# Problem Set 2

#### Prateek Kumar ([prateekk@mtu.edu](mailto:prateekk@mtu.edu))

12 September 2018

# 1. Estimating Parameters of distributions.

out10<-matrix(,0,2) #creating 3 empty matrices of size[0,2]  
out100<-matrix(,0,2)  
out10000<-matrix(,0,2)  
i=0 #defining variable i and j for iteration  
j=0  
  
#This function returns the mean and std deviation of value N  
fun\_dist<-function(N){  
 lis<-rnorm(N)  
 return(c(round(mean(lis),3),round(sd(lis),3))) #returns as a vector  
}  
  
for(i in c(10,100,10000)) #for loop for the 3 types of normal distribution  
{  
 if(i==10)  
 {  
 for(j in 1:500) #running for 500 times  
 {  
 out10<-rbind(out10,fun\_dist(i)) #binding the mean and sd values for N=10   
 }  
 colnames(out10)<-c('mean','sd') #assigning names to the columns  
 }  
 if(i==100)  
 {  
 for(j in 1:500) #running for 500 times  
 {  
 out100<-rbind(out100,fun\_dist(i)) #binding the mean and sd values for N=100  
 }  
 colnames(out100)<-c('mean','sd') #assigning names to the columns  
 }  
 if(i==10000)  
 {  
 for(j in 1:500) #running for 500 times  
 {  
 out10000<-rbind(out10000,fun\_dist(i)) #binding the mean and sd values for N=10000  
 }  
 colnames(out10000)<-c('mean','sd') #assigning names to the columns  
 }  
}  
  
#Means  
par(mfrow=c(1,3)) #create a matrix of 1 row 3 ncols for plots  
  
#histogram for mean of 10 normal distribution  
hist(out10[,1],main = 'Mean\_10',xlab = 'Mean range', border="red", col="yellow",xlim = c(-1.2,1.2),ylim = c(0,120),las=1,breaks = 10)   
  
lines(density(out10[,1])) #showing the density of the points over the mean range  
  
#histogram for mean of 100 normal distribution  
hist(out100[,1],main = 'Mean\_100',xlab = 'Mean range', border="red", col="yellow",xlim = c(-0.35,0.35),ylim = c(0,120),las=1,breaks = 10)  
  
lines(density(out100[,1])) #showing the density of the points over the mean range  
  
#histogram for mean of 10000 normal distribution  
hist(out10000[,1],main = 'Mean\_10000',xlab = 'Mean range', border="red", col="yellow",xlim = c(-0.035,0.035),ylim = c(0,120),las=1,breaks = 10)  
  
lines(density(out10000[,1])) #showing the density of the points over the mean range  
  
#Standard Deviation  
par(mfrow=c(1,3)) #create a matrix of 1 row 3 ncols for plots  
  
#histogram for std deviation of 10 normal distribution  
hist(out10[,2],main = 'sd\_10',xlab = 'sd range', border="red", col="yellow",xlim = c(.3,1.7),ylim = c(0,140),las=1,breaks = 10)  
lines(density(out10[,2])) #showing the density of the points over the sd range  
  
#histogram for std deviation of 100 normal distribution  
hist(out100[,2],main = 'sd\_100',xlab = 'sd range', border="red", col="yellow",xlim = c(.75,1.25),ylim = c(0,140),las=1,breaks = 10)  
lines(density(out100[,2])) #showing the density of the points over the sd range  
  
#histogram for std deviation of 10000 normal distribution  
hist(out10000[,2],main = 'sd\_10000',xlab = 'sd range', border="red", col="yellow",xlim = c(.97,1.03),ylim = c(0,140),las=1,breaks = 10)  
lines(density(out10000[,2])) #showing the density of the points over the sd range

# 

Figure 1: This figure shows the distribution of mean. We see in this figure that when the distribution size is small then the mean is scattered over the range and we do not get an accurate mean and as we process further taking a normal distribution of 100 and 10000 numbers then the range tends to decrease and the density around 0 increases.

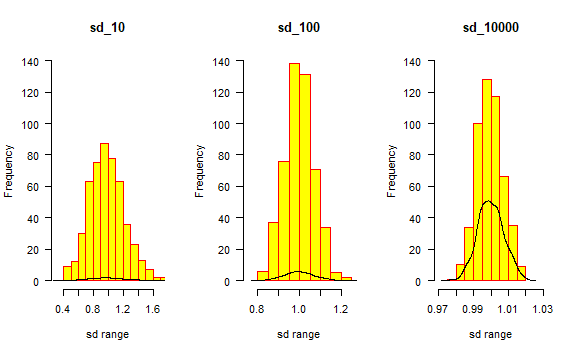


Figure 2: This figure shows the distribution of standard deviation. We see the similar case of means, we see in this figure that when the distribution size is small then the std. deviation is scattered over the range and we do not get an accurate std. deviation and as we process further taking a normal distribution of 100 and 10000 numbers then the range tends to decrease and the density around 1 increases.

# 2. Reading Data

#"Q2\_data.xlsx" contains data from the webpage  
Q2\_data <- read\_excel("Q2\_data.xlsx",skip = 1)   
#"State\_excel.xlsx" contains the list of northern and southern states  
Q2\_state <- read\_excel("State\_excel.xlsx")   
  
r\_total<-Q2\_data$Total[1:56] #filtering the republican total  
d\_total<-Q2\_data$Total\_\_1[1:56] #filtering the democrat total  
  
northern\_states<-c() #creating empty vectors of northern and southern states  
southern\_states<-c()  
  
for(i in Q2\_state$NS)  
{  
 northern\_states<-c(northern\_states,i)   
 #taking the list of northern states and storing it in a vector  
}  
for(i in Q2\_state$SS)  
{  
 if(is.na(i)==FALSE)  
 {  
 southern\_states<-c(southern\_states,i)  
 #taking the list of southern states and storing it in a vector  
 }  
}  
  
#calculating the means of the democrat and republican parties based upon the northern #and southern states  
mean\_rep\_ns<-mean(Q2\_data$Total[which(Q2\_data$State %in% northern\_states)])  
mean\_demo\_ns<-mean(Q2\_data$Total\_\_1[which(Q2\_data$State %in% northern\_states)])  
mean\_rep\_ss<-mean(Q2\_data$Total[which(Q2\_data$State %in% southern\_states)])  
mean\_demo\_ss<-mean(Q2\_data$Total\_\_1[which(Q2\_data$State %in% southern\_states)])  
  
#par(mfrow=c(2,1))  
  
#plotting based upon the republican and democrat totals  
plot(r\_total,d\_total, xlim = c(0,200), ylim = c(0,500), pch=20 ,xlab = 'Republican Delegates', ylab = 'Democrat Delegates',type = 'n', main='Plot of Republican and Democrat Totals')  
#alloting the text values of the states  
text(r\_total,d\_total, labels = Q2\_data$Abb, pos = 2, cex = 0.75)  
#Drawing the lines from the means of democrat and republicans totals  
abline(a=0,b=(mean\_demo\_ns/mean\_rep\_ns))  
abline(a=0,b=(mean\_demo\_ss/mean\_rep\_ss))  
  
#alloting the totals of the democrat and republican parties to variables based upon the northern #and southern states  
rep\_ns<-Q2\_data$Total[which(Q2\_data$State %in% northern\_states)]  
rep\_ss<-Q2\_data$Total[which(Q2\_data$State %in% southern\_states)]  
demo\_ns<-Q2\_data$Total\_\_1[which(Q2\_data$State %in% northern\_states)]  
demo\_ss<-Q2\_data$Total\_\_1[which(Q2\_data$State %in% southern\_states)]  
  
#plotting again based upon the republican and democrat totals  
plot(r\_total,d\_total, xlim = c(0,200), ylim = c(0,500), pch=20 ,xlab = 'Republican Delegates', ylab = 'Democrat Delegates',type = 'n',main='Plot of Republican and Democrat Totals\n showing Northern and Southern states')  
#Showing the northern states as blue  
text(rep\_ns,demo\_ns, labels = Q2\_state$NS\_abb, pos = 2, cex = 0.75, col = 'blue')  
#Showing the southern states as red  
text(rep\_ss,demo\_ss, labels = Q2\_state$SS\_abb, pos = 2, cex = 0.75, col = 'red')  
#Drawing the lines from the means of democrat and republicans totals  
abline(a=0,b=(mean\_demo\_ns/mean\_rep\_ns))  
abline(a=0,b=(mean\_demo\_ss/mean\_rep\_ss))  
#Legend for determing the northern and southern states  
legend(0, 500, legend=c("Northern States", "Southern States"), col=c("blue","red"), lty=1:2, cex=0.8, title="State types", text.font=4, bg='lightblue')

# 

Figure 3: This figure is the plot of Republican totals vs Democrat totals. From the figure we can see that ‘California’ is over represented by both democrats and republicans followed by ‘New York’ and ‘Florida’ for democrats and ‘Texas’ and ‘Florida’ for Republicans. And checking for the states which are underrepresented it’s a bit difficult to see from the plot but from the data we can find that for Republicans ‘Virgin Islands’, ‘N. Marianas’, ‘Guam’ and ‘Amer.Samoa’ is most under represented and is same for Democrats as well.

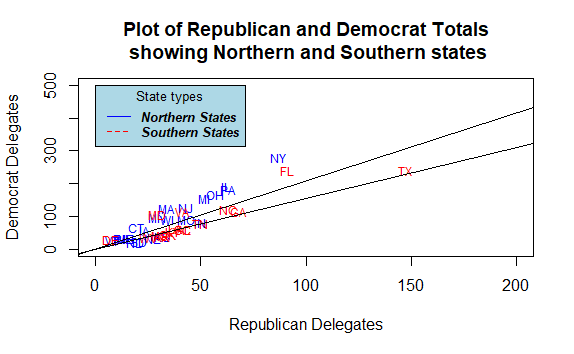


Figure 4: This figure clearly shows the northern and southern Democrat and Republican states. This division makes a total sense as we can easily distinguish between the northern and southern states of democrats and republicans and also we can easily identify as which states are overrepresented and underrepresented by both the parties.

# 3. Filtering and Sorting

member <- read.csv(file="senate-2014.csv", header=TRUE) #reading the .csv file  
summary(member) #shows result summaries of the dataset

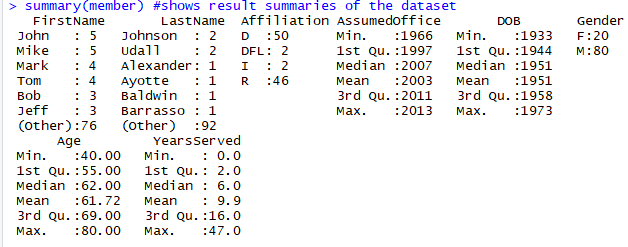


Figure 5: Summary of the data

#table(member$FirstName)  
#table(member$LastName)  
table(member$Affiliation) #shows the count of factor levels  
table(member$AssumedOffice)  
table(member$DOB)  
table(member$Gender)  
table(member$Age)  
table(member$YearsServed)

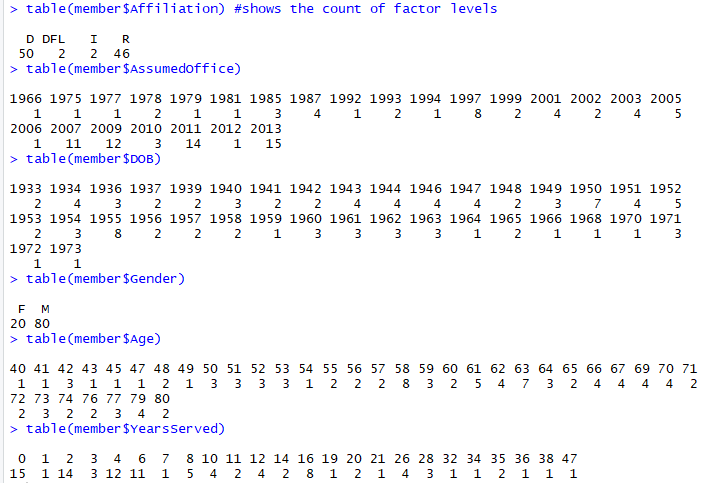


Figure 6: Table() of the columns output

par(mfrow=c(3,2))  
#boxplot of columns w.r.t. Years served  
boxplot(YearsServed~Affiliation,data=member, main='YearsServed~Affiliation', xlab="Affiliation", ylab="Years Served")   
boxplot(YearsServed~AssumedOffice,data=member, main='YearsServed~AssumedOffice', xlab="AssumedOffice", ylab="Years Served")  
boxplot(YearsServed~DOB,data=member, main='YearsServed~DOB', xlab="DOB", ylab="Years Served")  
boxplot(YearsServed~Gender,data=member, main='YearsServed~Gender', xlab="Gender", ylab="Years Served")  
boxplot(YearsServed~Age,data=member, main='YearsServed~Age', xlab="Age", ylab="Years Served")

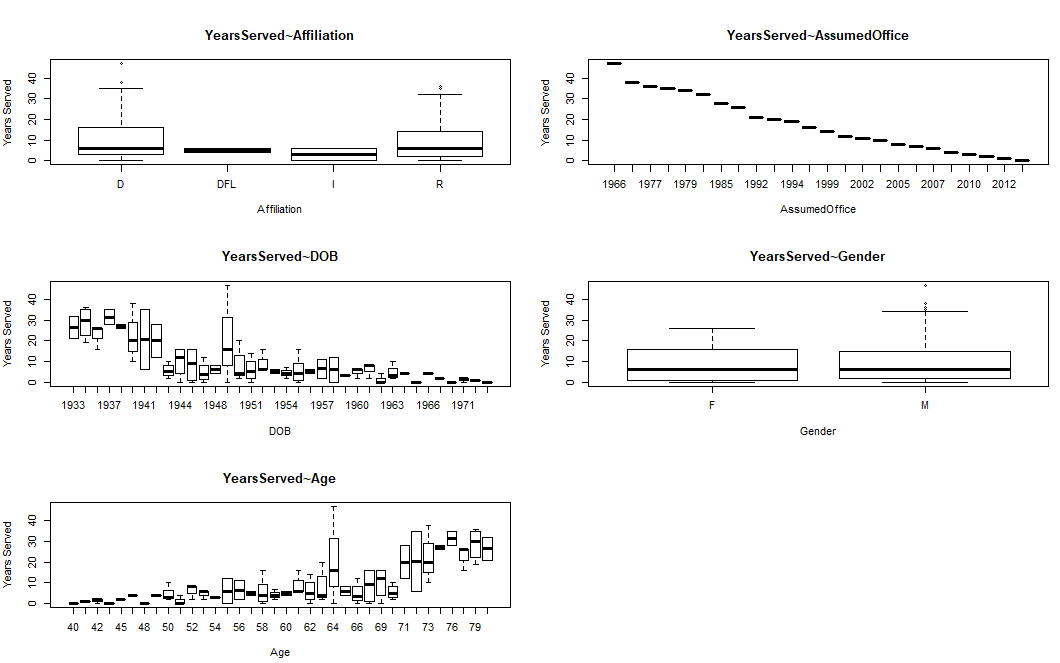


Figure 7:This figure shows the boxplots of the columns of ‘senate-2014’ with respect to the years served.

#histogram and barplots of columns  
par(mfrow=c(3,2))  
barplot(table(member$Affiliation), main='Affiliation')  
hist(member$AssumedOffice, main='AssumedOffice')  
hist(member$DOB, main='DOB')  
barplot(table(member$Gender), main='Gender')  
hist(member$Age, main='Age')  
hist(member$YearsServed, main='YearsServed')

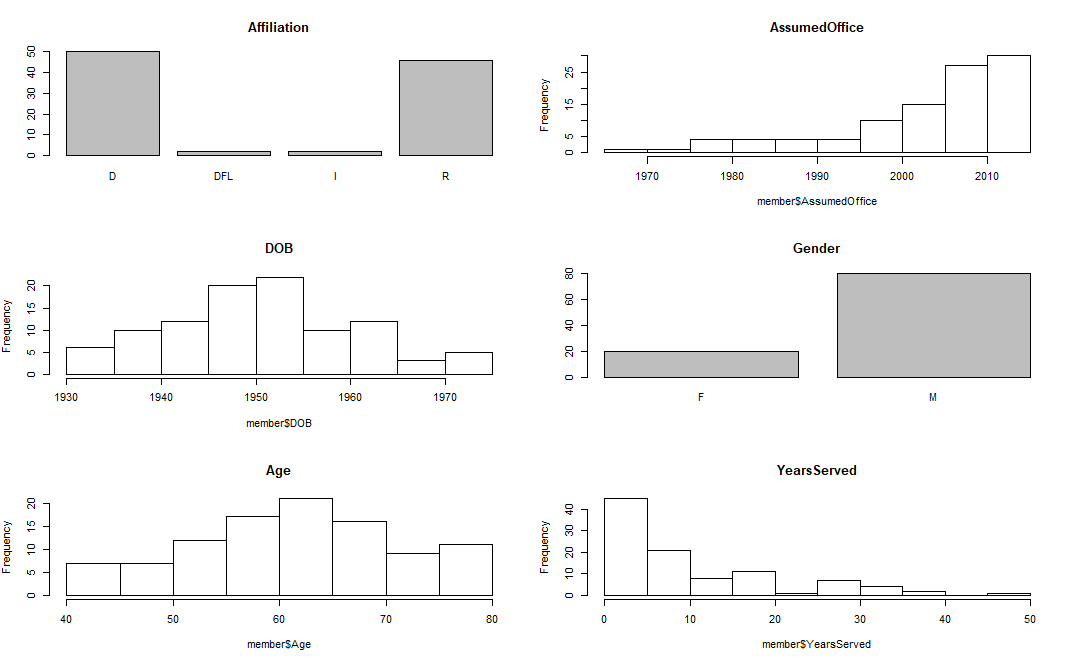


Figure 8: This figure shows the histograms of the columns of ‘senate-2014’. We have 2 barplots one for affiliation and one for gender because both of them has categorical data hence we cannot draw a histogram.

year\_10 <- member[which(member$YearsServed>10),] #selecting for years served > 10  
table(year\_10$Gender) #showing the count of factor levels  
table(year\_10$Affiliation)  
  
#calculating the mean age using aggregate function  
mean\_age<-aggregate(member$Age,list(party=member$Affiliation,gender=member$Gender),mean)  
mean\_age  
  
#calculating the mean age using tapply function  
mean\_age\_tapply<-tapply(member$Age,list(party=member$Affiliation,gender=member$Gender),mean)  
  
#Matplot for mean age using tapply  
matplot(mean\_age\_tapply,type = 'o',main='Mean age matplot')

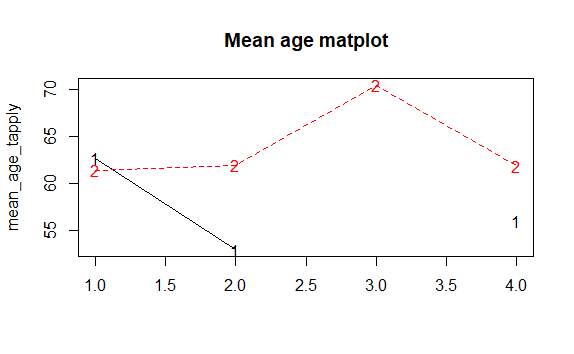


Figure 9: This figure is matplot of the mean age using tapply function.

#sorting the data by seniority   
newdata <- member[order(-member$YearsServed),]   
#Matplot for age and assumed office  
matplot(newdata$Age,newdata$AssumedOffice,type = 'o', main='Matplot of Age and Assumed Office', xlab="Age", ylab="Assumed Office")

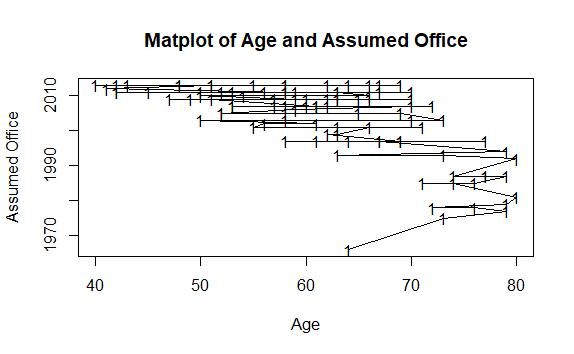


Figure 10: This figure is the matplot of ‘Age’ and ‘Assumed Office’ after sorting the data based upon the seniority of the people.

#sorting the data by age  
newdata\_age <- member[order(member$Age),]   
#Matplot for years served in order of age  
matplot(newdata\_age$YearsServed,type = 'o', main='Matplot of Age and Years Served',ylab="Years Served", xlab="Age")

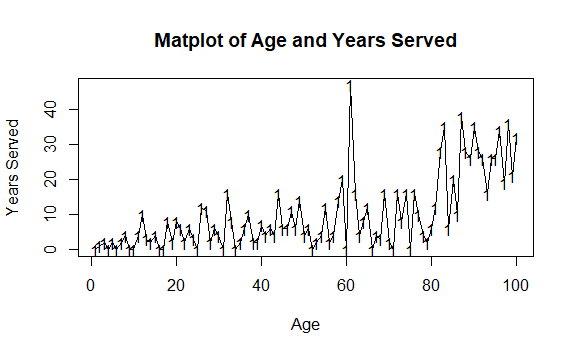


Figure 11: This figure is the matplot of ‘Age’ and ‘Years served’ after sorting the data based upon the age of the people.

# 4. Programming in R

#function DoLetters  
DoLetters<-function(x=1) #setting default value to 1  
{  
 return(LETTERS[1:x]) #returns alphabet upto x  
}  
DoLetters(8)



Figure 12: Sample output of function DoLetters()

#function DoLetters1  
DoLetters1<-function(x=1,y=26) #setting default range from 1 to 26  
{  
 if(x>0 && y>0 && x<27 && y<27)  
 {  
 return(LETTERS[x:y]) #returning the sublist of alphabets  
 }  
 else  
 {  
 warning("Please enter the values of x & y between 1 and 26")   
 #Error checking if the value passed is beyond the alphabet range  
 }  
}  
DoLetters1(3,6)





Figure 13: Sample output of function DoLetters1()

#histogram function  
myfunction<-function(x1,xlab='Range',ylab='Frequency')  
{  
 if(is.numeric(x1)==TRUE && is.matrix(x1)==FALSE)   
 {  
 hist(x1,xlab = xlab,ylab = ylab,main="Histogram")  
 #if class is numeric then plots the histogram  
 }  
 else if(is.factor(x1))   
 {  
 barplot(table(x1),xlab = xlab,ylab = ylab,main='Barplot')  
 #if class is factor then plots the barplot  
 }  
 else if(is.matrix(x1))   
 {  
 pairs(x1,main='Pairwise scatter plot of the matrix')  
 #if class is matrix then plots the pairs plot  
 }  
 else  
 {  
 warning("Please enter either numeric or factor or matrix")   
 }  
}  
x1 <- runif(1000)  
#hist(x1)  
x2 <- sample(as.factor(1:5),1000,replace=T)  
#hist(x2)  
a <- runif(1000)  
b <- runif(1000) + a  
c <- runif(1000) + b  
mat <- cbind(a,b,c)  
myfunction(x1)  
myfunction(x2)  
myfunction(mat)

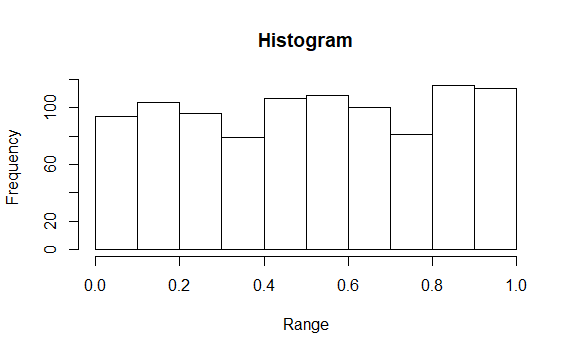


Figure 14: We get a histogram when the value passed through myfunction() is numeric.

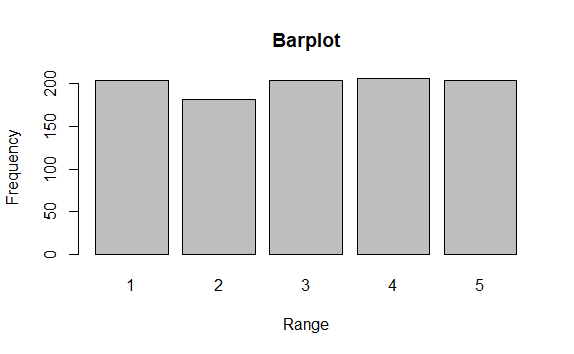


Figure 15:We get a barplot when the value passed through myfunction() is factor.

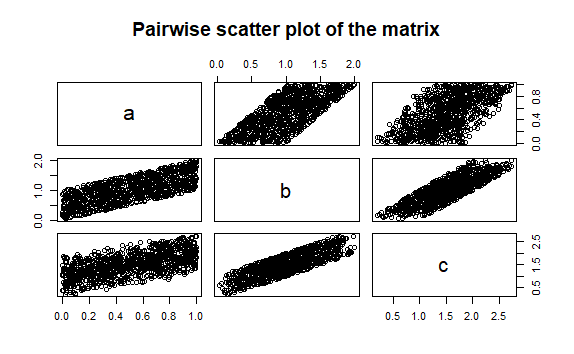


Figure 16:We get a pairwise scatter plot when the value passed through myfunction() is matrix.

#Getting the output using while loop  
i=1  
k=1  
mat\_while<-matrix( ,0,2)  
mat\_for<-matrix( ,0,2)  
  
while(i<nrow(mat)+1)  
{  
 if (mat[i,3]<mat[i,1]\*2)   
 {  
 mat\_while<-rbind(mat\_while,c(mat[i,2],i))  
 }  
 i=i+1  
}  
  
#Getting the output using for loop  
for (k in 1:nrow(mat))   
{  
 if (mat[k,3]<mat[k,1]\*2)   
 {  
 mat\_for<-rbind(mat\_for,c(mat[k,2],k))  
 }  
}  
  
#Getting the output without using loops  
my\_mat<-as.matrix(data.frame(mat[which(mat[,3]<mat[,1]\*2),2],which(mat[,3]<mat[,1]\*2)))  
colnames(my\_mat)<-c('b','V2')

my\_mat

#When we compare the values of mat\_while, mat\_for and my\_mat we see that all 3 matrices are identical hence we can say that our results are correct.

*Output Snippet:*

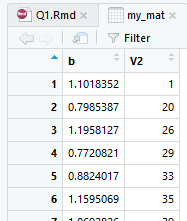


Figure 17: my\_mat matrix

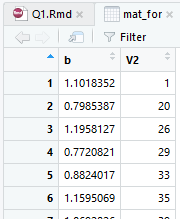


Figure 18: mat\_for matrix

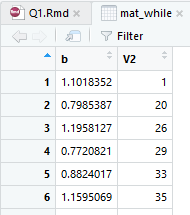


Figure 19: mat\_while matrix