ISLR | Chapter 10 Exercises

Marshall McQuillen

2/8/2019

Conceptual

1

NEED TO COME BACK TOO

- **A**. 10.12, illustrated below, is showing the the *within-cluster variation* is equal to twice the squared distance between each data point in cluster k (C_k) and that cluster's centroid, summed across all data points.

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{P}(x_{i,j}-x_{i',j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{P}(x_{i,j}-\bar{x}_{k,j})^2 \tag{1}$$

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{P}(x_{i,j}-x_{i',j})(x_{i,j}-x_{i',j}) = 2\sum_{i\in C_k}\sum_{j=1}^{P}(x_{i,j}-\bar{x}_{k,j})(x_{i,j}-\bar{x}_{k,j}) \tag{2}$$

$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{P}(x_{i,j}^2-2x_{i,j}x_{i',j}+x_{i',j}^2) = 2\sum_{i\in C_k}\sum_{j=1}^{P}(x_{i,j}^2-2\bar{x}_{k,j}x_{i,j}+\bar{x}_{k,j}^2) \tag{3}$$

$$\frac{|C_k|}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{P}(x_{i,j}^2-2x_{i,j}x_{i',j}+x_{i',j}^2) = 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}(x_{i,j}^2-2\bar{x}_{k,j}x_{i,j}+\bar{x}_{k,j}^2) \tag{4}$$

$$\sum_{i,i'\in C_k}\sum_{j=1}^{P}(x_{i,j}^2-2x_{i,j}x_{i',j}+x_{i',j}^2) = 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}(x_{i,j}^2-2\bar{x}_{k,j}x_{i,j}+\bar{x}_{k,j}^2) \tag{5}$$

$$\sum_{i\in C_k}\sum_{j=1}^{P}x_{i,j}^2 - 2\sum_{i,i'\in C_k}\sum_{j=1}^{P}x_{i,j}x_{i',j} + \sum_{i'\in C_k}\sum_{j=1}^{P}x_{i',j}^2 = 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}x_{i,j}^2 - 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}2\bar{x}_{k,j}x_{i,j} + 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}\bar{x}_{k,j}^2 \tag{6}$$

$$\sum_{i\in C_k}\sum_{j=1}^{P}x_{i,j}^2 - 2\sum_{i,i'\in C_k}\sum_{j=1}^{P}x_{i,j}x_{i',j} + \sum_{i'\in C_k}\sum_{j=1}^{P}x_{i',j}^2 = 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}x_{i,j}^2 - 4|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}\bar{x}_{k,j}x_{i,j} + 2|C_k|\sum_{i\in C_k}\sum_{j=1}^{P}\bar{x}_{k,j}^2 \tag{7}$$

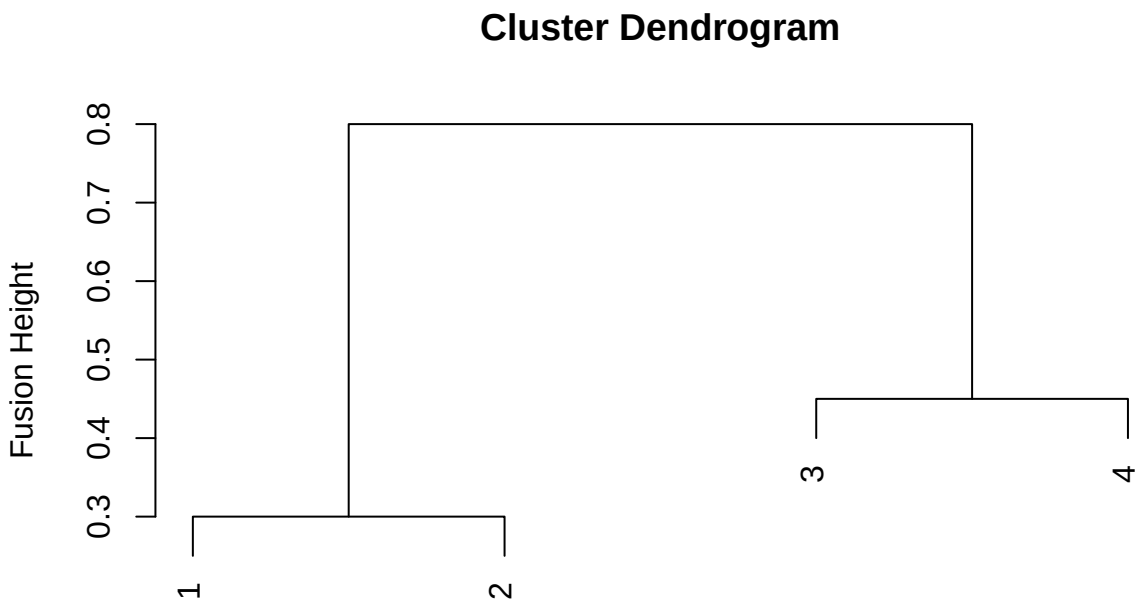
(8)

2

- **A**.

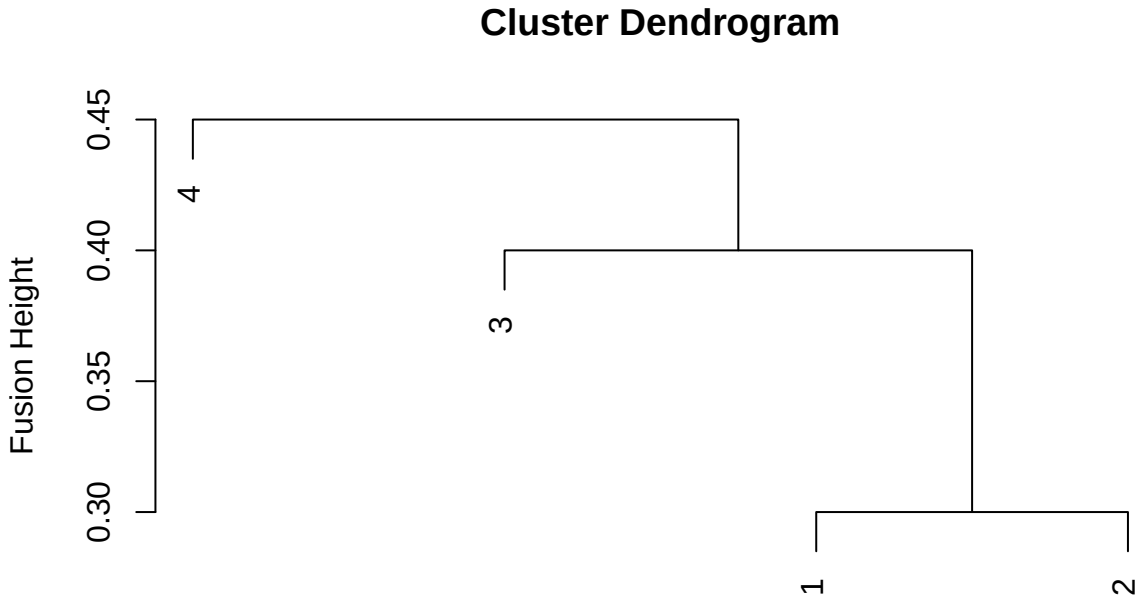
```
x <- matrix(c(0, 0.3, 0.4, 0.7,
              0.3, 0, 0.5, 0.8,
              0.4, 0.5, 0, 0.45,
              0.7, 0.8, 0.45, 0), ncol = 4, nrow = 4)
```

```
plot(hclust(as.dist(x), method = 'complete'),
     xlab = "",
     sub = "",
     ylab = "Fusion Height")
```



- B.

```
plot(hclust(as.dist(x), method = 'single'),
     xlab = "",
     sub = "",
     ylab = "Fusion Height")
```

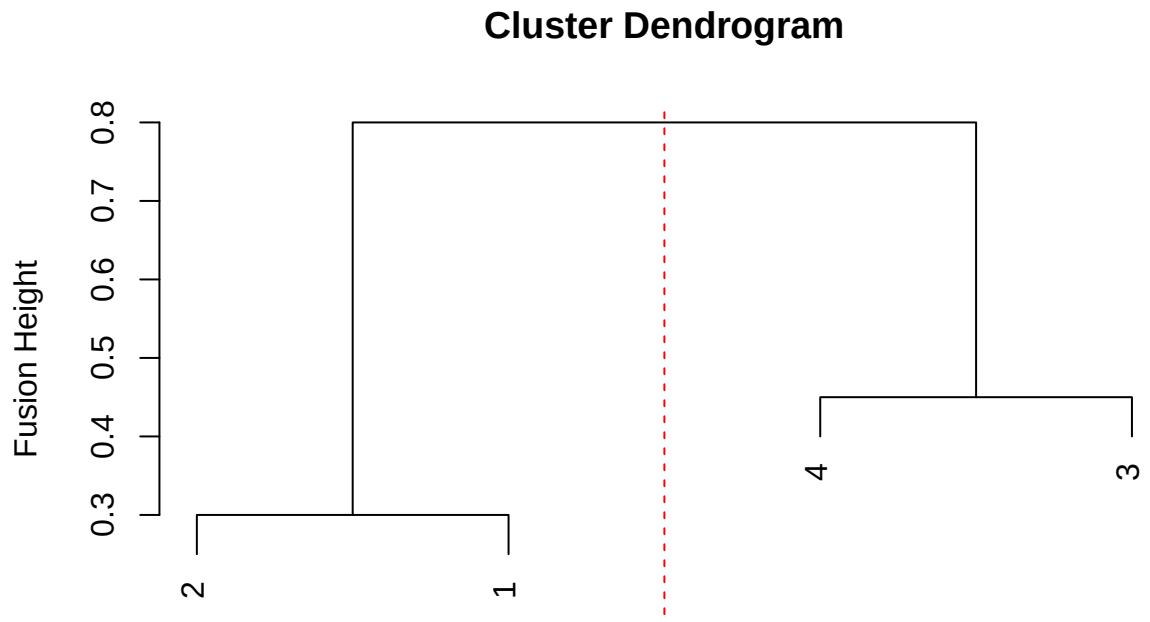


- **C**. Observations 1 & 2 will be in cluster A and observations 3 & 4 will be in cluster B (assuming one cuts the dendrogram at a height greater than 0.45).

- **D**. Although the answer to this question depends on where one cuts the dendrogram, the most likely clusters would contain observations 1 & 2 in cluster A and observations 3 & 4 in cluster B. This would results from a cut at a height greater than 0.3 and less than 0.4, which is the largest vertical distance on the dendrogram. If one were to make a cut between 0.4 and 0.45, then cluster A would contain observations 1, 2 & 3, while cluster B would consist of only observation 4. However, with the distance being greater between cluster's for the first grouping, that would be the more probable grouping.

- **E**. As shown below, one can simply switch the labels of the observations within each cluster to change the dendrogram without changing the meaning of the dendrogram. In addition, one could take the mirror image of the plot displayed along the dotted red line, producing a "new" dendrogram that has the same meaning.

```
plot(hclust(as.dist(x), method = 'complete'),
     xlab = "",
     sub = "",
     ylab = "Fusion Height", labels = c(2,1,4,3))
abline(v = 2.5, col = 'red', lty = 2)
```

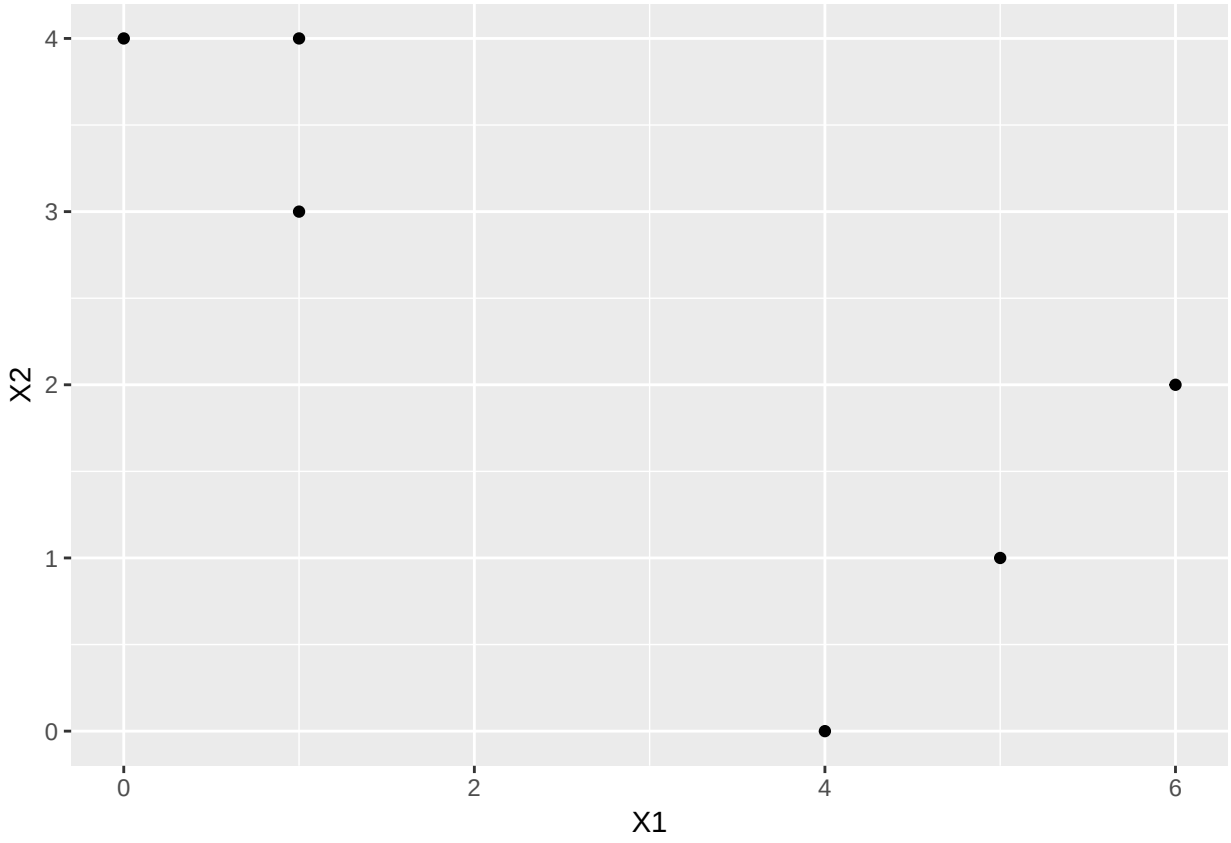


3

- **A**.

```
suppressPackageStartupMessages(library(ggplot2))
df <- data.frame(x1 = c(1,1,0,5,6,4),
```

```
                  x2 = c(4,3,4,1,2,0))
qplot(df$x1, df$x2, xlab = 'X1', ylab = 'X2')
```



- **B**.

```
suppressPackageStartupMessages(library(knitr))
df$group <- sample(c('A','B'), 6, replace = TRUE)
knitr::kable(df, caption = 'Sample Data with Group Assignments')
```

Table 1: Sample Data with Group Assignments

x1	x2	group
1	4	B
1	3	B
0	4	A
5	1	A
6	2	B
4	0	A

- **C**.

```r
group.a <- subset(df, group == 'A')
group.b <- subset(df, group == 'B')

# centroid calculation
calculate.centroid <- function(df, variables=c('x1','x2')) {

    coordinates <- NULL

    for (variable in variables) {
        column.mean <- mean(df[, variable])
        coordinates <- c(column.mean, coordinates)
    }

    return(coordinates)
}
group.a.centroid <- calculate.centroid(group.a)
group.b.centroid <- calculate.centroid(group.b)
print(paste("Group A centroid at coordinates",
            round(group.a.centroid[1], 2), 'and', round(group.a.centroid[2], 2)))
```

```
## [1] "Group A centroid at coordinates 1.67 and 3"
```

```r
print(paste("Group B centroid at coordinates",
            round(group.b.centroid[1], 2), 'and', round(group.b.centroid[2], 2)))
```

```
## [1] "Group B centroid at coordinates 3 and 2.67"
```

- **D**.

```r
centroid.matrix <- matrix(c(group.a.centroid, group.b.centroid),
                          nrow = 2,
                          ncol = 2)

reassign.cluster <- function(df,
                             col.idx=c(1,2),
                             cluster.column='group',
                             cluster.labels=c('A','B'),
                             centroids=centroid.matrix) {

    df.matrix <- as.matrix(df[, col.idx])

    updated.labels <- NULL
    for (i in 1:dim(df.matrix)[1]) {

        sqr.manhattan.dist <- (df.matrix[i, ] - centroids)^2
        euclidean.dist <- sqrt(colSums(sqr.manhattan.dist))
        closest.centroid <- which.min(euclidean.dist)
        updated.labels <- c(updated.labels, cluster.labels[closest.centroid])

    }

    return(data.frame(df.matrix, group = updated.labels))
```

```
}
updated.df <- reassign.cluster(df)
knitr::kable(updated.df, caption = "Sample Data After One Reassignment Iteration")
```

Table 2: Sample Data After One Reassignment Iteration

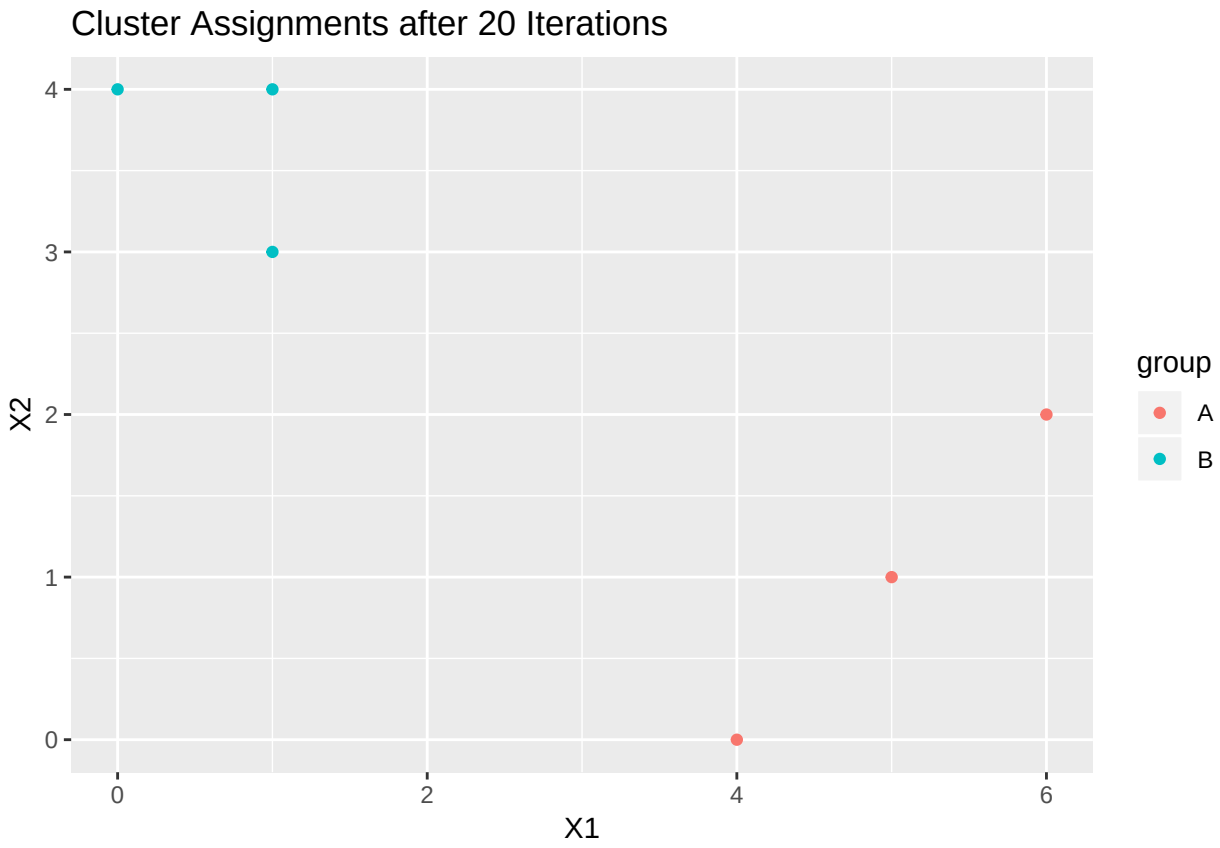
x1	x2	group
1	4	A
1	3	A
0	4	A
5	1	B
6	2	B
4	0	B

- **E**.

```
for (i in 1:20) {
    # split data
    group.a <- subset(df, group == 'A')
    group.b <- subset(df, group == 'B')

    # recalculate centroid
    group.a.centroid <- calculate.centroid(group.a)
    group.b.centroid <- calculate.centroid(group.b)

    # create centroid matrix
    centroid.matrix <- matrix(c(group.a.centroid, group.b.centroid), nrow = 2, ncol = 2)

    # re-assign group labels
    df <- reassign.cluster(df)
}
```

- **F**.

```
ggplot(df, aes(x = x1, y = x2, color = group)) +
    geom_point() +
    xlab('X1') +
    ylab('X2') +
    ggtitle('Cluster Assignments after 20 Iterations')
```



4

- **A**. *Assuming the maximal intercluster dissimilarity is* **not equal** *to the minimal intercluster dissimilarity*, the clusters {1, 2, 3} and {4, 5} will be fused higher on the dendrogram when complete linkage is used. While complete linkage uses the *maximal* intercluster dissimilarity, single linkage uses the *minimal* intercluster dissimilarity. So, if the first assumption is held, using complete linkage will lead to the fusion occurring higher on the dendrogram. However, if the maximal intercluster dissimilarity is equal to the minimal intercluster dissimilarity, then they would fuse at the same height, regardless of the linkage method used.
- **B**. Assuming that in both scenarios the clusters {5} and {6} are by themselves (i.e. not joined with any other clusters/observations), then the clusters would be joined at the same height regardless of whether single or complete linkage were to be used. As stated in part **A**, complete linkage uses the maximal intercluster dissimilarity while single linkage uses the minimal intercluster dissimilarity. In this scenario, both "clusters" only have one observation in them so the maximal and minimal intercluster dissimilarity will be the exact same.

5

The three different scaling options, and the results that one would expect by running the K-means algorithm with K = 2 are:

- *Variables unscaled* - Since the number of socks purchased will be greater than the number of computers purchased and will drive the dissimilarity between the clusters, one would expect the two clusters to be determined by the number of pairs of socks an indiviual shopper purchases. Perhaps those shoppers that purchased seven or fewer pairs of socks would be in one cluster and those who purchased more would be in the other cluser, or something similar.
- *Variables scaled to have standard deviation one* - When the number of items purchased is scaled by that item's standard deviation across the data set, then each variable is given (roughly) equal importance in the clustering algorithm, therefore one would expect clusters with those that bought computers and those that did not buy computers (the *quantity* of socks purchased would have very little affect on the clusters, since the computer variable could conceptually be grouped into the "haves" and the "have not's").
- *Variables scaled by dollars spent* - If the variables were measured in dollars spent (as opposed to quantity purchased), then one would expect the clusters to be dominated by whether a particular shopper purchased a computer or not.

6

- **A**. As shown in the equation below, the total variance in a data set is defined as the summation across all columns of the summation of the squared elements in attribute j divided by N (assuming each column has been centered to have mean 0).

$$\sum_{j=1}^{P} Var(\mathbf{X}) = \sum_{j=1}^{P} \frac{1}{N} \sum_{i=1}^{N} x_{i,j}^2 = Total\ Variance\ in\ \mathbf{X}$$

  After the data set has been "mapped" onto the principal components, saying the m^{th} principal component explaines 10% of the variance in the data set is saying that the summation of the squared elements *of the principal component* divided by N is equal to 10% of the original variation (given in the equation above) in the data set.

$$\frac{\sum_{i=1}^{N} z_{i,m}^2}{\sum_{j=1}^{P} \frac{1}{N} \sum_{i=1}^{N} x_{i,j}^2} = 0.1 \quad where \quad z_{i,m}^2 = \left(\sum_{j=1}^{P} \phi_{j,m} x_{i,j} \right)^2$$

  where $\phi_{j,m}$ = the loading associated with the j^{th} column for the m^{th} principal component.

Resources

Dendrograms in R