsData = read\_excel('TCS.xlsx')

tcsMean, tcsMode, tcsMedian, tcsHmean, tcsGmean, tcsStd, tcsCv, tcsSkew, tcsKurt, tcsdq = calculate\_statistics(tcsData)

wiproData = read\_excel('WIPRO.xlsx')

wiproMean, wiproMode, wiproMedian, wiproHmean, wiproGmean, wiproStd, wiproCv, wiproSkew, wiproKurt, wiprodq = calculate\_statistics(wiproData)

hclData = read\_excel('HCLTECH.xlsx')

hclMean, hclMode, hclMedian, hclHmean, hclGmean, hclStd, hclCv, hclSkew, hclKurt, hcldq = calculate\_statistics(hclData)

print("SBI, AXISBANK, HDFC, TCS, WIPRO, HCLTECH")

print("Mean: ", round(sbiMean, 2), round(axisMean, 2), round(hdfcMean, 2), round(tcsMean, 2), round(wiproMean, 2), round(hclMean, 2))

print("Mode: ", round(sbiMode, 2), round(axisMode, 2), round(hdfcMode, 2), round(tcsMode, 2), round(wiproMode, 2), round(hclMode, 2))

print("Median: ", round(sbiMedian, 2), round(axisMedian, 2), round(hdfcMedian, 2), round(tcsMedian, 2), round(wiproMedian, 2), round(hclMedian, 2))

print("Harmonic Mean: ", round(sbiHmean, 2), round(axisHmean, 2), round(hdfcHmean, 2), round(tcsHmean, 2), round(wiproHmean, 2), round(hclHmean, 2))

print("Geometric Mean: ", round(sbiGmean, 2), round(axisGmean, 2), round(hdfcGmean, 2), round(tcsGmean, 2), round(wiproGmean, 2), round(hclGmean, 2))

print("\n")

print("Standard Deviation: ", round(sbiStd, 2), round(axisStd, 2), round(hdfcStd, 2), round(tcsStd, 2), round(wiproStd, 2), round(hclStd, 2))

print("Coefficient of Variation: ", round(sbiCv, 2), round(axisCv, 2), round(hdfcCv, 2), round(tcsCv, 2), round(wiproCv, 2), round(hclCv, 2))

print("Skewness: ", round(sbiSkew, 2), round(axisSkew, 2), round(hdfcSkew, 2), round(tcsSkew, 2), round(wiproSkew, 2), round(hclSkew, 2))

print("Kurtosis: ", round(sbiKurt, 2), round(axisKurt, 2), round(hdfcKurt, 2), round(tcsKurt, 2), round(wiproKurt, 2), round(hclKurt, 2))

print("Data Quality", sbidq, axisdq, hdfcdq, tcsdq, wiprodq, hcldq)

# SBI, AXISBANK, HDFC, TCS, WIPRO, HCLTECH

# Mean: 569.83 1000.38 1492.02 3397.43 386.24 1274.52

# Mode: 569.25 996.22 1491.52 3374.5 382.8 1268.18

# Median: 547.0 955.45 1463.4 3330.65 378.25 1229.0

# Harmonic Mean: 569.62 999.82 1491.89 3396.18 386.1 1273.91

# Geometric Mean: 569.72 1000.1 1491.95 3396.8 386.17 1274.21

# Standard Deviation: 11.05 24.11 14.57 67.43 7.42 28.89

# Coefficient of Variation: 0.02 0.02 0.01 0.02 0.02 0.02

# Skewness: -0.33 -0.07 -0.16 1.06 1.14 0.62

# Kurtosis: -0.97 -1.03 -0.69 -0.39 -0.31 -0.16

# Data Quality 0.99963891201744 0.999444506663834 0.9999088193947605 0.9996309451541434 0.9996541688601999 0.9995153534656276

**Inference:**

**Central Tendency Measures:**

z

**Dispersion Measures:**

Standard deviation values are provided for each company, indicating the amount of variability or dispersion in the stock prices.

All companies have relatively low standard deviations, suggesting that the stock prices are generally stable.

**Shape of the Distribution:**

Skewness is negative for SBI, AXISBANK, HDFC, and WIPRO, indicating a slight leftward skewness in their distributions.

TCS and HCLTECH have positive skewness, suggesting a slight rightward skewness in their distributions.

Negative kurtosis values for all companies imply that the distributions are platykurtic, meaning they have lighter tails and are less peaked than a normal distribution.

**Stock Price Relationships:**

TCS has the highest mean, median, mode, harmonic mean, and geometric mean among the listed companies, suggesting relatively higher stock prices on average.

WIPRO has the lowest mean, median, mode, harmonic mean, and geometric mean, indicating comparatively lower stock prices on average.

**Volatility:**

TCS has the highest standard deviation, indicating higher volatility in its stock prices compared to other companies.

HCLTECH has the lowest standard deviation, suggesting lower volatility in its stock prices.

**Data Quality:**

Data Quality is above 95% for all of the stocks. Which shows that the quality of data is good.

**Tool/Language:**

Jupyter Notebook

R-studio

**References:**

<https://www.nseindia.com/get-quotes/equity?symbol=>

|  |  |
| --- | --- |
| Exp No: 3 | One Sample T test  Case Study1: Reliance Mart  Case Study 2: Hypothetical Data (Score) |
| Date: 27/11/23 |

**Aim:**

The aim of conducting a One Sample T-Test for Reliance Mart is to assess whether the mean of a specific variable (e.g., customer satisfaction scores, average transaction value, etc.) within the Reliance Mart dataset significantly differs from a predetermined benchmark or industry standard.

**Objective:**

The objective of conducting a One Sample T-Test for Reliance Mart is to statistically evaluate whether a specific metric, such as customer satisfaction scores or average transaction value, significantly differs from a predetermined benchmark or industry standard. By rigorously analyzing the mean of the chosen variable, this objective aims to provide actionable insights for Reliance Mart stakeholders. The findings will guide decision-makers in understanding the performance of key operational aspects, facilitating targeted improvements to enhance overall business success and customer experience.

**Program Code:**

**Python Code:**

import pandas as pd

from scipy.stats import ttest\_ind

def calculate\_statistics(data):

# T-test

t\_statistic, p\_value = ttest\_ind(data, [0], alternative='two-sided')

return t\_statistic, p\_value

def read\_excel(file\_path):

# Read excel file

df = pd.read\_excel(file\_path)

# Data is in a column named 'close'

data = df['Rice\_Bag\_Weight']

return data

hclData = read\_excel('RelianceMart.xlsx')

t\_stat, p\_value = calculate\_statistics(hclData)

print("T-Statistic: ", t\_stat)

print("P-Value: ", p\_value)

**Output:**

T-Statistic: -5.2369341

P-Value: 6.796453274044647e-27

**R Code:**

library("xlsx")

library("e1071")

# Read excel file

df <- read.xlsx("C:/Users/admin/Downloads/RelianceMart.xlsx", sheetName = "Sheet1")

# Data is in a column named 'Data'

data <- df$Rice\_Bag\_Weight

# Perform t-test

t\_test\_result <- t.test(data)

# Extract t-value and p-value

t\_value <- t\_test\_result$statistic

p\_value <- t\_test\_result$p.value

print\_statistics <- function(t\_value, p\_value) {

cat(paste("t-value:", t\_value, "\n"))

cat(paste("p-value:", p\_value, "\n"))

}

# Call the function with the statistics

print\_statistics(t\_value, p\_value)

**Output:**

t-value: -5.2369341

p-value: 1.74248682332357e-47

**Inference:**

The computed t-statistic of 42.21 is notably large, indicating a substantial difference between the groups being compared. The extremely low p-value of 6.80e-27 provides strong evidence against the null hypothesis, suggesting that the observed differences are statistically significant. This result implies a robust rejection of the null hypothesis, underscoring the significance of the observed effect or relationship.

**Tool/ Language:**

R Studio

Jupyter Notebook

|  |  |
| --- | --- |
| Exp No: 4 | Two Sample T test (Paired)  Case Study1: Pre and Post Test Score  Case Study 2: Crocin Tablet Temperature dataset |
| Date:28/11/23 |

**Aim:**

1. Evaluate the effectiveness of an educational intervention by conducting a paired Two Sample T test on pre and post-test scores, aiming to determine if there is a statistically significant difference in the academic performance of participants.

2. Investigate the impact of Crocin tablets on body temperature by employing a paired Two Sample T test on the temperature dataset. The aim is to assess whether there is a significant change in body temperature before and after the administration of Crocin tablets, providing insights into the tablet's potential therapeutic effects.

**Objective:**

1. Objective for Pre and Post Test Score Study:

- To statistically assess the effectiveness of the educational intervention by conducting a paired Two Sample T test on pre and post-test scores, aiming to quantify and validate any significant improvements in academic performance among participants.

2. Objective for Crocin Tablet Temperature Dataset Study:

- To determine the impact of Crocin tablets on body temperature through a paired Two Sample T test, with the objective of identifying and confirming any statistically significant changes in temperature post-administration, thereby contributing to the understanding of the tablets' potential therapeutic effects.

**Program Code:**

**Python Code 1:**

import pandas as pd

from scipy.stats import ttest\_rel

def calculate\_statistics(pre, post):

    # T-test

    t\_statistic, p\_value = ttest\_rel(pre, post)

    return t\_statistic, p\_value

def read\_excel(file\_path):

    # Read excel file

    df = pd.read\_excel(file\_path)

    # Data is in a column named 'close'

    pre = df['Pre\_Score']

    post = df['Post\_Score']

    return pre, post

pre, post = read\_excel('Pre\_Post\_Score.xlsx')

t\_stat, p\_value = calculate\_statistics(pre, post)

print("T-Statistic: ", t\_stat)

print("P-Value: ", p\_value)

**Output 1:**

****

**R Code 1:**

library("xlsx")

library("e1071")

# Read excel file

df <- read.xlsx("C:/Users/admin/Downloads/Pre\_Post\_Score.xlsx", sheetName = "Sheet1")

# Data is in a column named 'Data'

pre <- df$Pre\_Score

post <- df$Post\_Score

# Perform t-test

t\_test\_result <- t.test(pre, post, paired = TRUE)

# Extract t-value and p-value

t\_value <- t\_test\_result$statistic

p\_value <- t\_test\_result$p.value

print\_statistics <- function(t\_value, p\_value) {

    cat(paste("t-value:", t\_value, "\n"))

    cat(paste("p-value:", p\_value, "\n"))

}

# Call the function with the statistics

print\_statistics(t\_value, p\_value)

**Output 1:**

****

**Python Code 2:**

import pandas as pd

from scipy.stats import ttest\_rel

def calculate\_statistics(pre, post):

    # T-test

    t\_statistic, p\_value = ttest\_rel(pre, post)

    return t\_statistic, p\_value

def read\_excel(file\_path):

    # Read excel file

    df = pd.read\_excel(file\_path)

    # Data is in a column named 'close'

    pre = df['Before\_Crocin']

    post = df['After\_Crocin']

    return pre, post

pre, post = read\_excel('Crocin\_Data.xlsx')

t\_stat, p\_value = calculate\_statistics(pre, post)

print("T-Statistic: ", t\_stat)

print("P-Value: ", p\_value)

**Output 2:**

****

**R Code 2:**

library("xlsx")

library("e1071")

# Read excel file

df <- read.xlsx("C:/Users/admin/Downloads/Crocin\_Data.xlsx", sheetName = "Sheet1")

# Data is in a column named 'Data'

pre <- df$Before\_Crocin

post <- df$After\_Crocin

# Perform t-test

t\_test\_result <- t.test(pre, post, paired = TRUE)

# Extract t-value and p-value

t\_value <- t\_test\_result$statistic

p\_value <- t\_test\_result$p.value

print\_statistics <- function(t\_value, p\_value) {

    cat(paste("t-value:", t\_value, "\n"))

    cat(paste("p-value:", p\_value, "\n"))

}

# Call the function with the statistics

print\_statistics(t\_value, p\_value)

**Output 2:**

****

**Inference:**

Case 1 :The calculated t-value of -3.23 suggests a significant deviation from the null hypothesis. With a low p-value of 0.0044, there is strong evidence to reject the null hypothesis, indicating a statistically significant effect or relationship in the analyzed data.

Case 2: The t-value of 7.07 indicates a substantial difference from the null hypothesis, suggesting a significant effect in the analyzed data. The extremely low p-value (1.08e-07) underscores the strength of this evidence, providing strong support for rejecting the null hypothesis and implying a statistically significant result.

**Tool/ Language:**

R-Studio

VS Code