**MISSING VALUES**

**1)**

############## Load necessary packages###########

install.packages("moments")

install.packages("dplyr")

install.packages("mice")

library(dplyr)

library(moments)

library(mice)

############### Load data #################

df<-read.csv("/Users/krutikadeshmukh/Downloads/BodySystemTrainTable 2.csv")

head(df)

########create a 10 % sample #######

set.seed(123) ###for reproductibility

sample\_df<-df %>% sample\_frac(0.10)

sum(is.na(sample\_df)) ##[1] 2575026

##Remove empty column

sample\_df<-sample\_df[,!sapply(sample\_df,function(col)all(is.na(col)))]

####convert all the "NUll" string values to actual NA values

sample\_df<-sample\_df|>mutate\_all(~ifelse(.=="NULL",NA,.))

sum(is.na(sample\_df)) ##[1] 2368725

################################# Question1 #########################

#Remove Missing values for the 10% sample and report thr number of cases that remain

sample\_df\_cleaned<-sample\_df %>% filter(!is.na(sample\_df$bs1lr))

nrow(sample\_df\_cleaned) ##[1] 33460

################################Question2 ################################################

## plot progression and check distribution

hist(sample\_df\_cleaned$bs1lr, main = "Progression Distribution", xlab = "Progression Score")

qqnorm(sample\_df\_cleaned$bs1lr)

qqline(sample\_df\_cleaned$bs1lr)

## the data is not bimodal then check if it is symmetric

# Calculate mean and median

mean1<-mean(sample\_df\_cleaned$bs1lr)

median1<-median(sample\_df\_cleaned$bs1lr)

cat("Mean",mean1, "\n") ##Mean 1.585994

cat("Median",median1, "\n") #Median 1.7237061

##The mean is less than the median,it suggests that the distribution is negatively skewed.

##The skewness is -1.007672,and it is negative

##quantile transformation to improve QQplot

quantile\_transformed\_df <- qnorm(ppoints(length(sample\_df\_cleaned$bs1lr)))

hist(quantile\_transformed\_df,main="quantile-Treansformed Data",

xlab="quantile\_transformed Values")

qqnorm(quantile\_transformed\_df, main="quantile-Transformed Data",

xlab="quantile-Transformed values")

qqline(quantile\_transformed\_df, col = 2,lty = 2)

mean4<-mean(quantile\_transformed\_df)

median4<-median(quantile\_transformed\_df)

cat("Mean",mean4, "\n") ##Mean is -1.373322e-17

cat("Median",median4, "\n") ##Median -6.95719e-17

skewness4<-skewness(quantile\_transformed\_df)

cat("Skewness", skewness4,"\n") ##Skewness 8.88656e-16

###################################Question3 #####################

# Explore pairwise and Triplets of Independent variables

# Create all possible combinations of independent variables

combos\_pairs<-combn(names(sample\_df\_cleaned)[3:ncol(sample\_df\_cleaned)],2)

combos\_triplets<-combn(names(sample\_df\_cleaned)[3:ncol(sample\_df\_cleaned)],3)

all\_vars<-c(combos\_pairs[1,],combos\_pairs[2,],

combos\_triplets[2,],

names(sample\_df\_cleaned)[3:ncol(sample\_df\_cleaned)])

always\_missing<-names(sample\_df\_cleaned)[sapply(sample\_df\_cleaned[,names(sample\_df\_cleaned) %in% all\_vars], function(x)

all(is.na(x)))]

selected\_vars<-setdiff(all\_vars,always\_missing)

total\_vars<-length(selected\_vars)

cat("Total number of independent variables after excluding always missing", total\_vars)

# Total number of independent variables after excluding always missing: 20

###################################### Question 4 ####################

# Specify the variables for imputation

vars\_to\_impute <- c("bs1lr", "bs2lr", "bs3lr", "bs4lr", "bs5lr", "bs6lr",

"bs7lr", "bs8lr", "bs9lr", "bs10lr", "bs11lr", "bs12lr",

"bs13lr", "bs14lr", "bs16lr", "bs17lr", "bs18lr", "bs19lr")

# Set the number of imputations

num\_imputations <- 5

# Create multiple imputations

imp\_data <- mice::mice(sample\_df\_cleaned[, vars\_to\_impute],

m = num\_imputations, method = "pmm")

summary(imp\_data)

#Create an empty list to store regression results

regression\_results <- list()

#Fit the linear regression model for each imputed dataset

for (i in 1:num\_imputations) {

# Fit the linear regression model for "bs1lr" as the dependent variable

model <- lm(bs1lr ~ bs2lr + bs3lr + bs4lr + bs5lr + bs6lr +

bs7lr + bs8lr + bs9lr + bs10lr + bs11lr + bs12lr + bs13lr + bs14lr +

bs16lr + bs17lr + bs18lr + bs19lr,

data = mice::complete(imp\_data, action = i))

#store the summary of the model in the list

regression\_results[[i]] <- summary(model)

}

#Calculate and print the average R-squared across imputations

avg\_rsquared <- mean(sapply(regression\_results, function(result) result$r.squared))

cat("Average percent of variation explained:", round(avg\_rsquared \* 100, 2), "%\n")

############################ Question 5 ###################

# Repalce missing independent variables with 0

sample\_df\_cleaned[is.na(sample\_df\_cleaned)]<-0

# Fit the linear regression model

model<-lm(bs1lr ~ bs2lr + bs3lr + bs4lr + bs5lr + bs6lr +

bs7lr + bs8lr + bs9lr + bs10lr + bs11lr + bs12lr +

bs13lr + bs14lr + bs16lr + bs17lr + bs18lr+bs19lr,

data = sample\_df\_cleaned)

summary(model)

# calculate R-squared, which is the percent of variation explained

rsquared<-summary(model)$r.squared

cat("Percent of variation explained:", round(rsquared \* 100, 2), "%\n")

# Percent of variation explained: 4.61 %

##################### Question 6 #######################

head(sample\_df\_cleaned)

# Drop the independent varaiables that no need be used and replace NA with mean

df\_for\_Q6 <- sample\_df\_cleaned %>%

select(-id, -dm, -TestTrain) %>%

mutate\_all(~ifelse(is.na(.), mean(., na.rm = TRUE), .))

# Fit the linear regression model

model<-lm(bs1lr ~ bs2lr + bs3lr + bs4lr + bs5lr + bs6lr +

bs7lr + bs8lr + bs9lr + bs10lr + bs11lr + bs12lr +

bs13lr + bs14lr + bs16lr + bs17lr + bs18lr+bs19lr,

data = df\_for\_Q6)

summary(model)

# Calculate R-squared, which is the percent of varaoiation explained

rsquared<-summary(model)$r.squared

cat("Percent of variation explained:", round(rsquared \* 100, 2), "%\n")

# Percent of variation explained: 4.61 %

############################ Question7 ######################

###Indicate which method of replacing missing values fits the data best.#####

##To check which method fits the data best, we can compare the performance of these methods based on some relevant criteria

##like comparing R-squared values obtained from the regression models.

# A highest R-squared value indicates that the independent variables explain more of the variation in the dependent variable.

### A low R-squared indicates that the model does not explain much of the variablity in the dependent variable

## It sugggests that the independent variables in the model are not strong predictors of the dependent variable

2)

install.packages("leaps")

library(readr)

library(leaps)

library(dplyr)

#Reading the data and sampling the dataset using 10%

data = read.csv("/Users/krutikadeshmukh/Downloads/BodySystemTrainTable.csv")

#Cleaning the dataset

#Dropping the column id

data <- data[, -1]

## Get a sample of the using 10% of the dataset

sample\_size <- floor(0.10 \* nrow(data)) #calculate 10% of the number of rows in df set seed(123) #setting seed for reproductibility

df\_sample <- data[sample(nrow (data), sample\_size), ]

## Create a new dataframe without the specified columns

df\_sample <- df\_sample [, -c(9,10,11,12,13,14,15,16,17,18,19,20,21,22)]

df\_sample = df\_sample

## Zero imputation

sample\_data\_imputed <- df\_sample

sample\_data\_imputed[is.na(sample\_data\_imputed)] <- 0 #Missing Values (NA) are being repplaced by 0

model <- lm(bs7lr ~ dm+ bs1lr + bs3lr + bs4lr + bs5lr +bs6lr, data = sample\_data\_imputed) #Fitting a linear regression model

par(mfrow = c(2, 2))

plot(model)

#Check the normal distribution assumption of the response variable

#Filter and remove the rows that are greater than or equal to 10

w <- sample\_data\_imputed %>%

filter(sample\_data\_imputed$bs7lr > 10 )

sample\_data\_imputed <- sample\_data\_imputed %>%

filter(sample\_data\_imputed$bs7lr < 10)

#Normalize the data using cuberoot transformation

#Apply the transformations and store them in new variables

sample\_data\_imputed$probability\_bs7lr <- sample\_data\_imputed$bs7lr / (1 + sample\_data\_imputed$bs7lr) #Odds to Probability

sample\_data\_imputed$log\_odds\_bs7lr <- 1 / (1 + exp(-sample\_data\_imputed$bs7lr)) #Log of Odds to Probability

sample\_data\_imputed$log\_bs7lr <- log(sample\_data\_imputed$bs7lr) #Logarithm Transformation

sample\_data\_imputed$bs7lr <- (sample\_data\_imputed$bs7lr)^1/3 #Cubed root Transformation

#Create histograms and overlay density plots for each transformation

par(mfrow = c(2, 2))

hist(sample\_data\_imputed$probability\_bs7lr, prob = TRUE, main = "Odds to Probability")

lines(density(sample\_data\_imputed$probability\_bs7lr), lwd = 2)

hist(sample\_data\_imputed$log\_odds\_bs7lr, prob =TRUE, main = "Logs of Odds to Probability")

lines(density(sample\_data\_imputed$log\_odds\_bs7lr), lwd = 2)

hist(sample\_data\_imputed$log\_bs7lr, prob = TRUE, main = "Logarithmic transformation")

lines(density(sample\_data\_imputed$log\_bs7lr), lwd = 2)

hist(sample\_data\_imputed$bs7lr, prob = TRUE, main = "Cubed Root Transformation")

lines(density(sample\_data\_imputed$bs7lr), lwd = 2 )

#Subset selection with 'regsubsets'

#regsubsets function used to fit multiple linear regression models with different subsets of predictor variables

epsilon <- 1e-6 #A small positive constant

sample\_data\_imputed$log\_bs7lr <- log(sample\_data\_imputed$bs7lr + epsilon)

model <- regsubsets(bs7lr~., data = sample\_data\_imputed[, -1], nvmax = 7)

#Summarize the results

summary(model)