library(matlib)

library(ggplot2)

library(rsample)

# to import x.csv data

X\_CSV\_DATA=as.matrix(read.csv(file="C:/softwarica/MSC assignment/X\_1673241366257.csv",header = F))

colnames(X\_CSV\_DATA)<-c("X1","X2","X3","X4")

# to import y data

Y\_CSV\_DATA=as.matrix(read.csv(file="C:/softwarica/MSC assignment/y\_1673241374123.csv",header = F))

colnames(Y\_CSV\_DATA)<-c("Y")

# to import time data

Time\_CSV\_DATA = read.csv("C:/softwarica/MSC assignment/time\_1673241270748.csv", header = F, skip = 1)

Time\_CSV\_DATA = as.matrix(rbind(0, Time\_CSV\_DATA))

# Task 1.1 : to plot time series

X\_CSV\_DATA.ts<-ts(X\_CSV\_DATA,start = c(min(Time\_CSV\_DATA),max(Time\_CSV\_DATA)),frequency =1)

Y\_CSV\_DATA.ts<-ts(Y\_CSV\_DATA,start = c(min(Time\_CSV\_DATA),max(Time\_CSV\_DATA)),frequency =1)

# to plot graph

plot(X\_CSV\_DATA.ts,main = "Time series plot of X Signal", xlab = "Time", ylab = "Input signal")

plot(Y\_CSV\_DATA.ts,main = "Time series plot of Y Signal", xlab = "Time", ylab = "Output signal")

# task 1.2 : For Distribution of each EEG signal

#Creating a density if X signal

density\_of\_X=density(X\_CSV\_DATA)

plot(density\_of\_X,main = "Density plot of whole input signal")

# to create a Histogram of X signal

hist(X\_CSV\_DATA,freq = FALSE,main = "Density")

# to add density in the histogram

lines(density\_of\_X,lwd=2,col="brown")

rug(jitter(X\_CSV\_DATA))

# to create a density if X1 signal

density\_of\_X1=density(X\_CSV\_DATA[,"X1"])

hist(X\_CSV\_DATA[,"X1"],freq = FALSE,main = "Histogram and density plot of X1",xlab = "X1 Signal")

lines(density\_of\_X1,lwd=2,col="brown")

# Add the data-points with noise in the X-axis

rug(jitter(X\_CSV\_DATA[,"X1"]))

# to create a density if X2 signal

density\_of\_X2=density(X\_CSV\_DATA[,"X2"])

hist(X\_CSV\_DATA[,"X2"],freq = FALSE,main = "Histogram and density plot of X2",xlab = "X2 Signal")

lines(density\_of\_X2,lwd=2,col="brown")

rug(jitter(X\_CSV\_DATA[,"X2"]))

# to create a density if X3 signal

density\_of\_X3=density(X\_CSV\_DATA[,"X3"])

hist(X\_CSV\_DATA[,"X3"],freq = FALSE,main = "Histogram and density plot of X3",xlab = "X3 Signal")

lines(density\_of\_X3,lwd=2,col="brown")

rug(jitter(X\_CSV\_DATA[,"X3"]))

#to create a density if X4 signal

density\_of\_X4=density(X\_CSV\_DATA[,"X4"])

hist(X\_CSV\_DATA[,"X4"],freq = FALSE,main = "Histogram and density plot of X4",xlab = "X4 Signal")

lines(density\_of\_X4,lwd=2,col="brown")

rug(jitter(X\_CSV\_DATA[,"X4"]))

#to create a density if Y signal

density\_of\_Y=density(Y\_CSV\_DATA)

plot(density\_of\_Y,main = "Density plot of Y",,xlab = "Output Signal")

hist(Y\_CSV\_DATA,freq = FALSE,main = "Histogram and density plot of Y",xlab = "Output Signal")

lines(density\_of\_Y,lwd=2,col="brown")

rug(jitter(Y\_CSV\_DATA))

# Task 1.3 For Correlation and scatter plots

# to arrange plot in a single screen

par(mfrow=c(2,2))

# to plot X1 against Y

plot(X\_CSV\_DATA[,"X1"],Y\_CSV\_DATA,main = "Correlation betweeen X1 and Y signal", xlab = "X1 signal", ylab = "Output signal")

# to plot X2 against Y

plot(X\_CSV\_DATA[,"X2"],Y\_CSV\_DATA,main = "Correlation betweeen X2 and Y signal", xlab = "X2 signal", ylab = "Output signal")

# to plot X3 against Y

plot(X\_CSV\_DATA[,"X3"],Y\_CSV\_DATA,main = "Correlation betweeen X3 and Y signal", xlab = "X3 signal", ylab = "Output signal")

# to plot X4 against Y

plot(X\_CSV\_DATA[,"X4"],Y\_CSV\_DATA,main = "Correlation betweeen X4 and Y signal", xlab = "X4 signal", ylab = "Output signal")

# Task 2

# For Calculating ones for binding the data

ones = matrix(1 , length(X\_CSV\_DATA)/4,1)

ones

# Task 2.1

# For Calculating thetahat of Model 1

# to Bind data from equation of model 1.

X\_model1<-cbind(ones,X\_CSV\_DATA[,"X4"],X\_CSV\_DATA[,"X1"]^2,X\_CSV\_DATA[,"X1"]^3,X\_CSV\_DATA[,"X2"]^4,X\_CSV\_DATA[,"X1"]^4)

X\_model1

# For Calculating thetahat of Model 1

Model1\_thetahat=solve(t(X\_model1) %\*% X\_model1) %\*% t(X\_model1) %\*% Y\_CSV\_DATA

Model1\_thetahat

#For model 2

# to Bind data from equation of model 2.

X\_model2<-cbind(ones,X\_CSV\_DATA[,"X4"],X\_CSV\_DATA[,"X1"]^3,X\_CSV\_DATA[,"X3"]^4)

X\_model2

# for Calculating thetahat of Model 2

Model2\_thetahat=solve(t(X\_model2) %\*% X\_model2) %\*% t(X\_model2) %\*% Y\_CSV\_DATA

Model2\_thetahat

#Model 3

# to Bind data from equation of model 3.

X\_model3<-cbind(ones,(X\_CSV\_DATA[,"X3"])^3,(X\_CSV\_DATA[,"X3"])^4)

X\_model3

# For Calculating thetahat of Model 3

Model3\_thetahat=solve(t(X\_model3) %\*% X\_model3) %\*% t(X\_model3) %\*% Y\_CSV\_DATA

Model3\_thetahat

#model 4

# to bind data from equation of model 3.

X\_model4<-cbind(ones,X\_CSV\_DATA[,"X2"],(X\_CSV\_DATA[,"X1"])^3,(X\_CSV\_DATA[,"X3"])^4)

X\_model4

# for Calculating thetahat of Model 4

Model4\_thetahat=solve(t(X\_model4) %\*% X\_model4) %\*% t(X\_model4) %\*% Y\_CSV\_DATA

Model4\_thetahat

# for Model 5

# to Bind data from equation of model 5.

X\_model5<-cbind(ones,(X\_CSV\_DATA[,"X4"]),(X\_CSV\_DATA[,"X1"])^2,(X\_CSV\_DATA[,"X1"])^3,(X\_CSV\_DATA[,"X3"])^4)

X\_model5

# for Calculating thetahat of model 1

Model5\_thetahat=solve(t(X\_model5) %\*% X\_model5) %\*% t(X\_model5) %\*% Y\_CSV\_DATA

Model5\_thetahat

# Task 2.2

# for Calculating Y-hat and RSS Model 1

Y\_hat\_model1 = X\_model1 %\*% Model1\_thetahat

Y\_hat\_model1

# for Calculating RSS

RSS\_Model\_1=sum((Y\_CSV\_DATA-Y\_hat\_model1)^2)

RSS\_Model\_1

# for Calculating Y-hat and RSS of model 2

Y\_hat\_model2 = X\_model2 %\*% Model2\_thetahat

Y\_hat\_model2

RSS\_Model\_2=sum((Y\_CSV\_DATA-Y\_hat\_model2)^2)

RSS\_Model\_2

# for Calculating Y-hat and RSS of model 3

Y\_hat\_model3 = X\_model3 %\*% Model3\_thetahat

Y\_hat\_model3

RSS\_Model\_3=sum((Y\_CSV\_DATA-Y\_hat\_model3)^2)

RSS\_Model\_3

# for Calculating Y-hat and RSS of model 4

Y\_hat\_model4 = X\_model4 %\*% Model4\_thetahat

Y\_hat\_model4

RSS\_Model\_4=sum((Y\_CSV\_DATA-Y\_hat\_model4)^2)

RSS\_Model\_4

# for Calculating Y-hat and RSS of model 5

Y\_hat\_model5 = X\_model5 %\*% Model5\_thetahat

Y\_hat\_model5

RSS\_Model\_5=sum((Y\_CSV\_DATA-Y\_hat\_model5)^2)

RSS\_Model\_5

### Task 2.3 for Calculating likelihood and Variance of each model

N=length(Y\_CSV\_DATA)

# for Calculating the Variance of Model 1

Variance\_model1=RSS\_Model\_1/(N-1)

Variance\_model1

# for Calculating the log-likelihood of Model 1

likehood\_Model\_1=

-(N/2)\*(log(2\*pi))-(N/2)\*(log(Variance\_model1))-(1/(2\*Variance\_model1))\*RSS\_Model\_1

likehood\_Model\_1

# for Calculating Variance and log-likelihood of Model 2

Variance\_model2=RSS\_Model\_2/(N-1)

Variance\_model2

likehood\_Model\_2=

-(N/2)\*(log(2\*pi))-(N/2)\*(log(Variance\_model2))-(1/(2\*Variance\_model2))\*RSS\_Model\_2

likehood\_Model\_2

#for calculating Variance and log-likelihood of Model 3

Variance\_model3=RSS\_Model\_3/(N-1)

Variance\_model3

likehood\_Model\_3=

-(N/2)\*(log(2\*pi))-(N/2)\*(log(Variance\_model3))-(1/(2\*Variance\_model3))\*RSS\_Model\_3

likehood\_Model\_3

# for calculating Variance and log-likelihood of Model 4

Variance\_model4=RSS\_Model\_4/(N-1)

Variance\_model4

likehood\_Model\_4=

-(N/2)\*(log(2\*pi))-(N/2)\*(log(Variance\_model4))-(1/(2\*Variance\_model4))\*RSS\_Model\_4

likehood\_Model\_4

# for calculating Variance and log-likelihood of Model 5

Variance\_model5=RSS\_Model\_5/(N-1)

Variance\_model5

likehood\_Model\_5=

-(N/2)\*(log(2\*pi))-(N/2)\*(log(Variance\_model5))-(1/(2\*Variance\_model5))\*RSS\_Model\_5

likehood\_Model\_5

# Task 2.4

# for calculating AIC And BIC of Each model

# for calculating AIC and BIC of model 1

K\_model1<-length(Model1\_thetahat)

K\_model1

AIC\_model1=2\*K\_model1-2\*likehood\_Model\_1

AIC\_model1

BIC\_model1=K\_model1\*log(N)-2\*likehood\_Model\_1

BIC\_model1

## thetahat of model 2

K\_model2<-length(Model2\_thetahat)

K\_model2

##Calculating AIC and BIC of model 2

AIC\_model2=2\*K\_model2-2\*likehood\_Model\_2

AIC\_model2

BIC\_model2=K\_model2\*log(N)-2\*likehood\_Model\_2

BIC\_model2

## thetahat of model 3

K\_model3<-length(Model3\_thetahat)

K\_model3

## for calculating AIC and BIC of model 3

AIC\_model3=2\*K\_model3-2\*likehood\_Model\_3

AIC\_model3

BIC\_model3=K\_model3\*log(N)-2\*likehood\_Model\_3

BIC\_model3

## thetahat of model 4

K\_model4<-length(Model4\_thetahat)

K\_model4

## for calculating AIC and BIC of model 4

AIC\_model4=2\*K\_model4-2\*likehood\_Model\_4

AIC\_model4

BIC\_model4=K\_model4\*log(N)-2\*likehood\_Model\_4

BIC\_model4

## thetahat of model 5

K\_model5<-length(Model5\_thetahat)

K\_model5

## for calculating AIC and BIC of model 5

AIC\_model5=2\*K\_model5-2\*likehood\_Model\_5

AIC\_model5

BIC\_model5=K\_model5\*log(N)-2\*likehood\_Model\_5

BIC\_model5

par(mfrow=c(1,1))

## Task 2.5

## Error of model1

model1\_error <- Y\_CSV\_DATA-Y\_hat\_model1

model1\_error

## to plot the graph QQplot and QQ line of model 1

qqnorm(model1\_error, col = "darkblue",main = "QQ plot of model 1")

qqline(model1\_error, col = "brown",lwd=1)

## Error of model2

model2\_error <- Y\_CSV\_DATA-Y\_hat\_model2 # error of model 2

## to plot QQplot and QQ line of model 2

qqnorm(model2\_error, col = "darkblue",main = "QQ plot of model 2")

qqline(model2\_error, col = "brown")

## Error of model3

model3\_error <- Y\_CSV\_DATA- Y\_hat\_model3

## to plot QQplot and QQ line of model 3

qqnorm(model3\_error, col = "darkblue",main = "QQ plot of model 3")

qqline(model3\_error, col = "brown")

## Error of model4

model4\_error <- Y\_CSV\_DATA-Y\_hat\_model4

## to plot QQplot and QQ line of model 4

qqnorm(model4\_error, col = "darkblue",main = "QQ plot of model 4")

qqline(model4\_error, col = "brown")

## Error of model5

model5\_error <- Y\_CSV\_DATA- Y\_hat\_model5

## to plot QQplot and QQ line of model 5

qqnorm(model5\_error, col = "darkblue",main = "QQ plot of model 5")

qqline(model5\_error, col = "brown")

### Task 2.7

## we need to split the data of y into 2 form i.e. Traning and testing data set.

split\_Y\_CSV\_DATA<-initial\_split(data = as.data.frame(Y\_CSV\_DATA),prop=.7)

## Traning Y data split

Y\_training\_set<-training(split\_Y\_CSV\_DATA)

Y\_testing\_set<-as.matrix(testing(split\_Y\_CSV\_DATA))

## Testing Y data split

Y\_training\_data<-as.matrix(Y\_training\_set)

## Spliting the data of X into 2 form i.e. Traning and testing data set.

split\_X\_CSV\_DATA<-initial\_split(data = as.data.frame(X\_CSV\_DATA),prop=.7)

## Traning X data split

X\_training\_set<-training(split\_X\_CSV\_DATA)

## Testing X data split

X\_testing\_set<-as.matrix(testing(split\_X\_CSV\_DATA))

X\_testing\_data<-as.matrix(X\_testing\_set)

X\_training\_data<-as.matrix(X\_training\_set)

### to estimate model parameters using Traning set

traning\_ones=matrix(1 , length(X\_training\_set$X1),1)

# selected model 2 and equation is matching with model 2

X\_traning\_model<-cbind(traning\_ones,X\_training\_set[,"X4"],(X\_training\_set[,"X1"])^3,(X\_training\_set[,"X3"])^4)

traning\_thetahat=solve(t(X\_traning\_model) %\*% X\_traning\_model) %\*% t(X\_traning\_model) %\*% Y\_training\_data

### Model out/Prediction

Y\_testing\_hat = X\_testing\_data %\*% traning\_thetahat

Y\_testing\_hat

RSS\_testing=sum((Y\_testing\_set-Y\_testing\_hat)^2)

RSS\_testing

t.test(Y\_training\_data, mu=500, alternative="two.sided", conf.level=0.95)

C\_I1=-0.2783950

C\_I2=0.2762101

p2 <- plot(density(Y\_training\_data), col="blue", lwd=2,

main="Distribution of Traning Data")

abline(v=C\_I1,col="brown", lty=2)

abline(v=C\_I2,col="brown", lty=2)

thetaHat\_training =solve(t(X\_training\_data) %\*% X\_training\_data) %\*% t(X\_training\_data) %\*%Y\_training\_data

thetaHat\_training

length(thetaHat\_training)

dis\_test=density(Y\_training\_data)

plot((dis\_test))

plot(dis\_test,main = "Density plot of Y Signal")

### to calculate Confidential interval

z=1.96 ##(95%) Confidential interval

error=((Y\_testing\_set-Y\_testing\_hat))

n\_len=length(Y\_testing\_hat)

C\_I\_1= z sqrt( (error (1-error) ) / n\_len)

C\_I\_1

error

C\_I\_2= z sqrt( (error (1+error) / n\_len)

C\_I\_2

#Task 3

## we will use Model 2 , parameter are selected and kept constant.

arr\_1=0

arr\_2=0

f\_value=0

s\_value=0

Model2\_thetahat

#values from thetahat

thetebias <- 0.483065688 #choosen parameter

thetaone <-0.143578928 # chosen prarameter

thetatwo <- 0.010038614 # constant value

thetathree <- 0.001912836 # constant value

Epison <- RSS\_Model\_2 \* 2 ## fixing value of eplision

num <- 100 #number of iteration

## to calculate Y-hat for performing rejection ABC

counter <- 0

for (i in 1:num) {

range1 <- runif(1,-0.483065688,0.483065688) # calculating the range

range1

range2 <- runif(1,-0.143578928,0.143578928)

New\_thetahat <- matrix(c(range1,range2,thetatwo,thetathree))

New\_Y\_Hat <- X\_model2 %\*% New\_thetahat ## New Y hat

new\_RSS <- sum((Y\_CSV\_DATA-New\_Y\_Hat)^2)

new\_RSS

if (new\_RSS > Epison){

arr\_1[i] <- range1

arr\_2[i] <- range2

counter = counter+1

f\_value <- matrix(arr\_1)

s\_value <- matrix(arr\_2)

}

}

hist(f\_value)

hist(s\_value)

### to plot the graph

plot(f\_value,s\_value, col = c("brown", "blue"), main = "Joint and Marginal Posterior Distribution")

par(mfrow=c(1,1))