# DATA MINING PROJECT

## GROUP MEMBERS

## AJITESH PRAVEEN: 15MIS1104

## AMIT JANGIR : 15MIS1031

## SHIVA GOPI : 15MIS1166

# FACULTY NAME: Prof. MAHESWARI N

TOPIC: FINDING THE CLOSELY RELATED STATES OF INDIA BASED ON CRIMES COMMITED ON CHILDREN BY MEN AND WOMEN USING CLUSTERING ALGORITHM

ALGORITHM USED: CLUSTEIRNG (COMPLETE LINKAGE) ALGORITHM

APPLICATION USED: RSTUDIO

PROGRAMMING LANGUAGE USED: JAVA

IMPLEMENTATION USING RSTUDIO:

CODE:

utilities<-read.csv("E:/Study/Data Mining Dataset/child\_crime\_updated\_14thMarch.csv")

str(utilities)

z<-utilities[,-c(1,1)]

z1<-z[,-c(1,1)]

str(z1)

z1

distance <- dist(z1)

distance

hc.c <- hclust(distance)

plot(hc.c,labels=utilities$States)

hc.c

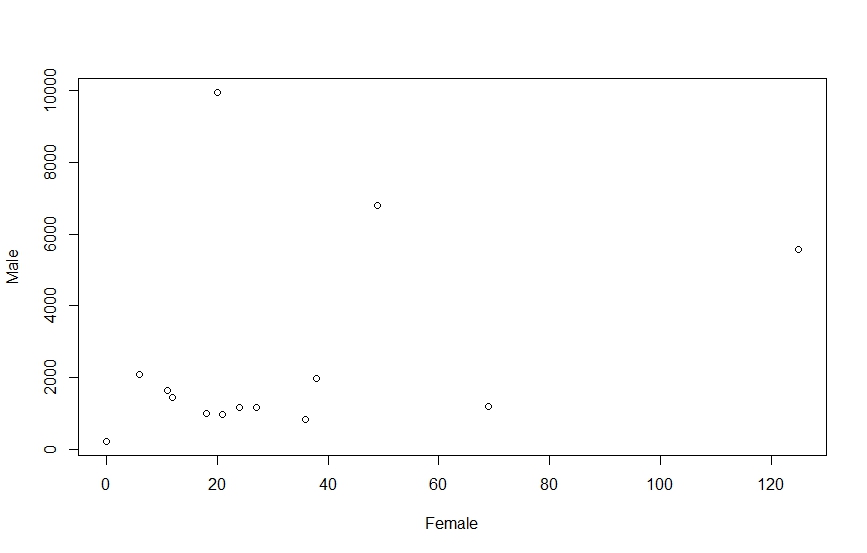
plot(Male ~ Female,utilities)

#with(utilities,text(Persons.arrested.during.the.year\_Male ~ Persons.arrested.during.the.year\_Female,labels=Crime.Head))

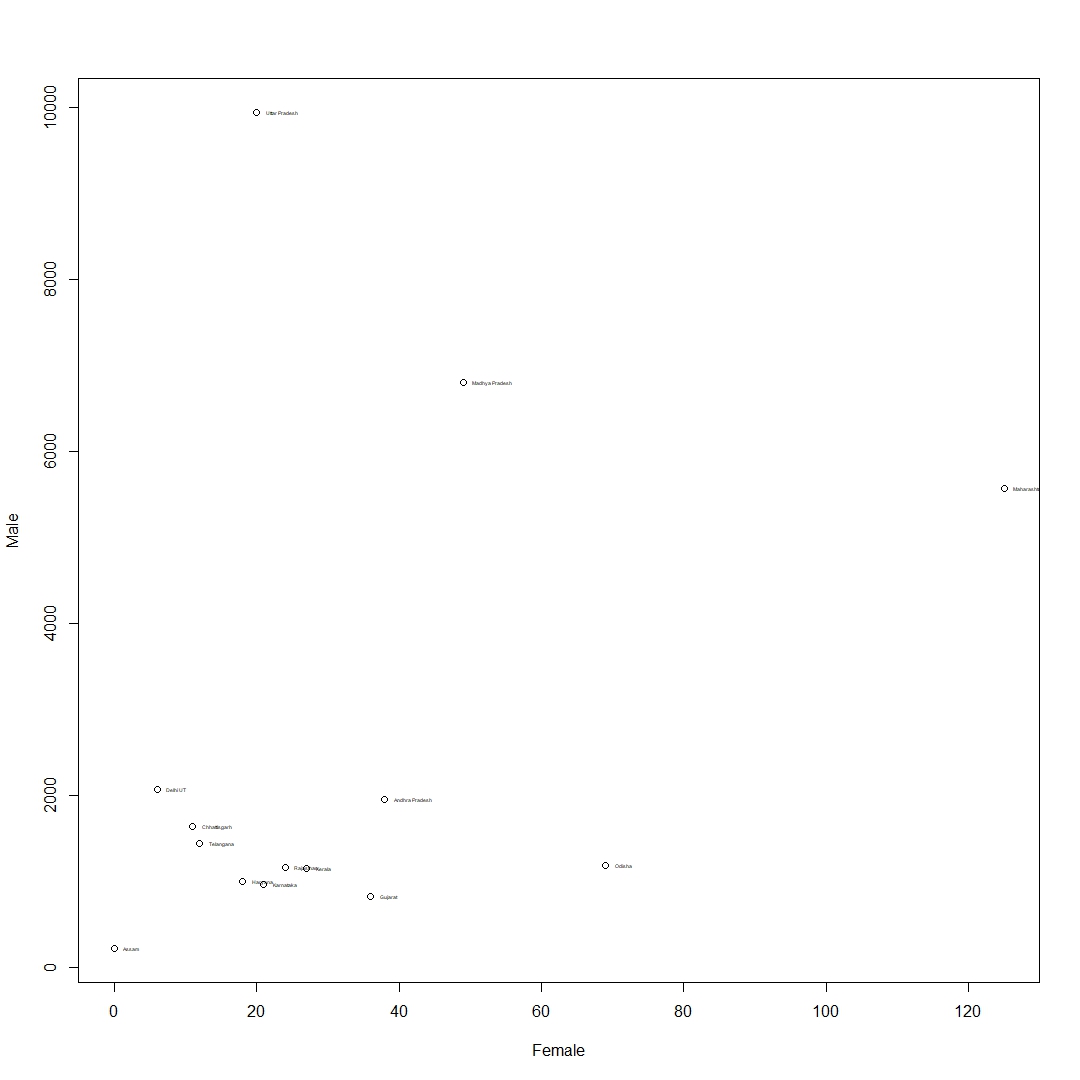
with(utilities,text(Male ~ Female,labels=States,pos=4,cex=.5))

OUTPUTS :

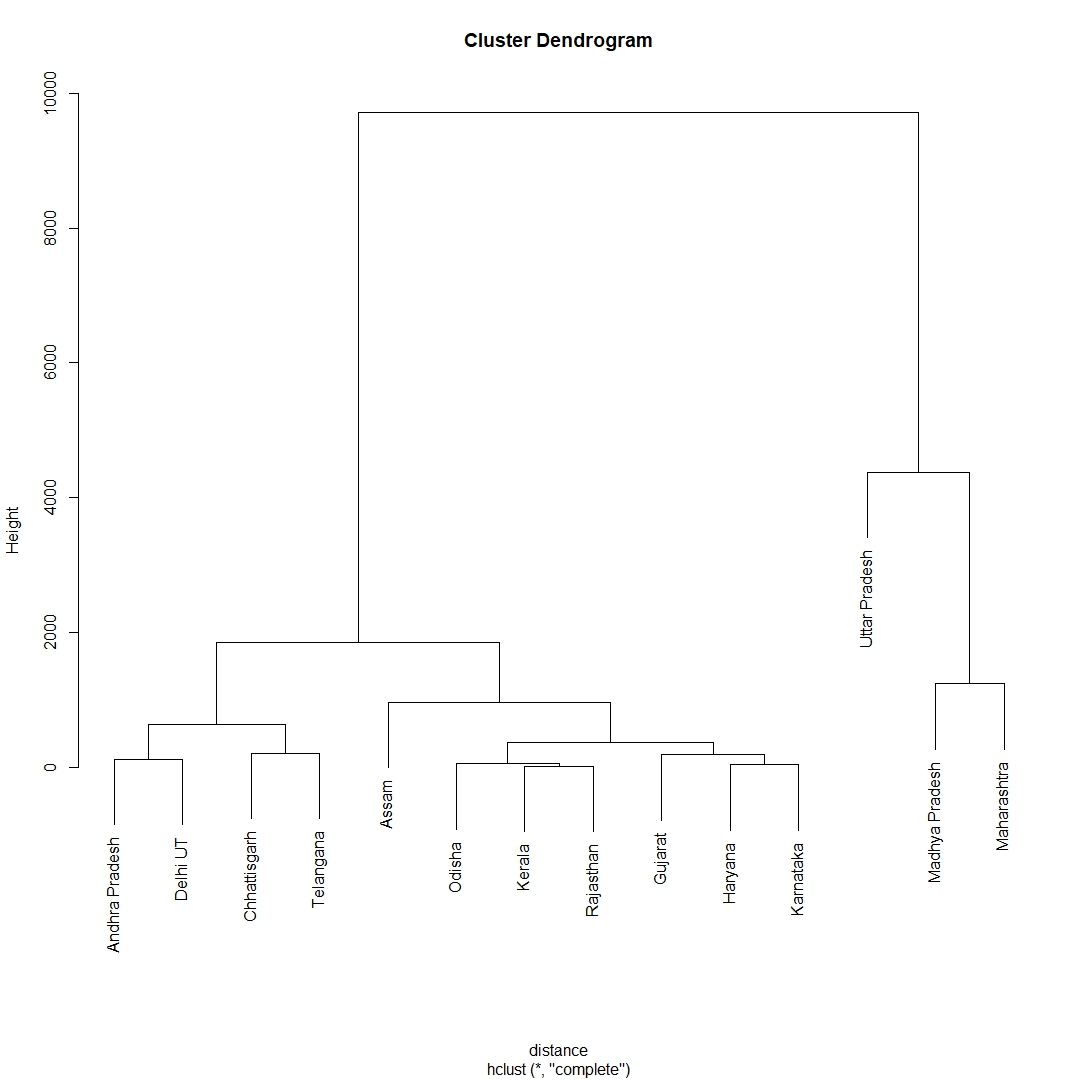
MALE VS FEMALE (PLOT)–



MALE VS FEMALE LABELED WITH STATE NAME (PLOT)



DENDOGRAM :



IMPLEMENTATION USING PROGRAMMING LANGUAGE (JAVA)-

CODE :

import java.io.\*;;

import java.util.\*;

import java.math.\*;

import java.text.DecimalFormat;

public class Crime {

public static void dendogram(String arr[])

{

HashSet<String> den1 = new HashSet<String>();

int size = 0;

double accuracy = 0;

System.out.println(" \n Clusters without repetition :: \n");

for(int i = 0 ; i<arr.length ; i++)

{

den1.add(arr[i]);

}

for(String i:den1)

{

System.out.println(i);

size++;

}

double size1 = size;

accuracy = (size1/12.0)\*100.0;

System.out.println("\n Total number of clusters in Rstudio dendogram : 12 \n");

System.out.println("Accuracy in comparison to Rstudio : \t" + df1.format(accuracy));

}

public static double[] euclidean\_distance(int m1[] , int f1[])

{

int m1\_length = m1.length;

int f1\_length = f1.length;

double temp = 0.0;

double dist[] = new double[200];

int a = 0;

for(int i = 0 ; i<m1\_length ; i++)

{

for(int j = 0 ; j<f1\_length ; j++)

{

temp = Math.sqrt(Math.pow(Math.abs(m1[i]-m1[j]),2)+Math.pow(Math.abs(f1[i]-f1[j]),2));

dist[a] = temp;

temp = 0.0;

a++;

}

}

return dist;

}

public static double[][] putting\_0(int size)

{

double arr[][] = new double[size][size];

for(int i = 0 ; i<size ; i++)

{

for(int j = 0 ; j<size ; j++)

{

if(i == j)

{

arr[i][j] = 0.0;

}

}

}

return arr;

}

public static double find\_max(double a , double b)

{

if(a>=b)

{

return a;

}

else

{

return b;

}

}

public static int[] find\_indexs(double arr[][] , int size)

{

double min = 1000000.0;

int index[] = new int[2];

for(int i = 0 ; i<size ; i++)

{

for(int j = 0 ; j<size ; j++)

{

if(arr[i][j] <= min && arr[i][j]>0.0)

{

min = arr[i][j];

index[0] = i;

index[1] = j;

}

}

}

return index;

}

public static int[][] clusters(double arr[][] , int size\_r , int size\_c)

{

int clus[][] = new int[14][2];

int size = size\_r;

//finding initial minimum

int index\_i;

int index\_j;

int[] min = find\_indexs(arr,size\_r);

index\_i = min[0];

index\_j = min[1];

clus[0][0] = index\_i;

clus[0][1] = index\_j;

int a = 1;

double min2 = 10000000.0;

double temp[][] = new double[size][size];

while(size>1)

{

System.out.println("\n");

for(int i = 0 ; i<size ; i++)

{

for(int j = 0 ; j<size ; j++)

{

if(i==j)

{

temp[i][j] = 0.0;

}

else if(i==index\_i || i==index\_j)

{

temp[i][j] = find\_max(arr[i][index\_i],arr[i][index\_j]);

}

else

{

temp[i][j] = arr[i][j];

}

System.out.print(temp[i][j] + "\t");

}

System.out.println("\n");

}

size--;

for(int i = 0 ; i<size ; i++)

{

for(int j = 0 ; j<size ; j++)

{

if(i!=index\_j || j!=index\_j)

{

if(temp[i][j]<=min2 && temp[i][j]>0.0)

{

min2 = arr[i][j];

clus[a][0] = i;

clus[a][1] = j;

index\_i = i;

index\_j = j;

}

}

}

}

min2 = 100000000.0;

a++;

}

return clus;

}

private static DecimalFormat df1 = new DecimalFormat(".#");

public static void main(String[] args) {

String csvFile = "E:/Study/Data Mining Dataset/plot\_final\_2/child\_crime\_updated\_14thMarch.csv";

String m[] = new String[15];

String f[] = new String[15];

int male[] = new int[14];

int female[] = new int[14];

double matrix[][] = new double[14][14];

int i = 0;

int index\_x = 0 , index\_y = 0;

String line = "";

String cvsSplitBy = ",";

try (BufferedReader br = new BufferedReader(new FileReader(csvFile)))

{

while ((line = br.readLine()) != null)

{

String[] temp = line.split(cvsSplitBy);

m[i] = temp[2];

f[i] = temp[3];

i++;

}

i = 0;

for(int j = 1; j<15 ; j++)

{

male[i] = Integer.parseInt(m[j]);

female[i] = Integer.parseInt(f[j]);

i++;

}

i = 0;

//Getting euclidean distance from the function

double[] distance = euclidean\_distance(male,female);

String alpha[] = {"Andra Pradesh","Assam","Chattisgharh","Gujrat","Haryana","Karnataka","Kerla","Madhya Pradesh","Maharashtra","Odhisa","Rajisthan","Telangana","Uttar Pradesh","Delhi","N"};

int count = 0;

// Creating euclidean distance matrix

System.out.println("Euclidean Distance \n");

for(i = 0 ; i<14 ; i++)

{

for(int j = 0 ; j<14 ; j++)

{

matrix[i][j] = Double.parseDouble(df1.format(distance[count]));

System.out.print(matrix[i][j] + "\t");

count++;

}

System.out.println("\n");

}

count = 0;

int s\_r = matrix.length;

int s\_c = matrix[0].length;

int[][] clusters\_index = clusters(matrix,s\_r,s\_c);

int clus\_a = 0;

int clus\_b = 0;

String dendo[] = new String[clusters\_index.length];

String dd = "";

String dd\_temp = "";

System.out.println("The clusters that are produced from starting to last are ::: \n");

for(i = 0 ; i<clusters\_index.length ; i++)

{

clus\_a = clusters\_index[i][0];

clus\_b = clusters\_index[i][1];

System.out.println(alpha[clus\_a] + "-" + alpha[clus\_b]);

dd = alpha[clus\_a] + "-" + alpha[clus\_b];

dendo[i] = dd;

dd = "";

}

dendogram(dendo);

}

catch (IOException e)

{

e.printStackTrace();

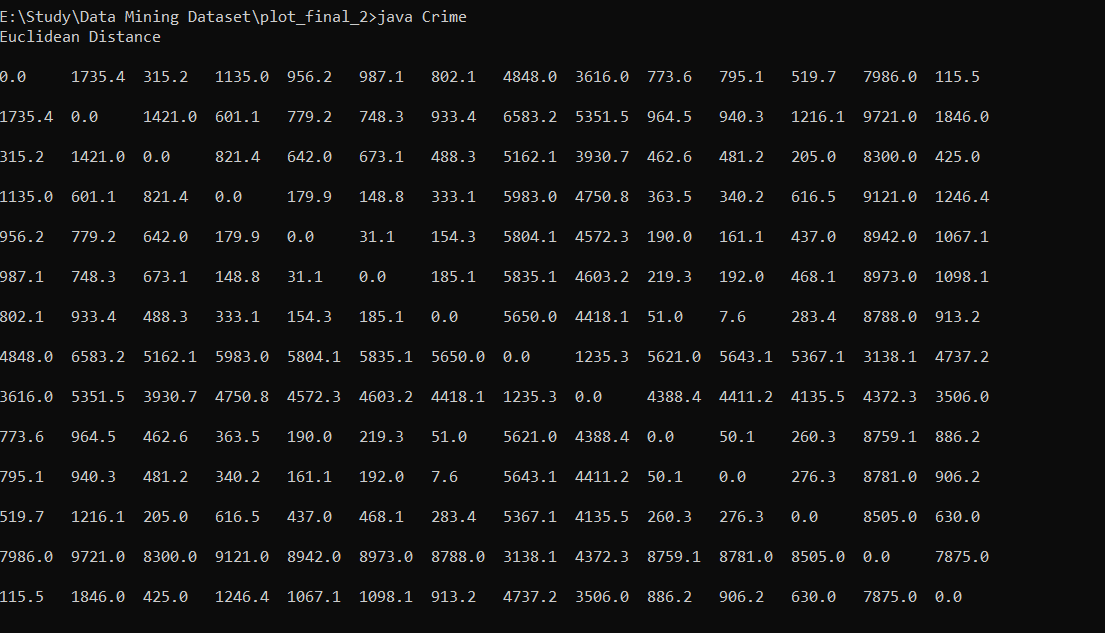
}

}

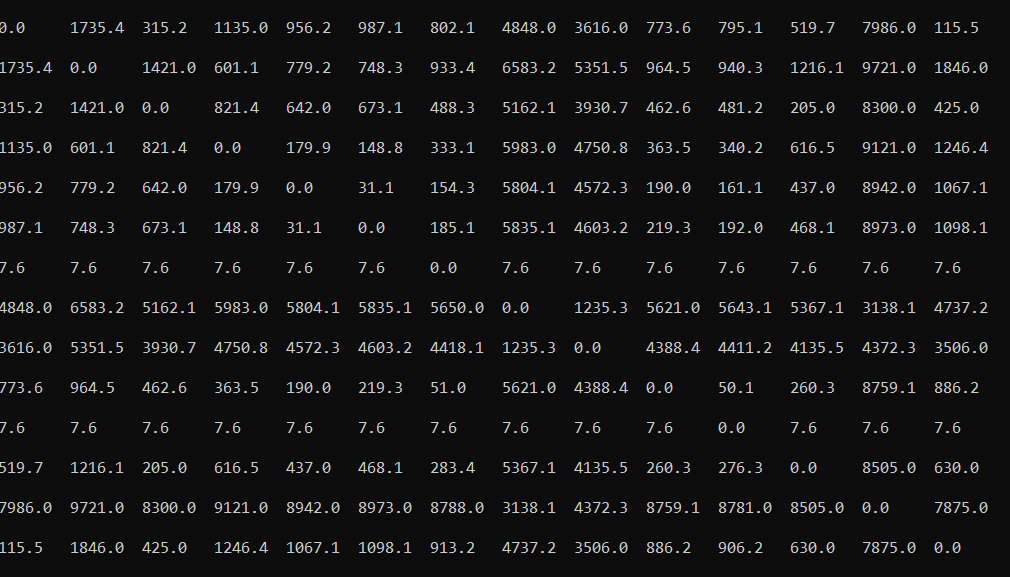
}

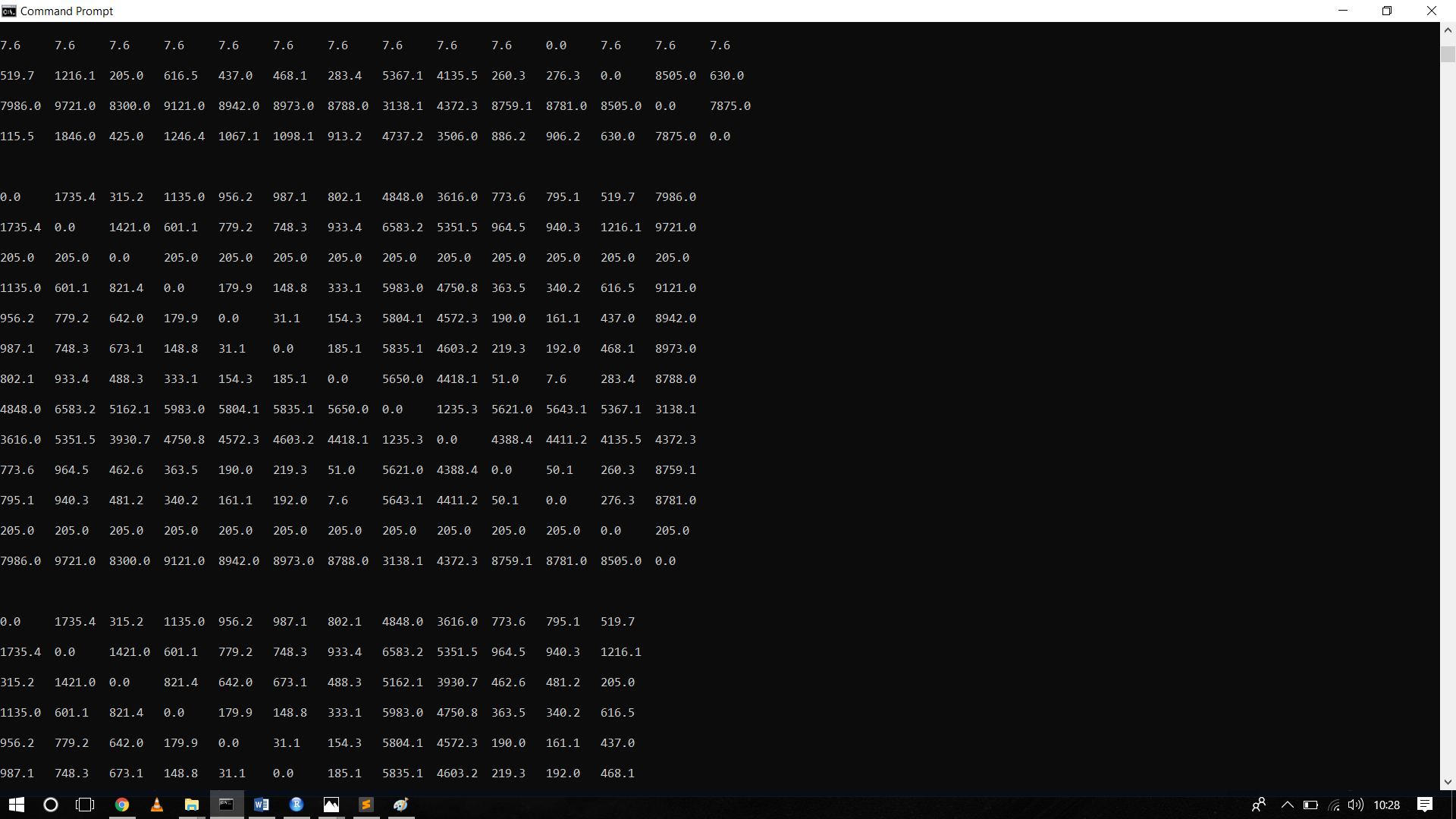
OUTPUTS :

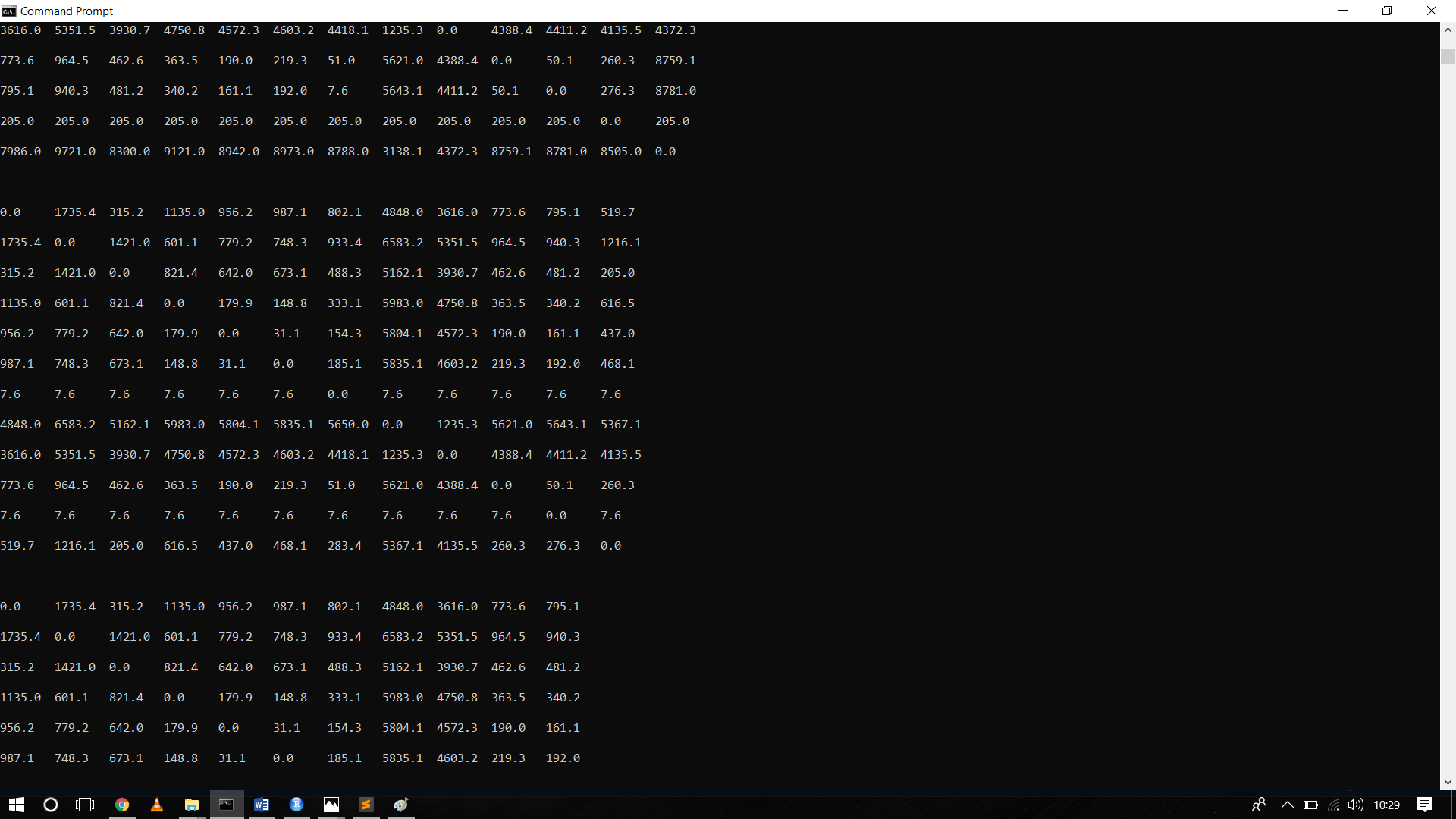
INITIAL EUCLIDEAN DISTANCE MATRIX:

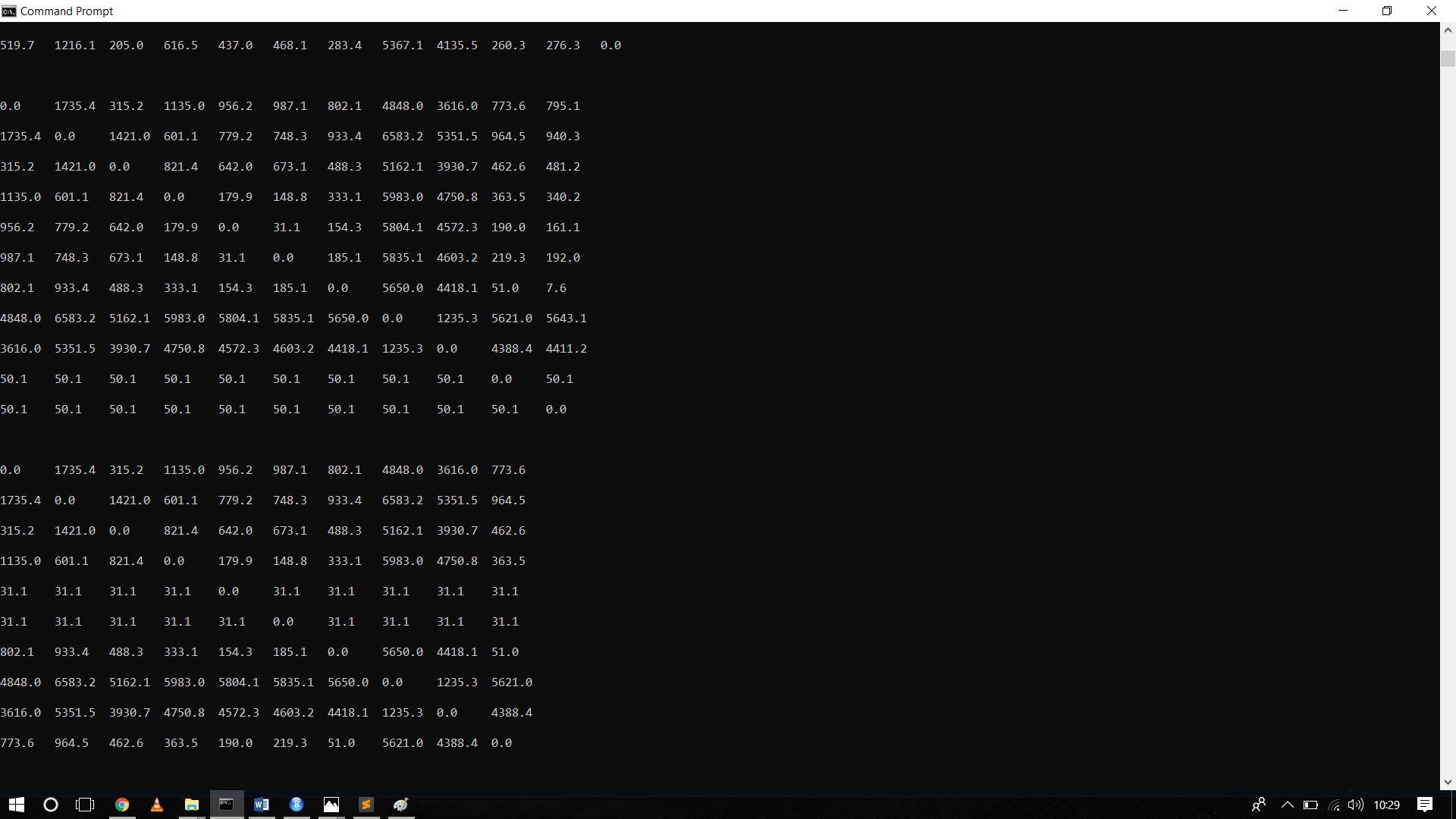


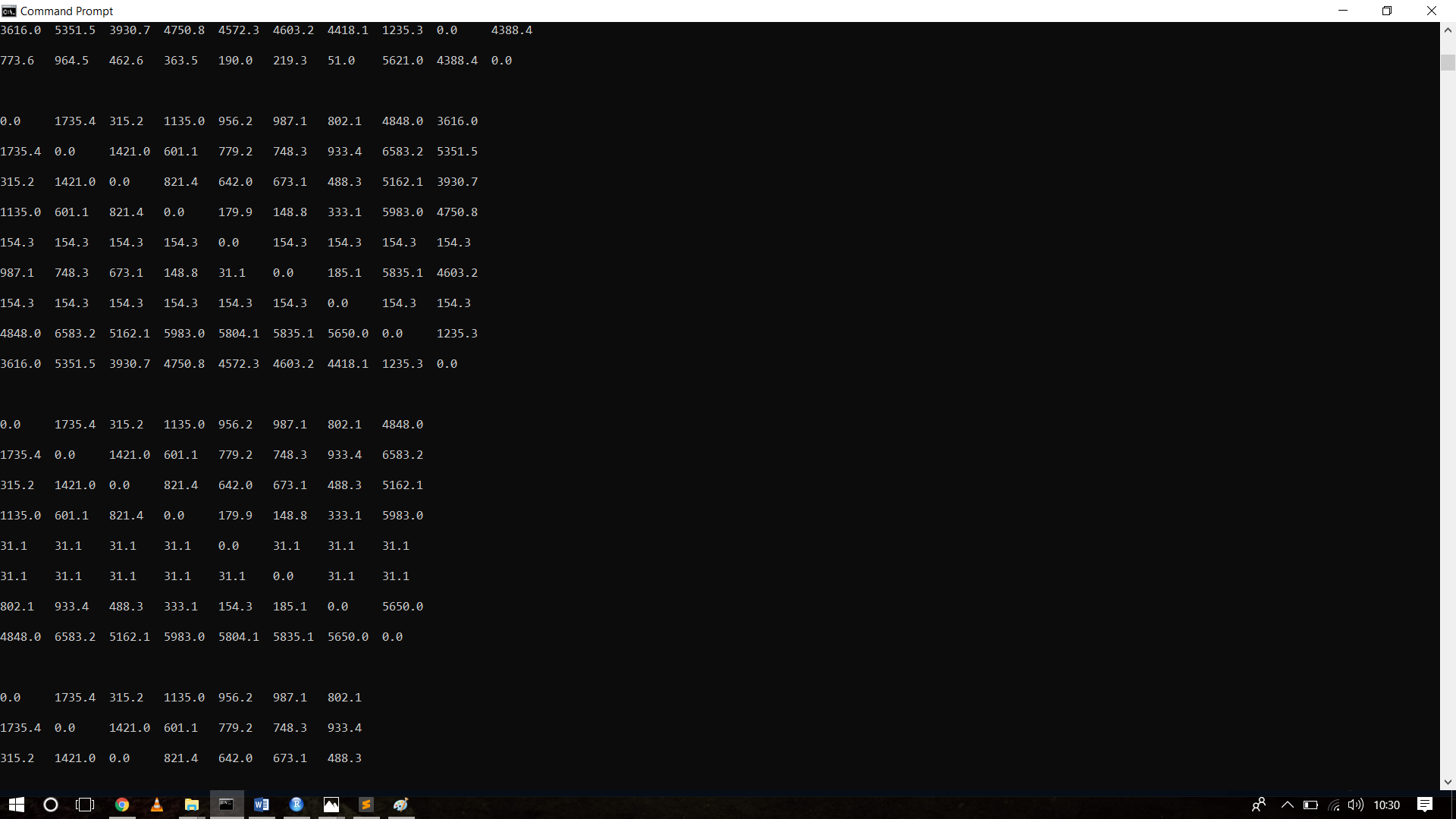
EUCLIDEAN DISTANCE MATRIX AFTER EACH ITERATION:

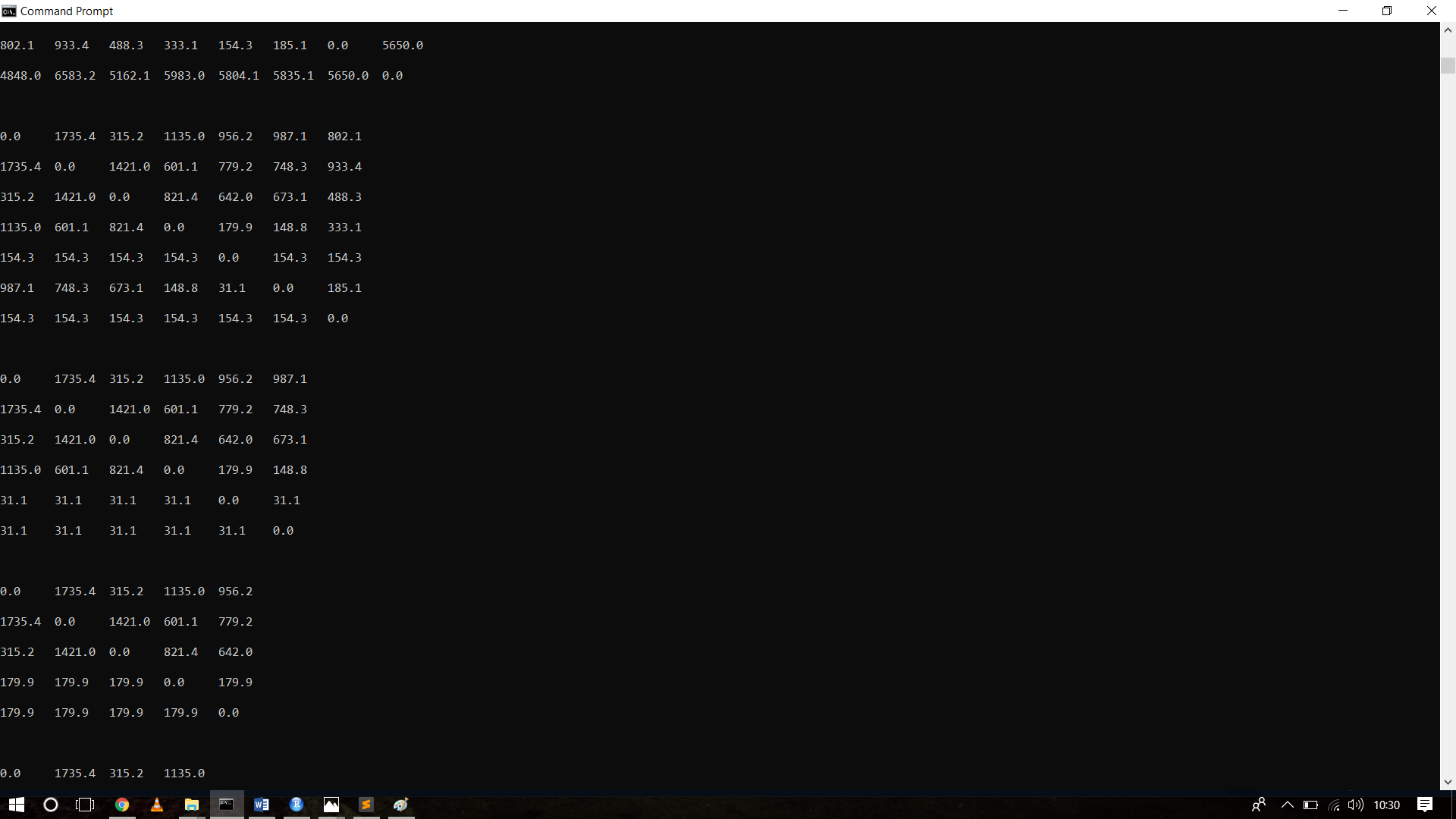


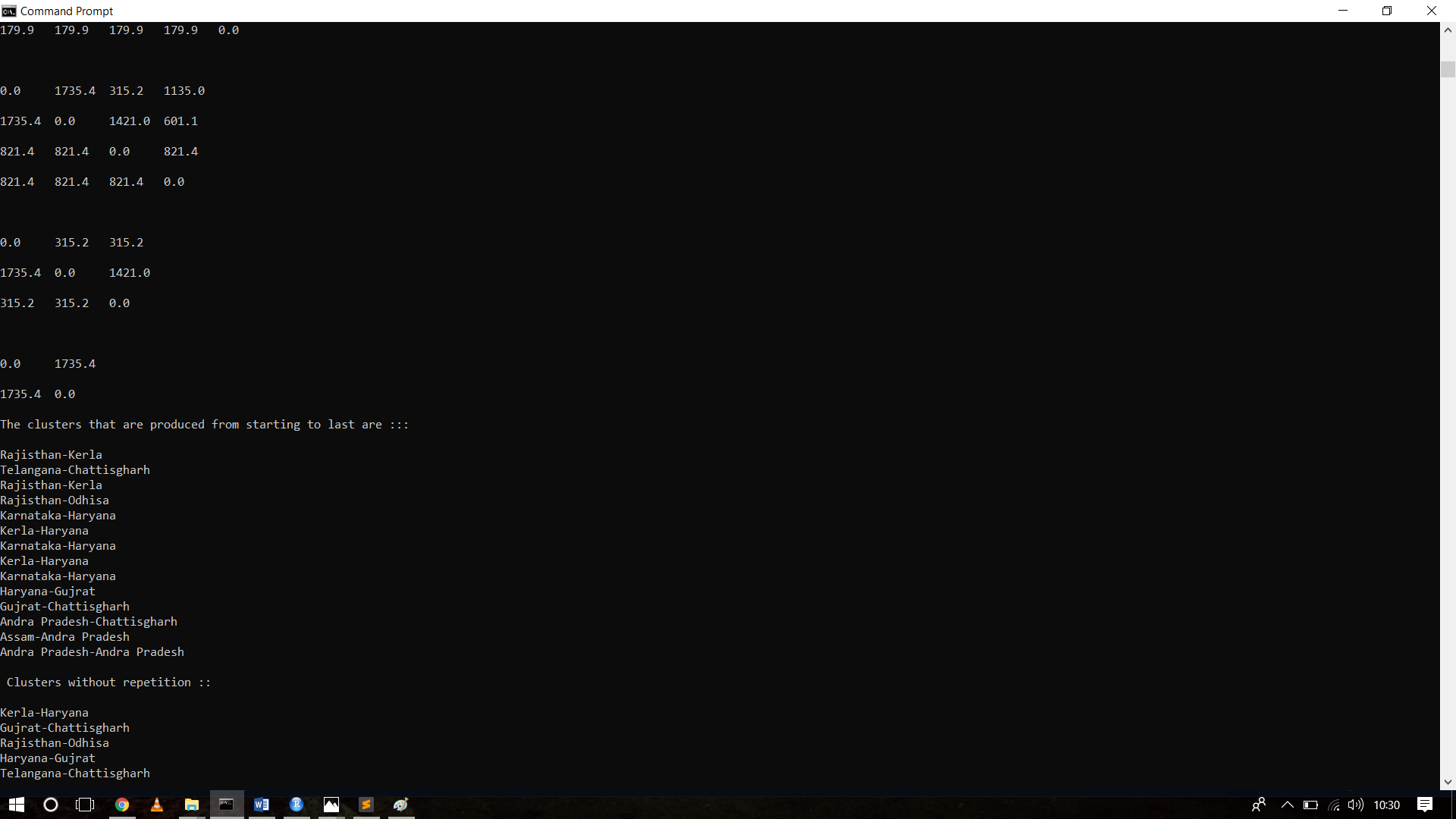




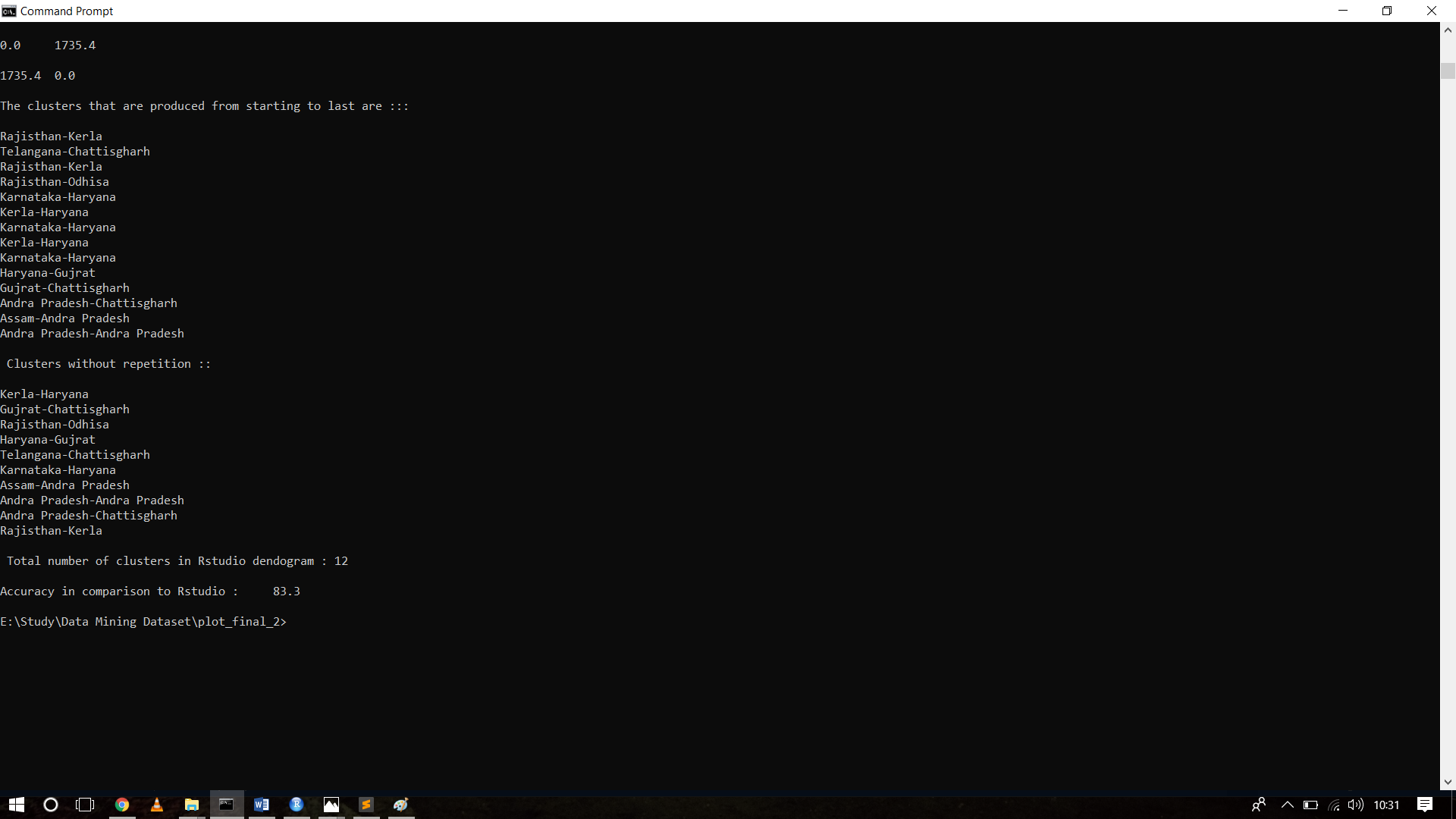








LIST OF CLUSTERS THAT ARE FORMED:



CONCLUSION:

By using clustering analysis we were able identify the states of India that are closely related to other states on the basis of crime committed on children.