PS5

Satyaki Basu Sarbadhikary

2023-03-28

Problem 1.

Use the dist() function to find the distance matrix using various distance functions for the French food data set. Interpret the distances.
Considering the weight matrix A = $diag(s^{-1}_{x_1x_1}, s^{-1}_{x_2x_2}, ..., , s^{-1}_{x_px_p})$, calculate the distance matrix.

In this dataset we are given with observations of food expenditure on 7 types of foods for 12 families. Let us denote the the observation of the i th family by $\tilde{xi}$ for i = 1(1)12.
To find out how close these $x\tilde{i}'s$ are from each other. We are going to find it out by using the distance measures.
The following matrix shows distances between different points. Here we have taken the "Euclidean Distance" to measure the distances between two points. The Euclidean distance between two points xi and xj is given by -

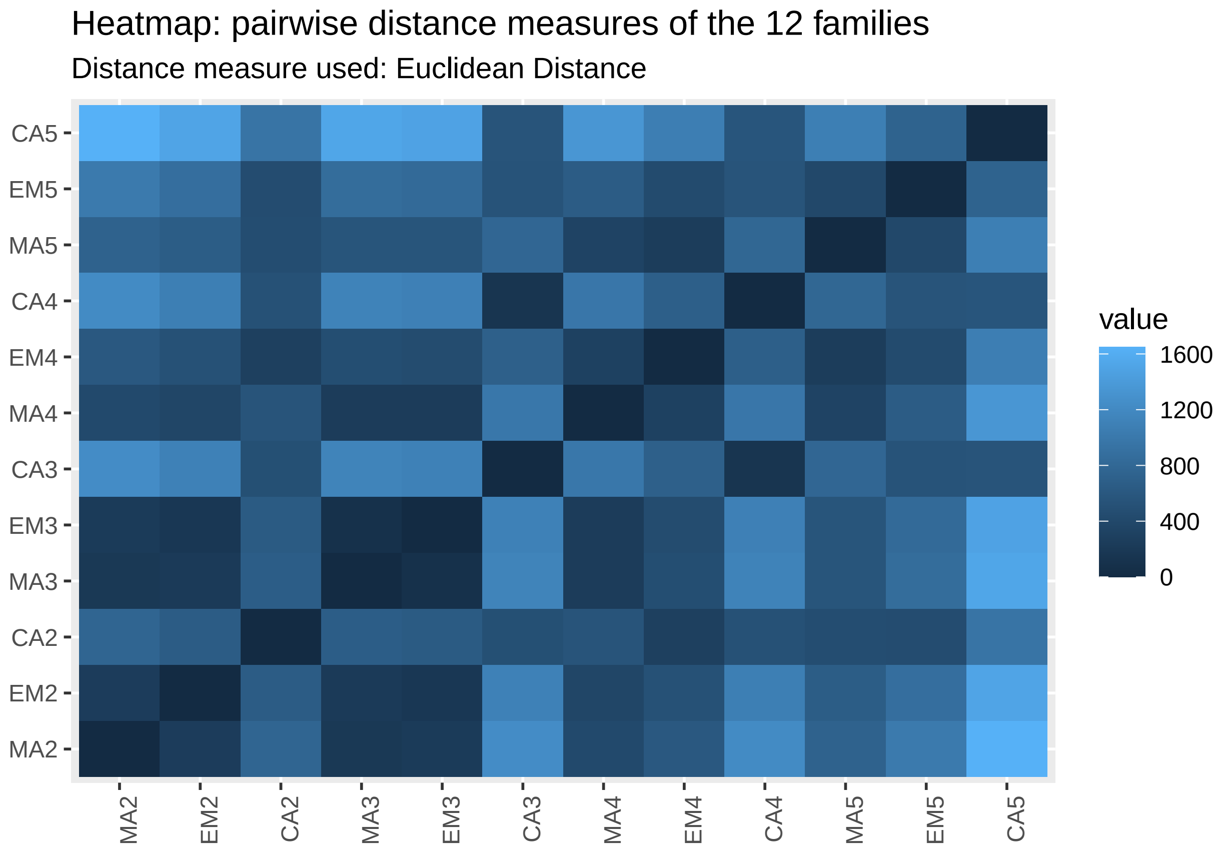
$$d_{ij} = \sqrt{\sum_{k=1}^{7}(x_{ik} - x_{jk})^2}, \qquad i, j = 1(|)7$$

```
library(proxy)
df=read.csv('food_data.csv')
X=as.matrix(df[-1])
d=as.matrix(proxy::dist(X,method = "Euclidean",upper = TRUE))
colnames(d)=df[,1]
rownames(d)=df[,1]; round(d,1)
```

```
##        MA2    EM2   CA2    MA3    EM3    CA3    MA4    EM4    CA4    MA5    EM5
## MA2    0.0  241.3 762.8  188.2  226.9 1230.6  411.2  601.3 1216.5  720.0 1012.7
## EM2  241.3    0.0 646.0  213.0  171.2 1098.1  367.7  503.8 1078.5  660.9  876.4
## CA2  762.8  646.0   0.0  664.4  633.2  491.2  547.3  285.8  505.7  456.2  450.6
## MA3  188.2  213.0 664.4    0.0   87.4 1130.7  237.0  465.9 1118.0  558.6  854.2
## EM3  226.9  171.2 633.2   87.4    0.0 1100.2  238.7  445.5 1089.8  555.2  820.9
## CA3 1230.6 1098.1 491.2 1130.7 1100.2    0.0  982.7  694.0  134.1  777.9  537.5
## MA4  411.2  367.7 547.3  237.0  238.7  982.7    0.0  303.4  974.0  332.7  649.0
## EM4  601.3  503.8 285.8  465.9  445.5  694.0  303.4    0.0  685.2  248.3  433.1
## CA4 1216.5 1078.5 505.7 1118.0 1089.8  134.1  974.0  685.2    0.0  781.5  544.2
## MA5  720.0  660.9 456.2  558.6  555.2  777.9  332.7  248.3  781.5    0.0  397.8
## EM5 1012.7  876.4 450.6  854.2  820.9  537.5  649.0  433.1  544.2  397.8    0.0
## CA5 1648.7 1506.2 941.4 1521.7 1485.7  543.7 1341.0 1063.1  564.4 1077.5  733.3
##        CA5
## MA2 1648.7
## EM2 1506.2
## CA2  941.4
## MA3 1521.7
## EM3 1485.7
## CA3  543.7
## MA4 1341.0
## EM4 1063.1
## CA4  564.4
## MA5 1077.5
## EM5  733.3
## CA5    0.0
```

To get a visual idea, we have used the heat-map of the distance matrix obtained below -

```
library(tidyverse)
d = reshape2::melt(d)
ggplot(data = d, aes(x = Var1, y = Var2, fill = value))+
  geom_tile()+
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
    axis.title = element_blank())+
  labs(title = "Heatmap: pairwise distance measures of the 12 families",
    subtitle = "Distance measure used: Euclidean Distance")
```



Comment
1. From the heat-map it seem that the following families are “closer” or “similar” with respect to their food expenditure pattern -
MA2 with EM2, MA3, EM3;
EM2 with MA3, EM3, MA4;

CA2 with EM4, EM5;
MA3 with MA4;
MA4 with EM4, EM5;
MA5 with EM5;

We have also obtained the heat-map of distance between the families using “Manhattan distance” as a distance measures. The Manhattan distance between two points xi and xj is given by -

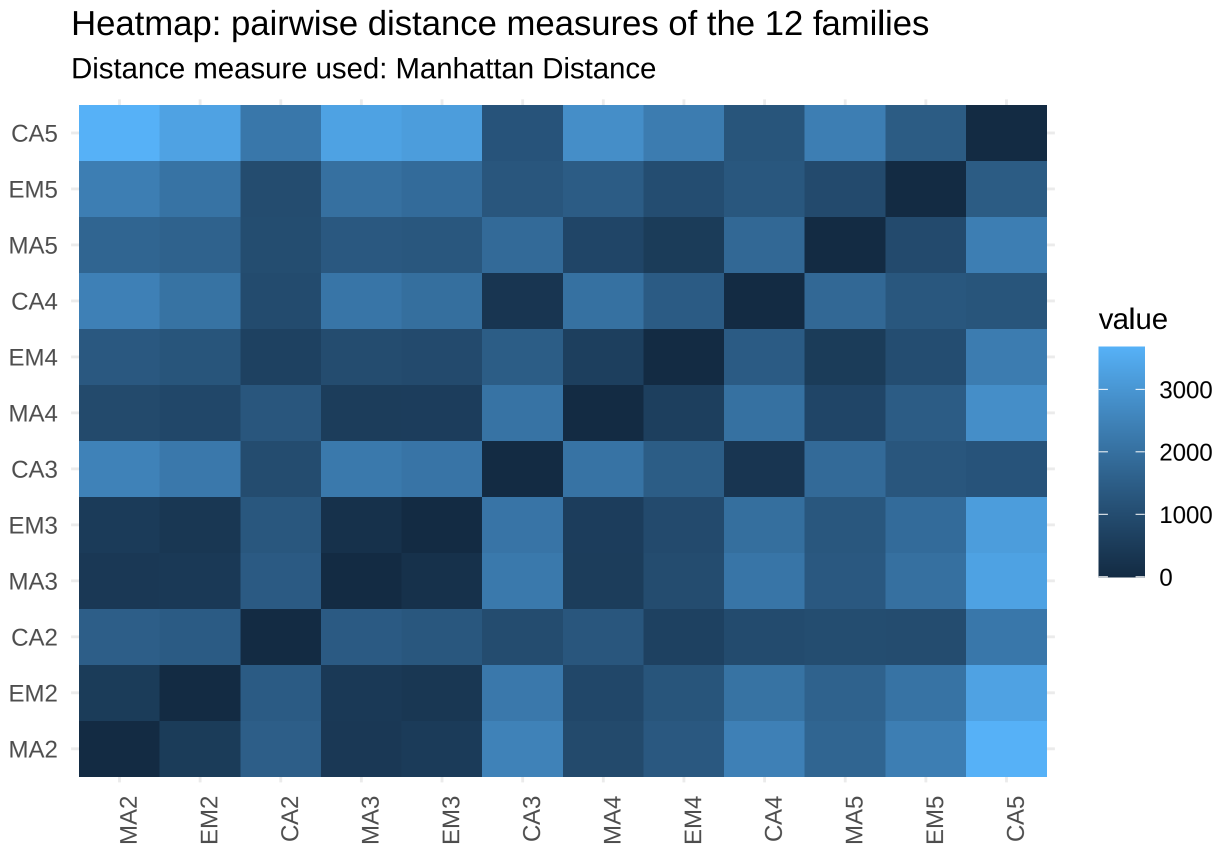
$$d_{ij} = \sum_{k=1}^{7} |x_{ik} - x_{jk}|, \qquad i, j = 1(|)7$$

The heat-map is given below -

```
d = as.matrix(proxy::dist(X, method = "Manhattan",upper = TRUE))
colnames(d) = df[,1]
rownames(d) = df[,1]

d = reshape2::melt(d)

ggplot(data = d, aes(x = Var1, y = Var2, fill = value))+
  geom_tile()+
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
    axis.title = element_blank())+
  labs(title = "Heatmap: pairwise distance measures of the 12 families",
    subtitle = "Distance measure used: Manhattan Distance")
```



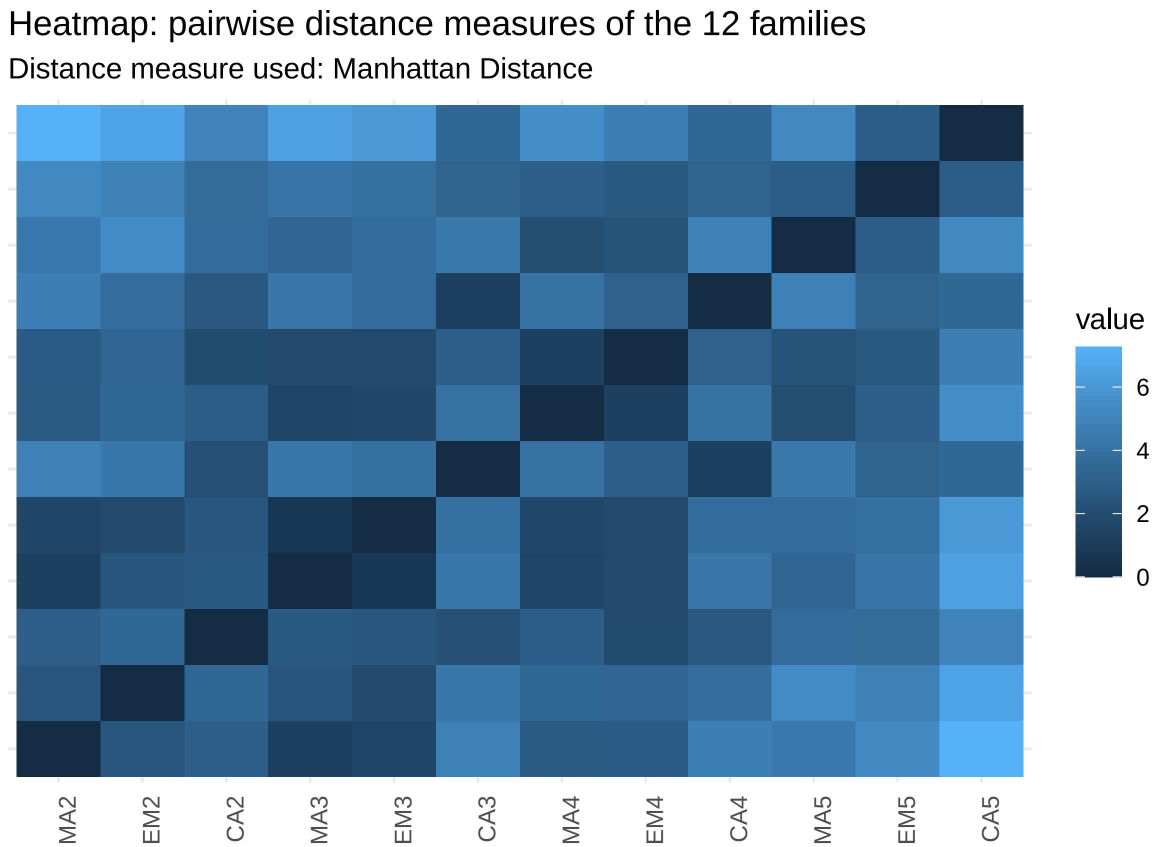
Comment
1. In case of Manhattan distance, the relative distance between the observations is not changing. But the scale of distance has been increased in the case of Manhattan distance.

Now we are going to scale down the measures of expenditures, and find out the distance matrix. The distance (distance as "Euclidean distance") heat map of the distance matrix of scaled component is given below -

```
X = scale(X)
d = as.matrix(proxy::dist(X, method = "Euclidean",upper = TRUE))
colnames(d) = df[,1]
rownames(d) = df[,1]

d = reshape2::melt(d)

ggplot(data = d, aes(x = Var1, y = Var2, fill = value))+
  geom_tile()+
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
    axis.title = element_blank())+
  labs(title = "Heatmap: pairwise distance measures of the 12 families",
    subtitle = "Distance measure used: Manhattan Distance")
```



Comment
1. After being scaled down, the scale of distances have been lowered to a significant amount. 2.But the relative picture of closeness has been changed in this case.Now the following families are "close" with respect to their food expenditure pattern.
MA2 with MA3, EM3;

EM2 with EM3;
MA3 with EM3, EM4, MA4;
CA with CA4;
MA4 with EM4, MA5.

Problem 2.

Define a new set of binary data based on the criterion

$$y_{ij} = \begin{cases} 1, & \text{if } x_{ij} > \bar{x}_i \\ 0, & \text{otherwise} \end{cases}$$

for i = 1(|) n and k = 1(|)p. This means that we transform the observations of the k-th variable to 1 if it is larger than the mean value of all observations of the k^{th} variable. Calculate the similarity (distance) measures for the binary data.

Here we are going to transform the French food data set. We are defining a new data matrix Y=(y_{ij}),where

$$y_{ij} = \begin{cases} 1, & \text{if } x_{ij} > \bar{x}_i \\ 0, & \text{otherwise} \end{cases}$$

where X = (x_{ij}) being the dataset for the original french food data set. lets transform the dataset. The tranformed Y dataset is given below -

```
transform = function(x){
  m = mean(x)
  return(ifelse(x>m,1,0))
}
Y = apply(X, 2, transform)
rownames(Y) = df[,1]
Y
```

```
##     bread vegetables fruits meat poultry milk wine
## MA2     0          0      0    0       0    0    1
## EM2     0          0      0    0       0    0    0
## CA2     0          1      1    1       1    0    1
## MA3     0          0      0    0       0    0    1
## EM3     0          0      0    0       0    0    0
## CA3     0          1      1    1       1    0    0
## MA4     1          0      0    0       0    1    1
## EM4     1          0      0    0       0    1    1
## CA4     0          1      1    1       1    0    0
## MA5     1          1      0    0       0    1    1
## EM5     1          1      1    1       1    1    0
## CA5     1          1      1    1       1    1    0
```

Now we are going to use some similarity measure on the families using this transformed dataset. The following table shows the pairwise similarity score, using Jaccard's similarity as a measure of proximity. Where proximity between two binary variables $\tilde{x_i}$ and $\tilde{x_j}$ is given by -

$$\frac{a1}{a1 + a2 + a3}$$

where $a1 = \sum_{k=1}^{p} I(x_{ik} = 0, x_{jk} = 0)$,
$a2 = \sum_{k=1}^{p} I(x_{ik} = 0, x_{jk} = 1)$ and
$a3 = \sum_{k=1}^{p} I(x_{ik} = 1, x_{jk} = 0)$

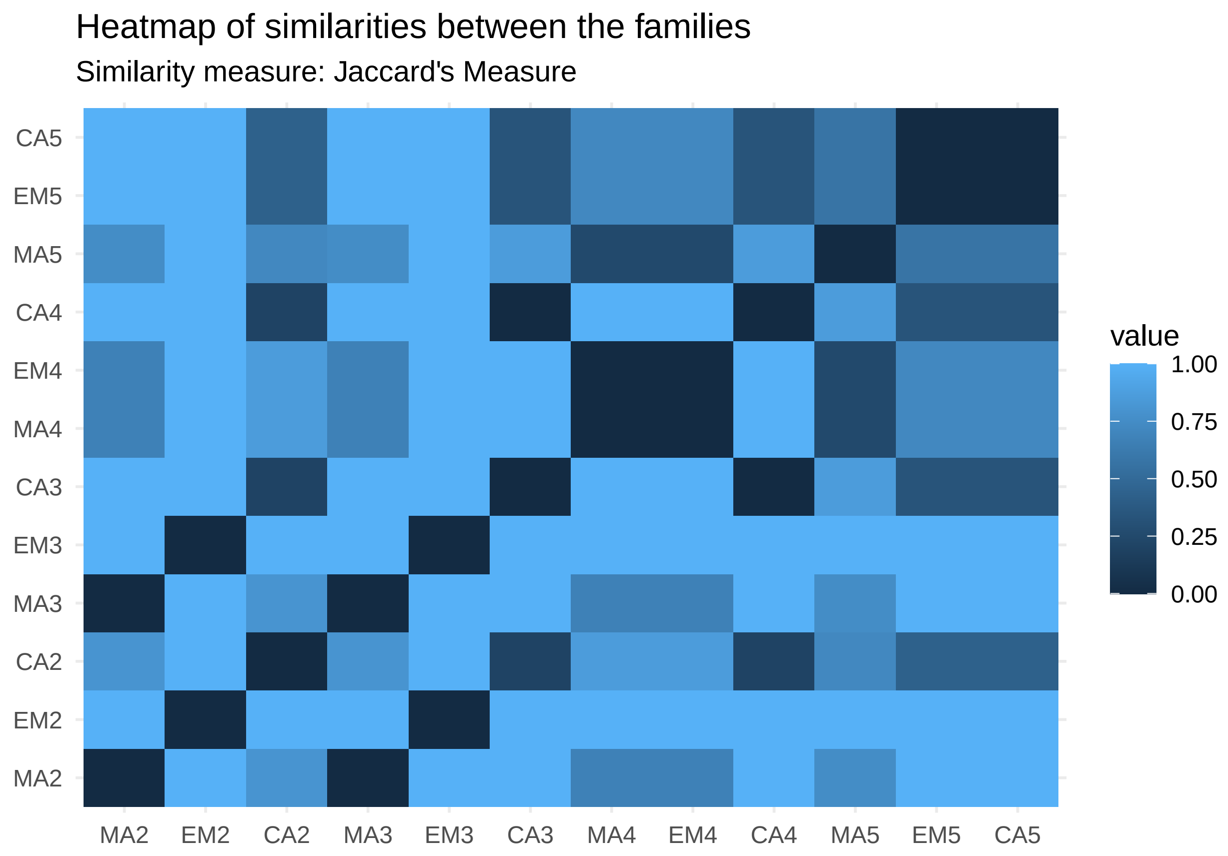
Following is the table showing the Jaccard's measure of similarity between the families.

```
d = as.matrix(proxy::dist(Y, method = "Jaccard"))
round(d, 3)
```

```
##       MA2 EM2   CA2   MA3 EM3   CA3   MA4   EM4   CA4   MA5   EM5   CA5
## MA2 0.000   1 0.800 0.000   1 1.000 0.667 0.667 1.000 0.750 1.000 1.000
## EM2 1.000   0 1.000 1.000   0 1.000 1.000 1.000 1.000 1.000 1.000 1.000
## CA2 0.800   1 0.000 0.800   1 0.200 0.857 0.857 0.200 0.714 0.429 0.429
## MA3 0.000   1 0.800 0.000   1 1.000 0.667 0.667 1.000 0.750 1.000 1.000
## EM3 1.000   0 1.000 1.000   0 1.000 1.000 1.000 1.000 1.000 1.000 1.000
## CA3 1.000   1 0.200 1.000   1 0.000 1.000 1.000 0.000 0.857 0.333 0.333
## MA4 0.667   1 0.857 0.667   1 1.000 0.000 0.000 1.000 0.250 0.714 0.714
## EM4 0.667   1 0.857 0.667   1 1.000 0.000 0.000 1.000 0.250 0.714 0.714
## CA4 1.000   1 0.200 1.000   1 0.000 1.000 1.000 0.000 0.857 0.333 0.333
## MA5 0.750   1 0.714 0.750   1 0.857 0.250 0.250 0.857 0.000 0.571 0.571
## EM5 1.000   1 0.429 1.000   1 0.333 0.714 0.714 0.333 0.571 0.000 0.000
## CA5 1.000   1 0.429 1.000   1 0.333 0.714 0.714 0.333 0.571 0.000 0.000
```

Let's draw the heatmap of the given table.

```
d = reshape2::melt(d)
ggplot(data = d, aes(x = Var1, y = Var2, fill = value))+
  geom_tile()+
  theme_minimal()+
  theme(axis.title = element_blank())+
  labs(title = "Heatmap of similarities between the families",
    subtitle = "Similarity measure: Jaccard's Measure")
```



Comment
Here most of the families seemed to have a very high similarity, except some few pairs of families like - MA3 and MA2 or EM3 and EM2, CA4 and CA3.

Problem 3.

Based on the variables: weekly average time spent in listening to music, reading story books, leisure time with friends, divide the students into different groups using agglomerative clustering based on
a. Single linkage b. Complete linkage c. Average linkage (both weighted and unweighted)

Given dataset on weekly average time spent in listening to music, reading story books and leisure time spent with the friends. Lets us first take a glance at the dataset.

```
rm(list=ls())
setwd("G:/My Drive/Semester 2/Paper 2/Practical")
df = read.csv("PS5.csv")
colnames(df) = c("names", "rollNumbers", "x1", "x2", "x3")
head(df)
```

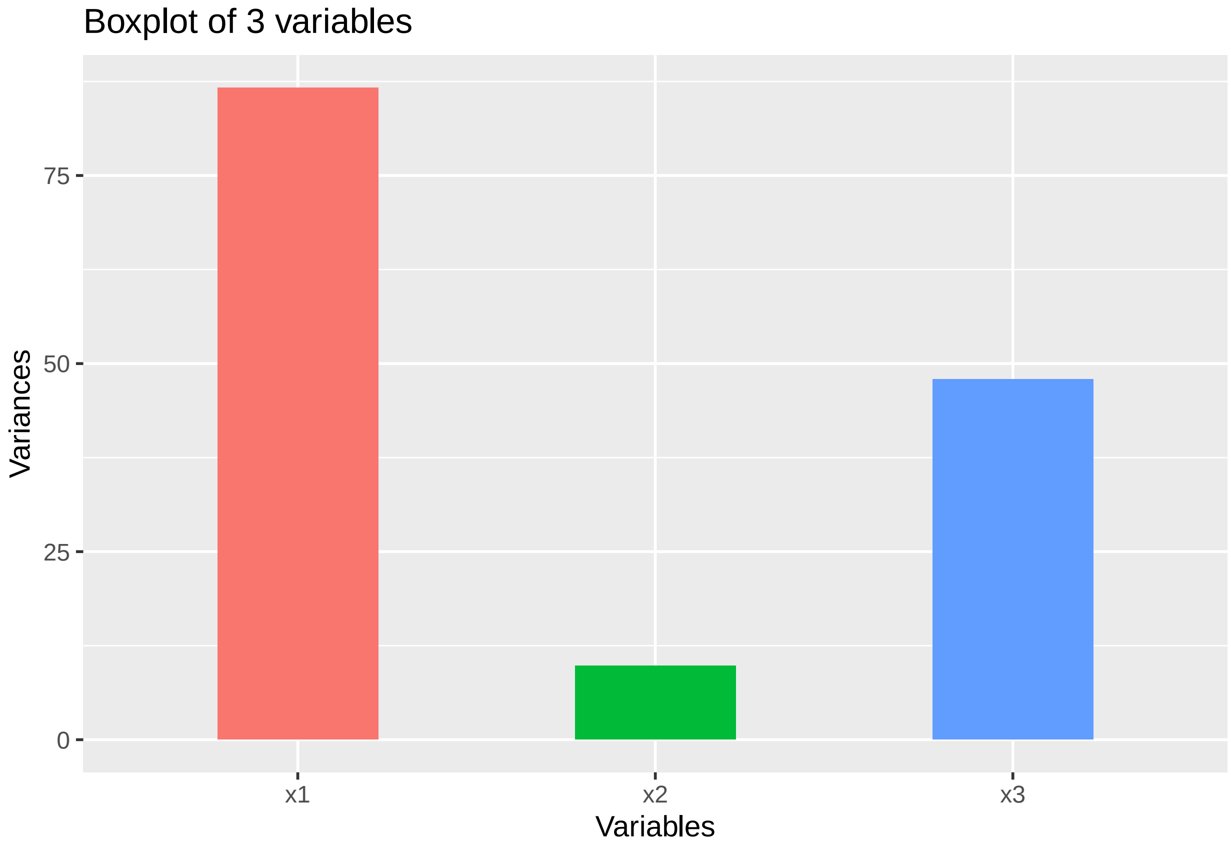
```
##                      names rollNumbers   x1   x2 x3
## 1 Pratyusha Mukhyopadhyay         402 14.0  7.0  3
## 2             Gourav Daga         403  6.0  2.0 10
## 3           Spandan Ghosh         404  4.0  0.0  8
## 4             Anuroop Roy         405 10.0  2.0 21
## 5         Shamie Dasgupta         406  5.5 10.5  5
## 6             Aishani Dey         407 15.0  0.0 30
```

```
dim(df)
```

```
## [1] 28  5
```

The dataset have the records on 28 students. Let us define the following variables x1 = Weekly average time spent in listening to music in hrs, by a randomly selected student. x2 = Weekly average time spent in reading story books in hrs, by a randomly selected student. x3 = Weekly average leisure time spent with friends in hrs, by a randomly selected student. lets check out the degree of variablity of these three variables on the data set.

```
df %>%
  gather(-c(names, rollNumbers), key = "Legend", value = "value") %>%
  group_by(Legend) %>%
  summarise(Variances = var(value)) %>%
  ggplot(aes(x = Legend, y = Variances))+
  geom_col(aes(fill = Legend), width = 0.45)+
  guides(fill = "none")+
  labs(x = "Variables", y = "Variances",
    title = "Boxplot of 3 variables")
```



Comment
The variability of the three variables are far apart from each others. We should apply scaling down to the variables so that they all have the same variances.
Let scale down the variables -

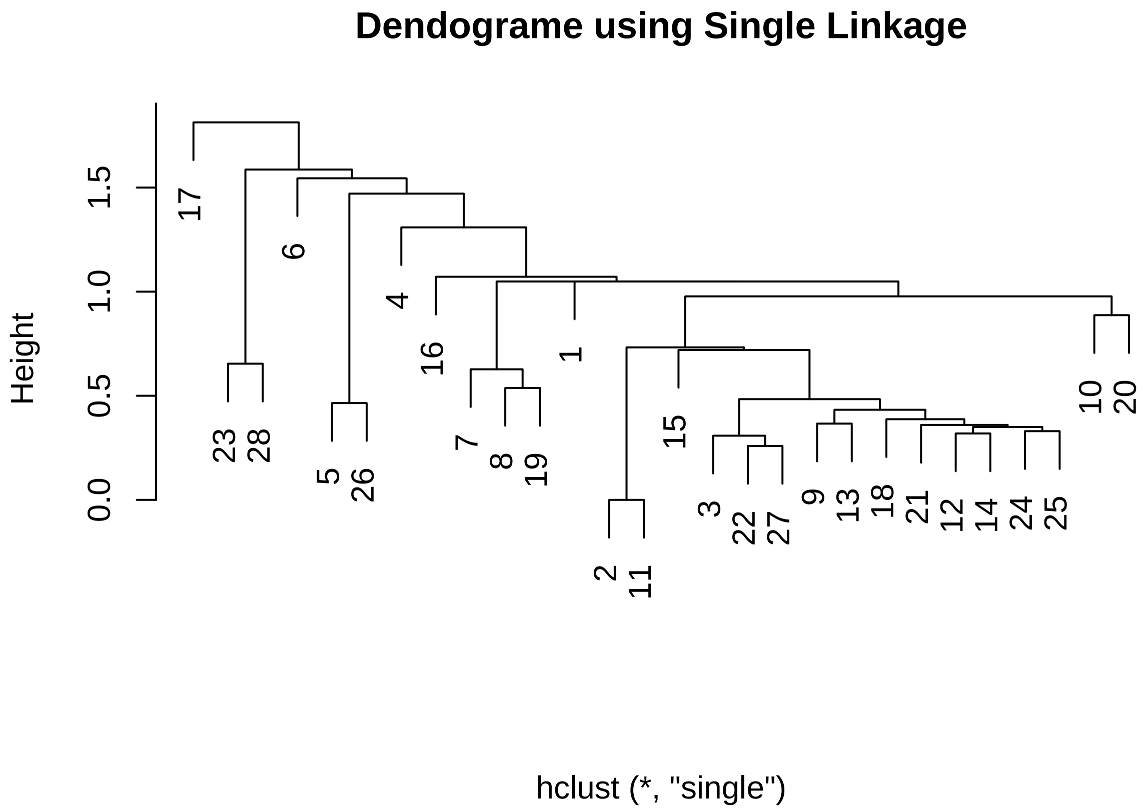
```
df[,c("x1", "x2", "x3")] = scale(df[,c("x1", "x2", "x3")]); head(df)
```

```
##                      names rollNumbers         x1         x2          x3
## 1 Pratyusha Mukhyopadhyay         402  0.3126591  1.2531445 -0.78683198
## 2             Gourav Daga         403 -0.5466739 -0.3417667  0.22444060
## 3           Spandan Ghosh         404 -0.7615071 -0.9797311 -0.06449442
## 4             Anuroop Roy         405 -0.1170074 -0.3417667  1.81358321
## 5         Shamie Dasgupta         406 -0.6003822  2.3695823 -0.49789696
## 6             Aishani Dey         407  0.4200757 -0.9797311  3.11379080
```

Now we are going to apply the hierarchial agglomerative techniques to cluster them. We have applied the agglomerative techniques to cluster the invidual students in the dataset, according to the time spend by them in listening music, time spend on story books and leisure time spend with friends. we have drawn the Densogrames of the dataset clustering. The densogrames are given below.

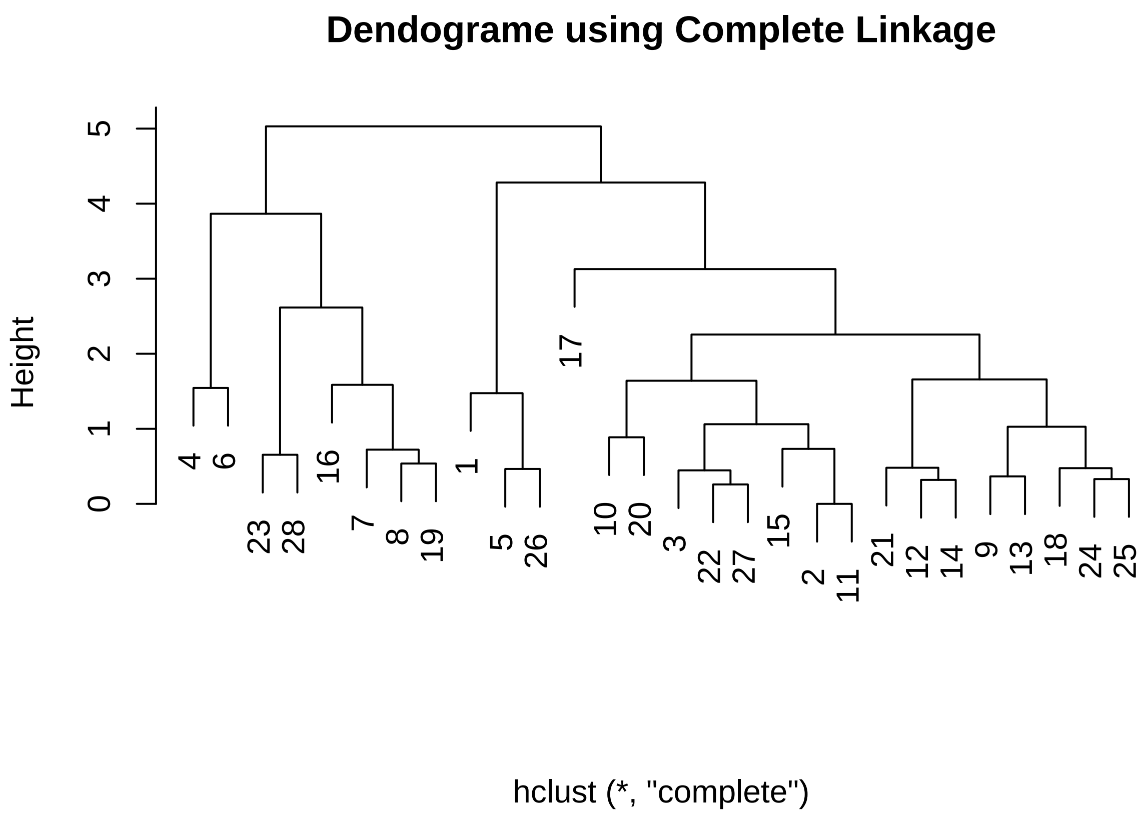
Using Single Linkage

```
library(stats)
d = dist(df[,c("x1", "x2", "x3")])
hc1 = hclust(d, method = "single")
plot(hc1,main = "Dendograme using Single Linkage", xlab = "")
```



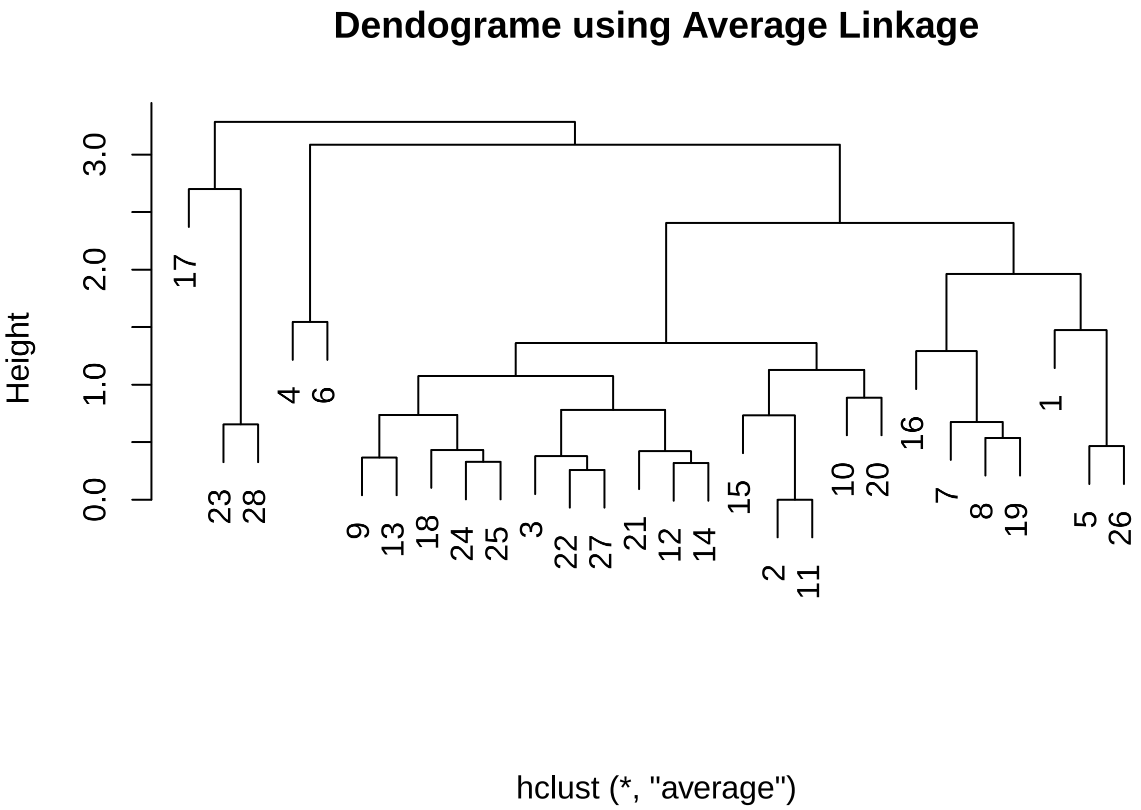
Using Complete Linkage

```
d = dist(df[,c("x1", "x2", "x3")],diag=TRUE)
hc2 = hclust(d, method = "complete")
plot(hc2,main = "Dendograme using Complete Linkage", xlab = "")
```



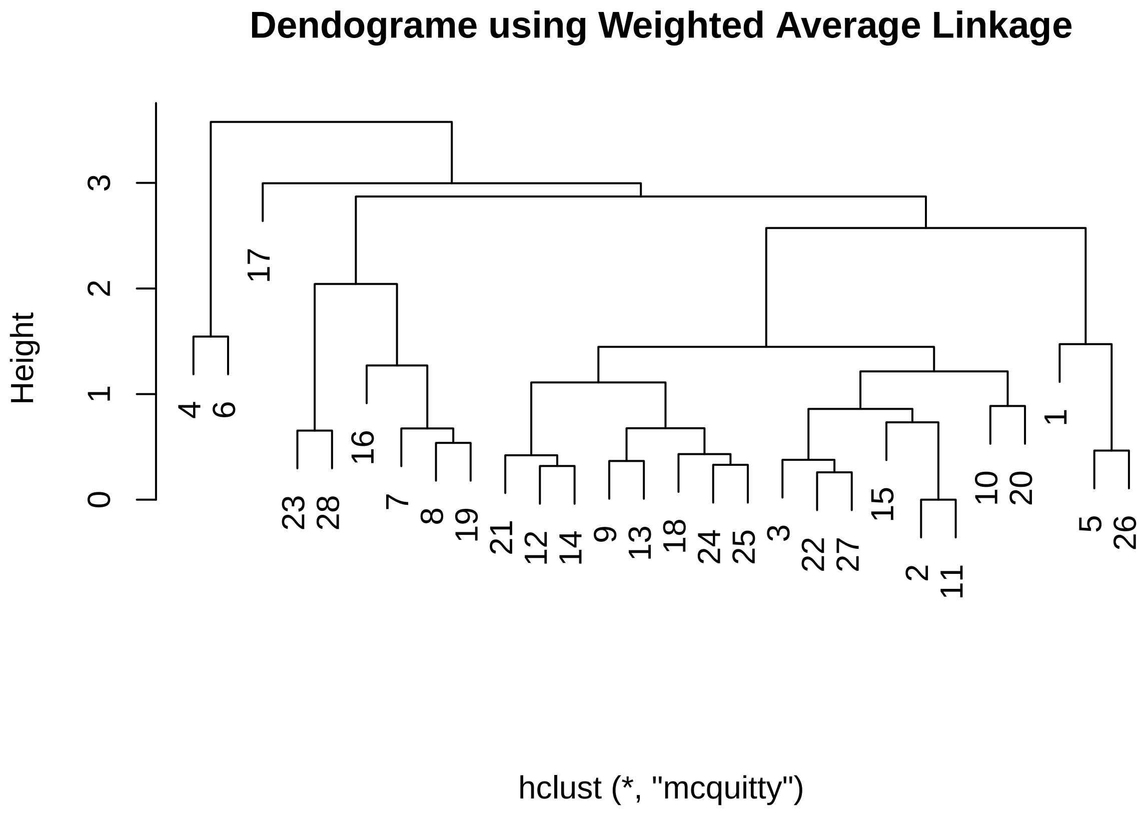
Using Simple Average Linkage

```
d = proxy::dist(df[,c("x1", "x2", "x3")])
hc3 = hclust(d, method = "average")
plot(hc3,
main = "Dendograme using Average Linkage", xlab = "")
```



Using Weighted Average Linkage

```
d = proxy::dist(df[,c("x1", "x2", "x3")])
hc4 = hclust(d, method = "mcquitty")
plot(hc4, main = "Dendograme using Weighted Average Linkage", xlab = "")
```



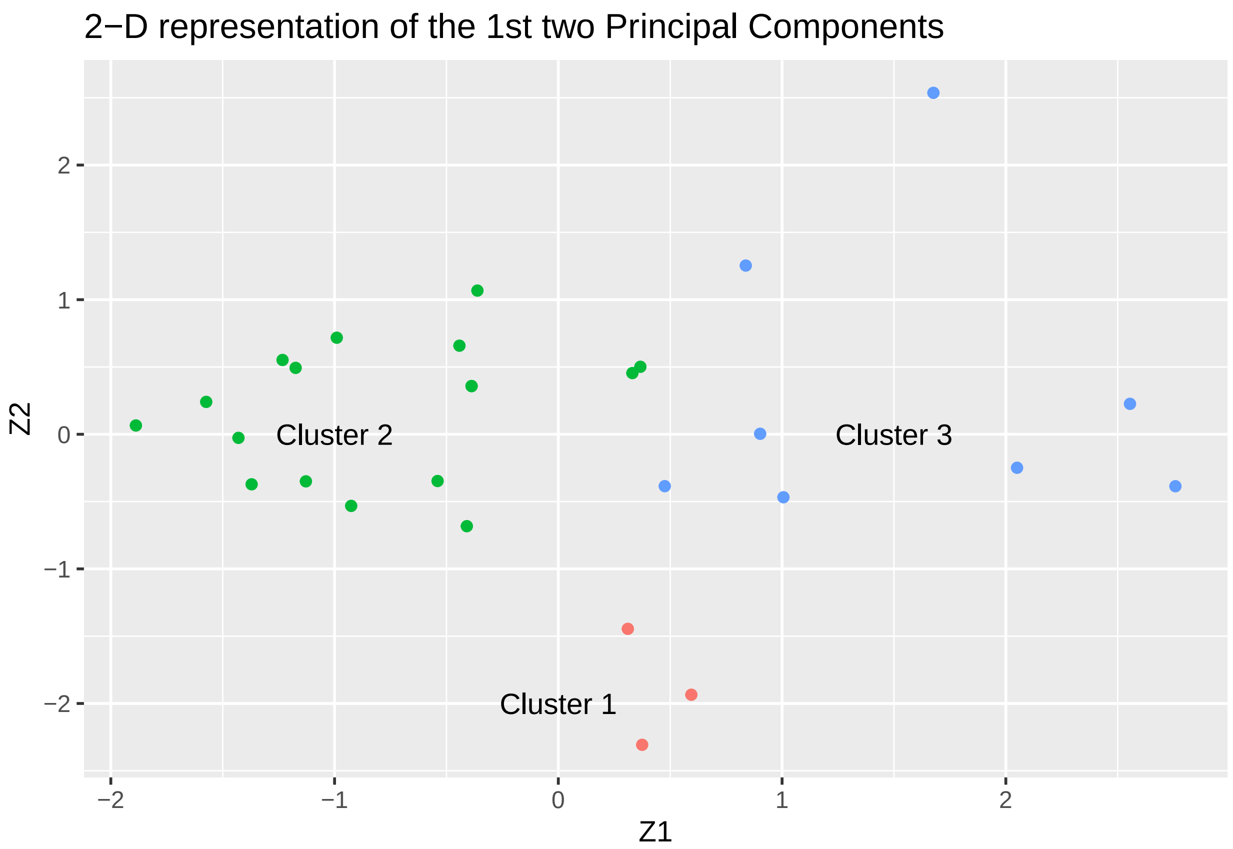
Comment The dendogram for the clustering using the "Complete Linkage" seems more rational than the others. We are going to classify the students in 3 clusters. Now, we are going to represent the students with clusters in a 2-D representation using Principle Component Analysis.

```
eigen_vals = eigen(cov(df[c("x1", "x2", "x3")]))$values
perc = 100*cumsum(eigen_vals)[2]/sum(eigen_vals); perc
```

```
## [1] 81.6325
```

Here, the first two Principle Components explaines about 81.63% of the total variance of the dataset. The 2D representation of the data is given below

```
U = eigen(cov(df[c("x1", "x2", "x3")]))$vectors
df[,c("Z1", "Z2")] = as.matrix(df[,c("x1", "x2", "x3")])%*%U[,1:2]
df[, "cluster"] = cutree(hc2, k = 3)
df %>%
ggplot(aes(x = Z1, y = Z2))+
  geom_point(aes(color = as.factor(cluster)))+
  guides(color = "none")+
  annotate("text", x = 0, y = -2, label = "Cluster 1")+
  annotate("text", x = -1, y = 0, label = "Cluster 2")+
  annotate("text", x = 1.5, y = 0, label = "Cluster 3")+
  labs(title = "2-D representation of the 1st two Principal Components")
```



Comment The clustering seems visually appealing in the representation. The student names according to the cluster is given in the following table -

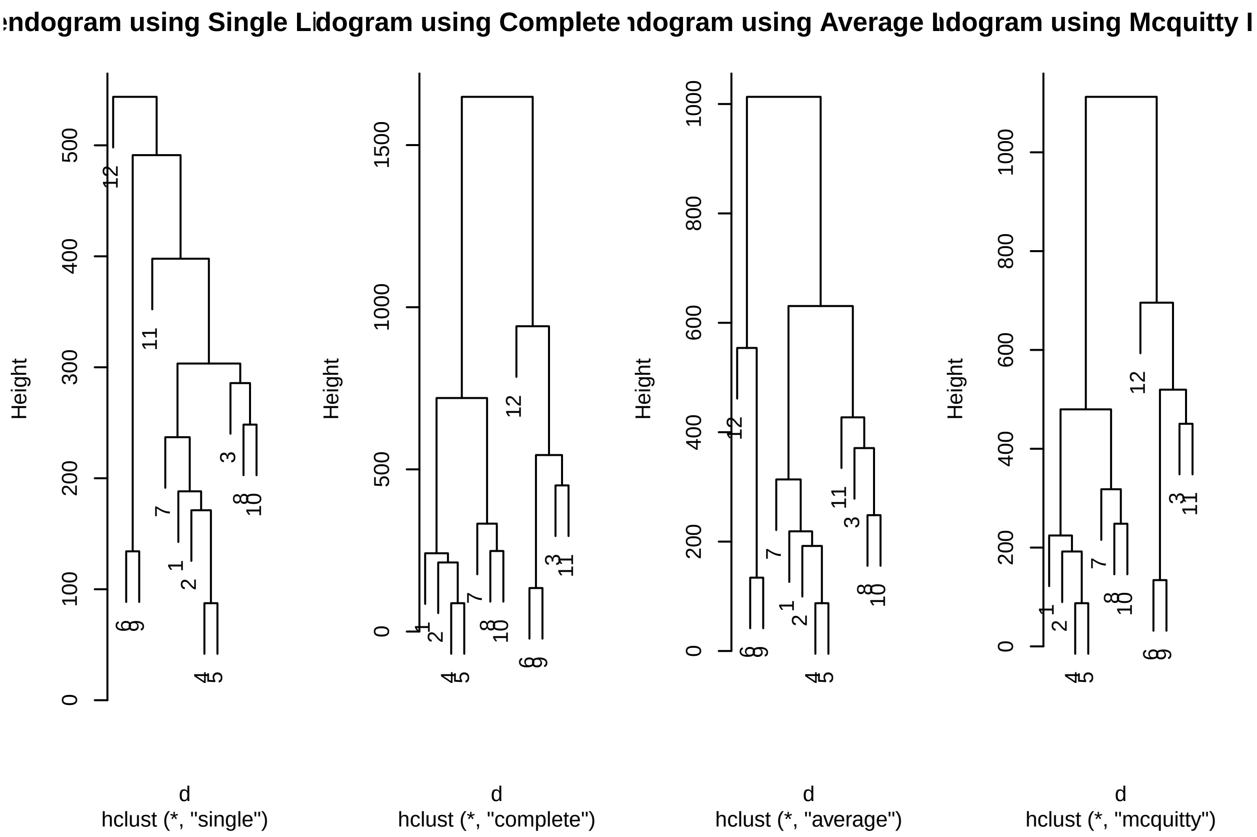
```
df[, c("names", "cluster")]
```

```
##                         names cluster
## 1     Pratyusha Mukhyopadhyay       1
## 2                Gourav Daga       2
## 3              Spandan Ghosh       2
## 4                Anuroop Roy       3
## 5            Shamie Dasgupta       1
## 6                Aishani Dey       3
## 7               Sruba Sarkar       3
## 8      Shubhradeep Chatterjee       3
## 9              Kankana Ghosh       2
## 10               Arnab Dutta       2
## 11          Kanchan Chowdhury       2
## 12         Purnaloke Sengupta       2
## 13          Soumyadeep Poddar       2
## 14              Priyankar Dey       2
## 15               Arunima Pal       2
## 16                 Sayan Das       3
## 17              Srijan Kundu       2
## 18               Trideep Roy       2
## 19            Debanjan Dutta       3
## 20             Adarsh Dalmia       2
## 21 Satyaki Basu Sarbadhikary       2
## 22         Sukanya Mukherjee       2
## 23             Protim Mondal       3
## 24        Dibyangana Debnath       2
## 25                Ashika Deb       2
## 26              Neha Agarwal       1
## 27             Anoushka Saha       2
## 28           Mehuli Bhandari       3
```

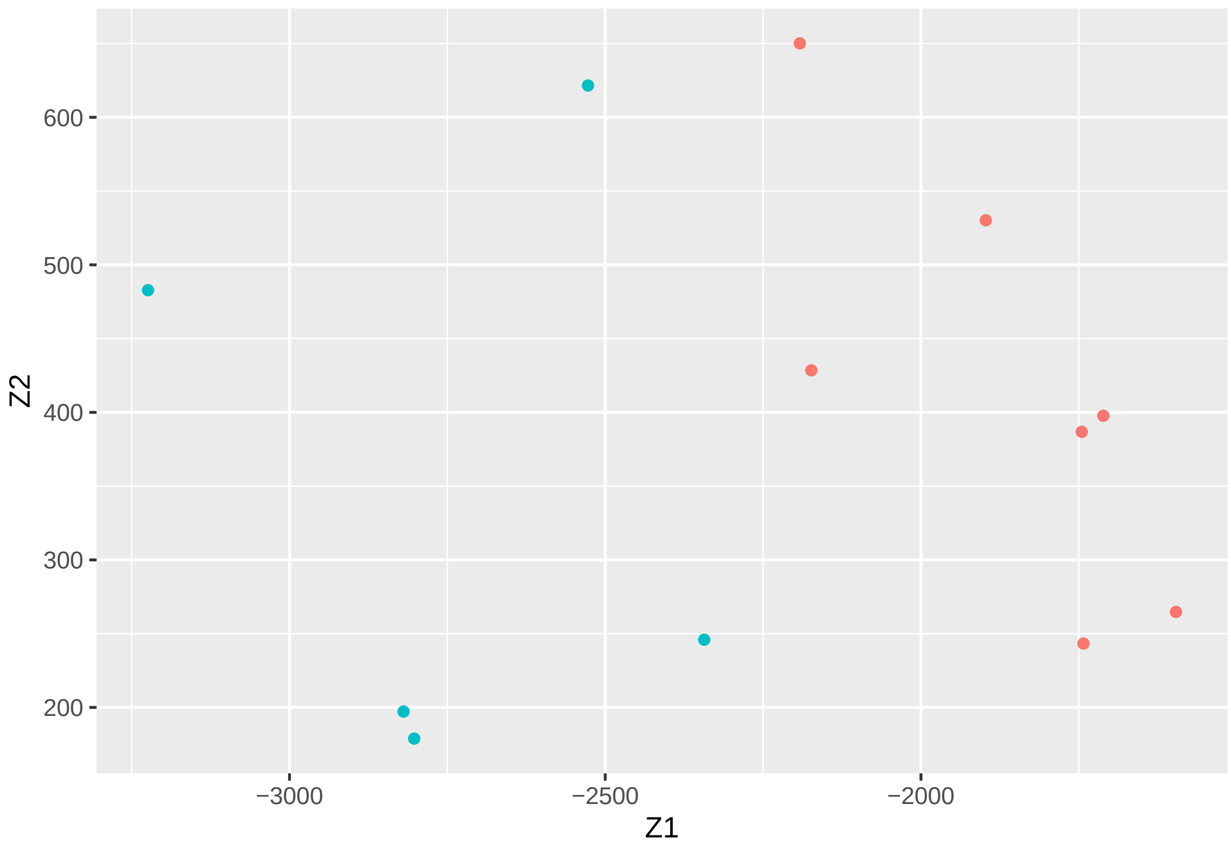
```
setwd("G:/My Drive/Semester 2/Paper 2/Practical")
data=read.csv("food_data.csv");data
```

```
##      X bread vegetables fruits meat poultry milk wine
## 1  MA2   332        428    354 1437     526  247  427
## 2  EM2   293        559    388 1527     567  239  258
## 3  CA2   372        767    562 1948     927  235  433
## 4  MA3   406        563    341 1507     544  324  407
## 5  EM3   386        608    396 1501     558  319  363
## 6  CA3   438        843    689 2345    1148  243  341
## 7  MA4   534        660    367 1620     638  414  407
## 8  EM4   460        699    484 1856     762  400  416
## 9  CA4   385        789    621 2366    1149  304  282
## 10 MA5   655        776    423 1848     759  495  486
## 11 EM5   584        995    548 2056     893  518  319
## 12 CA5   515       1097    887 2630    1167  561  284
```

```
par(mfrow=c(1,4))
d=proxy::dist(data[,-1])
hc1=hclust(d,method = "single")
plot(hc1,main="Dendogram using Single Linkage")
hc2=hclust(d,method = "complete")
plot(hc2,main="Dendogram using Complete Linkage")
hc3=hclust(d,method = "average")
plot(hc3,main="Dendogram using Average Linkage")
hc4=hclust(d,method = "mcquitty")
plot(hc4,main="Dendogram using Mcquitty Linkage")
```



```
U=eigen(cov(data[,-1]))$vectors
data[,c("Z1","Z2")]=as.matrix(data[-1])%*%U[,1:2]
data[,"cluster"]=cutree(hc2, k = 2)
data %>%
  ggplot(aes(x=Z1,y=Z2))+
  geom_point(aes(color=as.factor(cluster)))+
  guides(color="none")
```



```
k <- 3
km <- kmeans(data[,c(-1,-9,-10,-11)], k)
km$centers
```

```
##    bread vegetables   fruits   meat  poultry     milk     wine
## 1 390.20   563.6000 369.2000 1518.4  566.600 308.6000 372.4000
## 2 446.00   909.6667 732.3333 2447.0 1154.667 369.3333 302.3333
## 3 517.75   809.2500 504.2500 1927.0  835.250 412.0000 413.5000
```

```
km$cluster
```

```
##  [1] 1 1 3 1 1 2 1 3 2 3 3 2
```