1.

Try to do it without loop:

attach(data)

mat=matrix(0,nrow=nrow(data),ncol=4)

mat[,1]=as.numeric((data[,1]+data[,2])\*(data[,1]+data[,3]))

mat[,2]=as.numeric((data[,1]+data[,2])\*(data[,2]+data[,4]))

mat[,3]=as.numeric((data[,3]+data[,4])\*(data[,1]+data[,3]))

mat[,4]=as.numeric((data[,3]+data[,4])\*(data[,2]+data[,4]))

# Calculate Chi-Square statistics for data

mat=mat/rowSums(data)

chi\_sqvalue=rowSums((data-mat)^2/mat)

# Get P-value

pvalue=1-pchisq(chi\_sqvalue,1)

# Count the rows where p-value reaches significant.

linenum=summary(pvalue<0.05)

However this time I got 935 rows of significance.

2.a.

f=function(X,p){

mdist<-as.matrix(dist(X)) ## calculate the Euclidian distance

mat<-matrix(rep(0,dim(X)[1]\*p),nrow=dim(X)[1],ncol=p) ## initialize the output matrix

k=p+1

for(i in 1:dim(X)[1]){

mvec<-mdist[i,] ## pick out distance for each row

pmvec<-order(mvec,decreasing=F)[2:k] ## pick out the closest p rows for row i

mat[i,]=mat[i,]+pmvec} ## put the p rows in the output matrix

return(mat) ## output the result

}

b.

fma=function(X,p){

mat<-matrix(rep(0,dim(X)[1]\*p),nrow=dim(X)[1],ncol=p)

k=p+1

for(i in 1:dim(X)[1]){

madist<-as.matrix(mahalanobis(X,X[i],cov(X)))

pamvec<-order(madist,decreasing=F)[2:k]

mat[i,]=mat[i,]+pamvec}

return(mat)}

c.

f=function(X,p){

mdist<-as.matrix(dist(X))

mat<-matrix(rep(0,dim(X)[1]\*p),nrow=dim(X)[1],ncol=p)

k=p+1

for(i in 1:dim(X)[1]){

mvec<-mdist[i,]

pmvec<-order(mvec,decreasing=F)[2:k]

mat[i,]=mat[i,]+pmvec}

return(mat)

}

fma=function(X,p){

mat<-matrix(rep(0,dim(X)[1]\*p),nrow=dim(X)[1],ncol=p)

k=p+1

for(i in 1:dim(X)[1]){

madist<-as.matrix(mahalanobis(X,X[i],cov(X)))

pamvec<-order(madist,decreasing=F)[2:k]

mat[i,]=mat[i,]+pamvec}

return(mat)}

# Generate the test Matrix

test\_p=5\*(1:5)

test\_nrow=1000\*(1:10)

test\_ncol=10\*(1:10)

# Generate the time recorder for each nrow,ncol,p

timer\_euclidean=array(0,c(length(test\_p),length(test\_nrow),length(test\_ncol)))

timer\_mahalanobis=array(0,c(length(test\_p),length(test\_nrow),length(test\_ncol)))

for(idx\_p in (1:length(test\_p))){

for(idx\_nrow in (1:length(test\_nrow))){

for(idx\_ncol in (1:length(test\_ncol))){

# Generate the random Matrix for testing the time rX=matrix(rnorm(test\_nrow[idx\_nrow]\*test\_ncol[idx\_ncol]),c(test\_nrow[idx\_nrow],test\_ncol[idx\_ncol]))

test\_time=proc.time()

dummyrun=f(rX,test\_p[idx\_p])

timer\_euclidean[idx\_p,idx\_nrow,idx\_ncol]=as.numeric(proc.time()-test\_time)[1]

test\_time=proc.time()

dummyrun2=fma(rX,test\_p[idx\_p])

timer\_mahalanobis[idx\_p,idx\_nrow,idx\_ncol]=as.numeric(proc.time()-test\_time)[1]

}

}

}

# Save the time data in a file

save(timer\_euclidean,timer\_mahalanobis,file="test\_timer.RData")

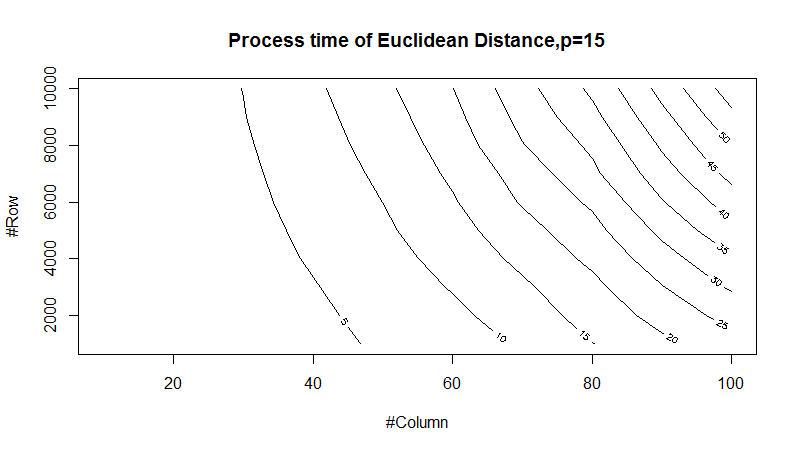
q(save="no")load('test\_timer.RData')

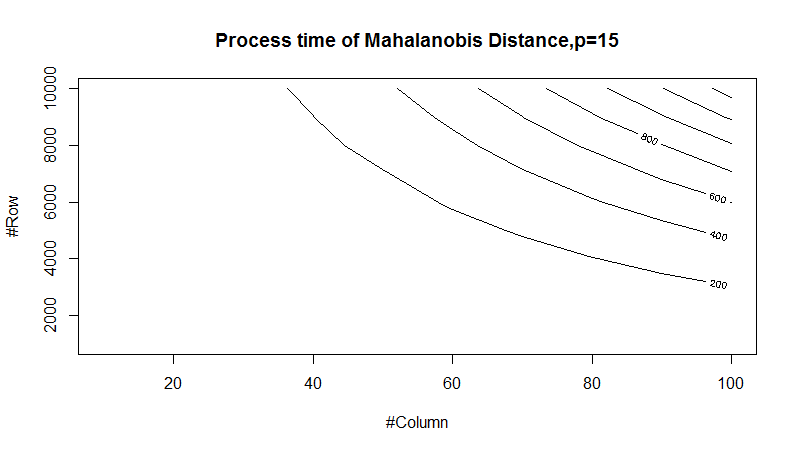
# Load the file and plot time against nrow,ncol,p

load("C:/Users/Heathtasia/Desktop/test\_timer.RData")

contour(x=10\*(1:10),y=1000\*(1:10),timer\_euclidean[3,,],xlab="#Column",ylab="#Row",main="Process time of Euclidean Distance,p=15")

contour(x=10\*(1:10),y=1000\*(1:10),timer\_mahalanobis[3,,],xlab="#Column",ylab="#Row",main="Process time of Mahalanobis Distance,p=15")





3.

library(dplyr)

data2 =read.csv("E:/MA 2015/STAT506ps2\_1.csv",header=FALSE)

index<-duplicated(data2$Id)

data2=data2[!index,]

data2$State=gsub("\t","",data2$State)

data2$State=gsub("\\'","",data2$State)

data2$State=gsub(" ","",data2$State)

data2$State=gsub("\\(","",data2$State)

data2$State=gsub("\\)","",data2$State)

data2$State=gsub("\"","",data2$State)

data2$State=toupper(data2$State)

data2$State=gsub("HAWAII","HI",data2$State)

data2$State=gsub("KANSAS","KS",data2$State)

data2$State=gsub("ARIZONA","AZ",data2$State)

data2$State=gsub("CONN","CT",data2$State)

data2$State=gsub("CTCT","CT",data2$State)

data2$State=gsub("CALIF","CA",data2$State)

data2$State=gsub("MICH","MI",data2$State)

data2$State=gsub("IND","IN",data2$State)

ata2$State=gsub("HA","HI",data2$State)

data2$State=gsub("UNKNOWN",NA,data2$State)

data2$State=gsub("OTHER",NA,data2$State)

data2$State=gsub("IAKS",NA,data2$State)

data2$State=gsub("AD",NA,data2$State)

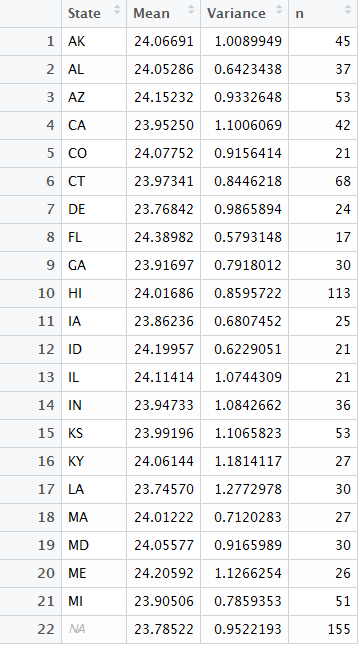
data2$State=gsub("OI",NA,data2$State)

data2$State=gsub("IB",NA,data2$State)

gb=group\_by(data2,State)

su<-summarize(gb,Mean=mean(weight),Variance=var(weight),n=n())

The output is below, with the same output as in Problem Set 1.



4.

library(foreign)

dental<-read.xport("C:/Users/Heathtasia/Desktop/OHXDEN\_F.XPT")

dgraphic<-read.xport("C:/Users/Heathtasia/Desktop/DEMO\_F.XPT")

dentaldata=inner\_join(dental,dgraphic,by="SEQN")

dgb=group\_by(dentaldata,RIDAGEYR)

test\_try<-summarize(dgb,Percentage=length(OHX01TC[OHX01TC==2])/length(OHX01TC)-mean(is.na(OHX02TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX01")

test\_try<-summarize(dgb,Percentage=length(OHX02TC[OHX02TC==2])/length(OHX02TC)-mean(is.na(OHX02TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX02")

test\_try<-summarize(dgb,Percentage=length(OHX03TC[OHX03TC==2])/length(OHX03TC)-mean(is.na(OHX03TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX03")

test\_try<-summarize(dgb,Percentage=length(OHX04TC[OHX04TC==2])/length(OHX04TC)-mean(is.na(OHX04TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX04")

test\_try<-summarize(dgb,Percentage=length(OHX05TC[OHX05TC==2])/length(OHX05TC)-mean(is.na(OHX05TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX05")

test\_try<-summarize(dgb,Percentage=length(OHX06TC[OHX06TC==2])/length(OHX06TC)-mean(is.na(OHX06TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX06")

test\_try<-summarize(dgb,Percentage=length(OHX07TC[OHX07TC==2])/length(OHX07TC)-mean(is.na(OHX07TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX07")

test\_try<-summarize(dgb,Percentage=length(OHX08TC[OHX08TC==2])/length(OHX08TC)-mean(is.na(OHX08TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX08")

test\_try<-summarize(dgb,Percentage=length(OHX09TC[OHX09TC==2])/length(OHX09TC)-mean(is.na(OHX09TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX09")

test\_try<-summarize(dgb,Percentage=length(OHX10TC[OHX10TC==2])/length(OHX10TC)-mean(is.na(OHX10TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX10")

test\_try<-summarize(dgb,Percentage=length(OHX11TC[OHX11TC==2])/length(OHX11TC)-mean(is.na(OHX11TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX11")

test\_try<-summarize(dgb,Percentage=length(OHX12TC[OHX12TC==2])/length(OHX12TC)-mean(is.na(OHX12TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX12")

test\_try<-summarize(dgb,Percentage=length(OHX13TC[OHX13TC==2])/length(OHX13TC)-mean(is.na(OHX13TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX13")

test\_try<-summarize(dgb,Percentage=length(OHX14TC[OHX14TC==2])/length(OHX14TC)-mean(is.na(OHX14TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX14")

test\_try<-summarize(dgb,Percentage=length(OHX15TC[OHX15TC==2])/length(OHX15TC)-mean(is.na(OHX15TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX15")

test\_try<-summarize(dgb,Percentage=length(OHX16TC[OHX16TC==2])/length(OHX16TC)-mean(is.na(OHX16TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX16")

test\_try<-summarize(dgb,Percentage=length(OHX17TC[OHX17TC==2])/length(OHX17TC)-mean(is.na(OHX17TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX17")

test\_try<-summarize(dgb,Percentage=length(OHX18TC[OHX18TC==2])/length(OHX18TC)-mean(is.na(OHX18TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX18")

test\_try<-summarize(dgb,Percentage=length(OHX19TC[OHX19TC==2])/length(OHX19TC)-mean(is.na(OHX19TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX19")

test\_try<-summarize(dgb,Percentage=length(OHX20TC[OHX20TC==2])/length(OHX20TC)-mean(is.na(OHX20TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX20")

test\_try<-summarize(dgb,Percentage=length(OHX21TC[OHX21TC==2])/length(OHX21TC)-mean(is.na(OHX21TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX21")

test\_try<-summarize(dgb,Percentage=length(OHX22TC[OHX22TC==2])/length(OHX22TC)-mean(is.na(OHX22TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX22")

test\_try<-summarize(dgb,Percentage=length(OHX23TC[OHX23TC==2])/length(OHX23TC)-mean(is.na(OHX23TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX23")

test\_try<-summarize(dgb,Percentage=length(OHX24TC[OHX24TC==2])/length(OHX24TC)-mean(is.na(OHX24TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX24")

test\_try<-summarize(dgb,Percentage=length(OHX25TC[OHX25TC==2])/length(OHX25TC)-mean(is.na(OHX25TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX25")

test\_try<-summarize(dgb,Percentage=length(OHX26TC[OHX26TC==2])/length(OHX26TC)-mean(is.na(OHX26TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX26")

test\_try<-summarize(dgb,Percentage=length(OHX27TC[OHX27TC==2])/length(OHX27TC)-mean(is.na(OHX27TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX27")

test\_try<-summarize(dgb,Percentage=length(OHX28TC[OHX28TC==2])/length(OHX28TC)-mean(is.na(OHX28TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX28")

test\_try<-summarize(dgb,Percentage=length(OHX29TC[OHX29TC==2])/length(OHX29TC)-mean(is.na(OHX29TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX29")

test\_try<-summarize(dgb,Percentage=length(OHX30TC[OHX30TC==2])/length(OHX30TC)-mean(is.na(OHX30TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX30")

test\_try<-summarize(dgb,Percentage=length(OHX31TC[OHX31TC==2])/length(OHX31TC)-mean(is.na(OHX31TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX31")

test\_try<-summarize(dgb,Percentage=length(OHX32TC[OHX31TC==2])/length(OHX32TC)-mean(is.na(OHX32TC)),n=n())

plot(test\_try$RIDAGEYR,test\_try$Percentage,xlab="Age",ylab="Existing Percentage",main="OHX32")

