####################### Module 3 Assignment #######################################

##Get Current Working Directory

getwd()

##Set working Directory to current folder

setwd('~/Module\_3\_Assignment/')

###Load and Install Required Packages

install.packages('readxl')

install.packages('modeest')

install.packages('moments')

install.packages('dplyr')

install.packages("ggplot2")

install.packages("plotly")

install.packages("tibble")

install.packages("tibble")

install.packages("tibble")

install.packages("BSDA")

library(readxl)

library(modeest)

library(moments)

library(dplyr)

library(ggplot2)

library(plotly)

library(tibble)

library(BSDA)

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 1:Import the data to check its class and structure and display the head and tail of the data

#Import data

input\_df <- read\_excel('Input\_data.xlsx')

print('------------------------')

#Check the class of the data frame

class(input\_df)

print('------------------------')

#Check the structure of the data frame

str(input\_df)

print('------------------------')

#Print the head of dataframe

head(input\_df)

print('------------------------')

#Print the tail of dataframe

tail(input\_df)

print('------------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 2: Calculate variaous parameters (listed below)

#a.Difference in the means of the pre and post variables

diff\_mean = mean(input\_df$Pre, na.rm = T) - mean(input\_df$Post,na.rm = T)

print(diff\_mean)

print('------------------------')

#b.Values that divide the pre and post variable data into equal halves : MEDIAN

pre\_median = median(input\_df$Pre, na.rm = T)

print(pre\_median)

print('------------------------')

post\_median = median(input\_df$Post, na.rm = T)

print(post\_median)

print('------------------------')

#c.Mode for the pre variable

pre\_mode <- mfv(input\_df$Pre)

print(pre\_mode)

print('------------------------')

#d.First and third quantile for the pre and post variables

pre\_q1 <- quantile(input\_df$Pre, 0.25)

print(pre\_q1)

print('------------------------')

pre\_q3 <- quantile(input\_df$Pre, 0.75)

print(pre\_q3)

print('------------------------')

post\_q1 <- quantile(input\_df$Post, 0.25)

print(post\_q1)

print('------------------------')

post\_q3 <- quantile(input\_df$Post, 0.75)

print(post\_q3)

print('------------------------')

#e.Range of the pre and post variables

pre\_range <- range(input\_df$Pre)

print(pre\_range)

print('------------------------')

post\_range <- range(input\_df$Post)

print(post\_range)

print('------------------------')

#f.Variance and standard deviation for the pre and post variables

pre\_variance <- var(input\_df$Pre)

print(pre\_variance)

print('------------------------')

post\_variance <- var(input\_df$Post)

print(post\_variance)

print('------------------------')

pre\_sd <- sd(input\_df$Pre)

print(pre\_sd)

print('------------------------')

post\_sd <- sd(input\_df$Post)

print(post\_sd)

print('------------------------')

#g.Coefficient of variation and mean absolute deviation for the pre and post variables

pre\_cv <- sd(input\_df$Pre) / mean(input\_df$Pre) \* 100

print(pre\_cv)

print('------------------------')

post\_cv <- sd(input\_df$Post) / mean(input\_df$Post) \* 100

print(post\_cv)

print('------------------------')

pre\_mad <- mean(abs(input\_df$Pre - mean(input\_df$Pre)))

print(pre\_mad)

print('------------------------')

post\_mad <- mean(abs(input\_df$Post - mean(input\_df$Post)))

print(post\_mad)

print('------------------------')

#h.Interquartile range of the pre and post variables

pre\_iqr <- pre\_q3 - pre\_q1

print(pre\_iqr)

print('------------------------')

post\_iqr <- post\_q3 - post\_q1

print(post\_iqr)

print('------------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 3:Measure the skewness for pre and post variables and apply the Agostino test to check the skewness

# Calculate the skewness for the Pre variable

skewness\_pre <- skewness(input\_df$Pre)

print(skewness\_pre)

print('------------------------')

# Calculate the skewness for the Post variable

skewness\_post <- skewness(input\_df$Post)

print(skewness\_post)

print('------------------------')

# Apply the Agostino test to the Pre variable

agostino\_test\_pre <- agostino.test(input\_df$Pre)

print(agostino\_test\_pre)

print('------------------------')

# Apply the Agostino test to the Post variable

agostino\_test\_post <- agostino.test(input\_df$Post)

print(agostino\_test\_post)

print('------------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 4:Identify the nature of distribution through kurtosis for both pre and post variables and confirm the result through the Anscombe test

# Calculate the kurtosis for the Pre variable

kurtosis\_pre <- kurtosis(input\_df$Pre)

print(kurtosis\_pre)

print('------------------------')

# Calculate the kurtosis for the Post variable

kurtosis\_post <- kurtosis(input\_df$Post)

print(kurtosis\_post)

print('------------------------')

# Apply the Anscombe test to the Pre variable

anscombe\_test\_pre <- anscombe.test(input\_df$Pre)

print(anscombe\_test\_pre)

print('------------------------')

# Apply the Anscombe test to the Post variable

anscombe\_test\_post <- anscombe.test(input\_df$Post)

print(anscombe\_test\_post)

print('------------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 5:Plot a graph to check the skewness and peakedness in the distribution of pre and post variables

# Plot the distribution of the pre variable

plot(density(input\_df$Pre), main="Distribution of Pre Variable",

xlab="Value", ylab="Density",

sub=paste("Skewness:", skewness\_pre, "Kurtosis:", kurtosis\_pre))

print('-----------------------')

# Plot the distribution of the post variable

plot(density(input\_df$Post), main="Distribution of Post Variable",

xlab="Value", ylab="Density",

sub=paste("Skewness:", skewness\_post, "Kurtosis:", kurtosis\_post))

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 6:Compute the frequency and relative frequency for each brand of cold drink

# Calculate the frequency and relative frequency for each brand

df\_grouped <- input\_df %>%

group\_by(input\_df$`Cold-Drink`) %>%

summarise(

frequency = n(),

relative\_frequency = n() / dim(input\_df)[1]

)

# Rename the column

df\_grouped <- df\_grouped %>%

rename(cold\_drink = `input\_df$\`Cold-Drink\``)

print(df\_grouped)

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 7:Create a pie chart and bar chart to show the preferences of the cold drinks available and provide the necessary labels

# Create a pie chart

pie\_chart <- df\_grouped %>%

plot\_ly(labels = ~cold\_drink,values = ~frequency, type = "pie") %>%

layout(title = "Preferences of Cold Drinks",

xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),

yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

pie\_chart

# Create a bar chart

bar\_chart = plot\_ly(data = df\_grouped,

x = ~cold\_drink,

y = ~frequency,

type = "bar"

)

# Print the bar chart

bar\_chart

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 8:Plot a density graph on the cold-drink frequency and comment on the skewness and kurtosis

# Plot the distribution of the cold-drink frequency

plot(density(df\_grouped$frequency), main="Distribution of cold-drink frequency",

xlab="Value", ylab="Density")

print('-----------------------')

kurtosis\_value = kurtosis(df\_grouped$frequency)

# Comment on Skewness and Kurtosis

print("The plot looks bit drifted toward right, hence it indicates that the distribution is slightly skewed to the right.")

print(paste("The kurtosis is", kurtosis\_value, ". This indicates that the distribution is slightly leptokurtic."))

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 9:Convert the ‘Status’, ‘Rating’, and ‘Outlook’ variables into factor types and summarize them

# Convert the variable to factor type

input\_df$Status <- as.factor(input\_df$Status)

input\_df$Rating <- as.factor(input\_df$Rating)

input\_df$Outlook <- as.factor(input\_df$Outlook)

# summarize above fvariables

summary(input\_df$Status)

print('-----------------------')

summary(input\_df$Rating)

print('-----------------------')

summary(input\_df$Outlook)

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 10: Calculate the difference in the average pre-training satisfaction ratings of member and observer status and for the post-training member and observer status

# Group the data by status

status\_grouped\_data <- input\_df %>% group\_by(Status)

# Calculate the mean of Pre and Post satisfaction ratings

mean\_pre <- status\_grouped\_data %>% summarise(mean\_pre = mean(Pre))

mean\_post <- status\_grouped\_data %>% summarise(mean\_post = mean(Post))

# Combine the mean of Pre and Post satisfaction ratings into a single data frame

status\_results <- merge(mean\_pre, mean\_post, by = "Status")

# Calculate the difference between observer and member values

difference\_mean\_pre <- status\_results$mean\_pre[2] - status\_results$mean\_pre[1]

difference\_mean\_post <- status\_results$mean\_post[2] - status\_results$mean\_post[1]

# Print the difference

print(difference\_mean\_pre)

print('-----------------------')

print(difference\_mean\_post)

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 11:Compute the average pre-satisfaction and post-satisfaction ratings of employees with a ‘Stable’ Outlook

# Filter the data for employees with a Stable Outlook

data\_stable <- input\_df %>% filter(toupper(Outlook) == "STABLE")

# Calculate the average pre-satisfaction rating

mean\_pre\_stable <- mean(data\_stable$Pre)

# Calculate the average post-satisfaction rating

mean\_post\_stable <- mean(data\_stable$Post)

# Print the results

print(paste("The average pre-satisfaction rating for employees with a Stable Outlook is", mean\_pre\_stable))

print(paste("The average post-satisfaction rating for employees with a Stable Outlook is", mean\_post\_stable))

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 12: Construct a confidence interval at a 2.5%, 5%, and 1% level of significance for the salary variable

# Calculate the sample mean and standard deviation of the salary variable

mean\_salary <- mean(input\_df$Salary)

sd\_salary <- sd(input\_df$Salary)

# Calculate the confidence intervals

ci\_2.5 <- mean\_salary + c(-1.96, 1.96) \* sd\_salary

ci\_5 <- mean\_salary + c(-1.645, 1.645) \* sd\_salary

ci\_1 <- mean\_salary + c(-2.576, 2.576) \* sd\_salary

# Print the confidence intervals

print(paste("The mean salary is :",mean\_salary))

print('-----------------------')

print(paste("The 95% confidence interval for the salary variable is", ci\_5[1],"to", ci\_5[2] ))

print('-----------------------')

print(paste("The 90% confidence interval for the salary variable is", ci\_2.5[1],"to",ci\_2.5[2]))

print('-----------------------')

print(paste("The 99% confidence interval for the salary variable is", ci\_1[1],"to",ci\_1[2]))

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 13:Construct a 99%, 95%, and 90% confidence interval estimate for the pre and post variables

# Calculate the mean and standard deviation of the pre variable

mean\_pre <- mean(input\_df$Pre)

std\_dev\_pre <- sd(input\_df$Pre)

# Calculate the confidence intervals

ci\_99\_pre <- mean\_pre + c(-2.576, 2.576) \* std\_dev\_pre

ci\_95\_pre <- mean\_pre + c(-1.96, 1.96) \* std\_dev\_pre

ci\_90\_pre <- mean\_pre + c(-1.645, 1.645) \* std\_dev\_pre

#Print the confidence Interval for Pre Variable

print(paste("The mean salary of Pre variable is :",mean\_pre))

print('-----------------------')

print(paste("The 99% confidence interval for the pre variable is", ci\_99\_pre[1],"to", ci\_99\_pre[2]))

print('-----------------------')

print(paste("The 95% confidence interval for the pre variable is", ci\_95\_pre[1],"to", ci\_95\_pre[2]))

print('-----------------------')

print(paste("The 90% confidence interval for the pre variable is", ci\_90\_pre[1],"to", ci\_90\_pre[2]))

print('-----------------------')

# Calculate the mean and standard deviation of the post variable

mean\_post <- mean(input\_df$Post)

std\_dev\_post <- sd(input\_df$Post)

# Calculate the confidence intervals

ci\_99\_post <- mean\_post + c(-2.576, 2.576) \* std\_dev\_post

ci\_95\_post <- mean\_post + c(-1.96, 1.96) \* std\_dev\_post

ci\_90\_post <- mean\_post + c(-1.645, 1.645) \* std\_dev\_post

#Print the confidence Interval for Post Variable

print(paste("The mean salary of Post variable is :",mean\_post))

print('-----------------------')

print(paste("The 99% confidence interval for the post variable is", ci\_99\_post[1],"to", ci\_99\_post[2]))

print('-----------------------')

print(paste("The 95% confidence interval for the post variable is", ci\_95\_post[1],"to", ci\_95\_post[2]))

print('-----------------------')

print(paste("The 90% confidence interval for the post variable is", ci\_90\_post[1],"to", ci\_90\_post[2]))

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 14:Solve the below tasks:

# Task 14a:Take a sample of 50 observations from the pre and post dataset (without replacement)

# Create a sample of 50 observations from the pre variable

sample\_pre <- sample(input\_df$Pre, 50, replace = T)

# Create a sample of 50 observations from the post variable

sample\_post <- sample(input\_df$Post, 50, replace = T)

# Combine the two samples into a single data frame

sample <- cbind(sample\_pre, sample\_post)

sample\_df <- as.data.frame(sample)

# Print the sample

print(sample\_df)

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

# Task 14b:Construct a null hypothesis to examine whether the sample (50 observations) mean score of pre and post variables is significantly different from the population (1000 observations)

# Calculate the mean of the pre variable in the population

mean\_pre\_population <- mean(input\_df$Pre)

# Calculate the mean of the post variable in the population

mean\_post\_population <- mean(input\_df$Post)

# Calculate the mean of the pre variable in the sample

mean\_pre\_sample <- mean(sample\_df$sample\_pre)

# Calculate the mean of the post variable in the sample

mean\_post\_sample <- mean(sample\_df$sample\_post)

# Construct the null hypothesis

# H0: There is no significant difference between the mean score of pre and post variables in the population and the sample

print("H0: There is no significant difference between the mean score of pre and post variables in the population and the sample")

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

# Task 14c:Compute corresponding Z values for pre and post variables in the sample

# Calculate the mean of the pre variable in the sample

mean\_pre\_sample <- mean(sample\_df$sample\_pre)

# Calculate the standard deviation of the pre variable in the sample

std\_dev\_pre\_sample <- sd(sample\_df$sample\_pre)

# Calculate the z-score for the pre variable

z\_pre <- (sample\_df$sample\_pre - mean\_pre\_sample) / std\_dev\_pre\_sample

# Calculate the mean of the post variable in the sample

mean\_post\_sample <- mean(sample\_df$sample\_post)

# Calculate the standard deviation of the post variable in the sample

std\_dev\_post\_sample <- sd(sample\_df$sample\_post)

# Calculate the z-score for the post variable

z\_post <- (sample\_df$sample\_post - mean\_post\_sample) / std\_dev\_post\_sample

# Print the z-scores

print(z\_pre)

print(z\_post)

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 15: Using the p-value method, determine whether the sample mean for the pre and post variables differs significantly from the population mean at the 10% significance level

###Starting with Z Test Hypothesis Testing####

# Calculate the mean of the pre variable in the population

mean\_pre\_population <- mean(input\_df$Pre)

# Calculate the sd of the pre variable in the population

sd\_pre\_population <- sd(input\_df$Pre)

# Calculate the mean of the post variable in the population

mean\_post\_population <- mean(input\_df$Post)

# Calculate the sd of the pre variable in the population

sd\_post\_population <- sd(input\_df$Post)

z.test(sample\_pre, sample\_post, alternative = 'two.sided',

conf.level = 0.90, sigma.x = sd\_pre\_population, sigma.y = sd\_post\_population)

p\_value\_pre\_post <- z.test(sample\_pre, sample\_post, alternative = 'two.sided',

conf.level = 0.90, sigma.x = sd\_pre\_population, sigma.y = sd\_post\_population)$p.value

print(p\_value\_pre\_post)

print('-----------------------')

print(paste("Observation based on Results:", "As the p-value is < 0.05 it clearly states that

there is a significant difference in mean of 2 categories. The same can be validated

by looking at the mean of Pre and Post sample"))

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 16:        Calculate the critical Z value for the 10% level of significance and the decision rule using the critical value approach

###Starting with Z Test Hypothesis Testing####

z.test(sample\_pre, sample\_post, alternative = 'two.sided',

conf.level = 0.90, sigma.x = sd\_pre\_population, sigma.y = sd\_post\_population)

critical\_z\_value = qnorm(p = 0.1/2, lower.tail = T) ## As Z-value is negative

print(critical\_z\_value)

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 17: Compute the T-statistics value for the pre and post variables

###Hypothesis Testing using T-Distribution###

t.test(sample\_pre, sample\_post, alternative = 'two.sided', conf.level = 0.90)

t.test(sample\_pre, sample\_post, alternative = 'greater', conf.level = 0.90)

t.test(sample\_pre, sample\_post, alternative = 'less', conf.level = 0.90)

print(paste(" Difference in Post and Pre sample mean: ",mean\_post\_sample - mean\_pre\_sample))

print('-----------------------')

print("The above three test confirms that there is a significant difference in Pre and Post Training

the p-value is quite significant in two-sample test, suggesting there is a statistically significant

difference is score of Pre and post survey. And the One-tail test confirms that the post sample mean is better than Pre-sample mean")

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 18:        Calculate the p-value and the decision using the p-value approach for pre and post variables at a 10% level of significance

p\_value\_pre <- t.test(sample\_pre, alternative = 'two.sided',

conf.level = 0.90)$p.value

print(p\_value\_pre)

print('-----------------------')

p\_value\_post <- t.test(sample\_post, alternative = 'two.sided',

conf.level = 0.90)$p.value

print(p\_value\_post)

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##

## Task 19:Calculate the critical T value for the level of significance of 10% and the decision rule using the critical value approach

###Hypothesis Testing using T-Distribution###

t.test(sample\_pre, sample\_post, alternative = 'two.sided', conf.level = 0.90)

DOF <- length(sample\_pre) - 1

print(DOF)

print('-----------------------')

critical\_t\_value <- qt(p = 0.1/2, df = DOF, lower.tail = T)## As t-value is negative

print(critical\_t\_value)

print('-----------------------')

##\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*##