Clustering

The Issue:

In this lab, we are given the USArrests dataset, which includes information of number of arrests per 100,000 residents for each of three crimes (: Assault, Murder, and Rape) in the United States. Our task is to perform principal component analysis (PCA) to reduce the dimensionality of the data and interpret the principal components. We will also use k-means clustering and hierarchical clustering to identify patterns and groupings in the data.

The USArrests dataset has five attributes for each state: State name, Murder rate, Assault rate, UrbanPop (which measures the percentage of the population living in urban areas), and Rape rate.

First, we will perform PCA to identify the most important patterns in the data and reduce the number of variables.

Next, we will use k-means clustering to group the states into clusters based on their crime rates. We will need to choose an appropriate number of clusters (k) that balances the benefits of having more clusters.

Finally, we will perform hierarchical clustering to identify clusters in a hierarchical structure, using a dendrogram. This will allow us to explore different levels of granularity in the clustering solution.

Findings:

From PCA we interpret that, first loading vector demonstrates a much correlation between major crimes and urbanization level. Whereas the second loading vector demonstrates a less correlation between these crimes and urbanization level. Thus, murder, assault, and rape occur together in states, although there is little correlation between these crimes and urban population.

Performing k-means clustering, we find that as “k” value is increasing the “tot.withinss” decreasing, indicating that algorithm has successfully identified clusters that are homogenous and well-separated from each other.

Performing hierarchical clustering resulting in attractive tree-based observation called dendrograms. Average and complete linkage comparatively yield more balanced dendrograms.

Discussion:

From PCA, the first main component is responsible for 62.0% of the variation in the data, with the second principal component accounting for 24.7%, the third for 8.91%, and the fourth for 4.34%.

From k-means clustering, with k=2, we get “tot.withinss” value as 128.6, indicating that the data points within each cluster are not tightly packed around their respective centroids, and the clusters may be overlapping or poorly separated from each other. With k=3, we get “tot.withinss” value as 97.97, indicating that the total variation within each cluster is relatively low. If we increase k-value, “tot.withinss” value decreases, resulting in tightly packed cluster.

From hierarchical clustering, code yielded balanced dendrograms from complete and average linkage but yielded a poor dendrogram from single linkage.

Appendix A: Methods

PCA is a technique used to reduce the dimensionality of a dataset by identifying patterns in the data. To perform PCA, we first calculate the mean and variance of the variables in the dataset. If the variables have different values for mean and variance, we use the prcomp() function in R to perform PCA. This gives us the center, scale, rotation, sdev, and x values. Center and scale corresponding to the mean and standard deviation of the variables. We can then plot the first two principal components using the biplot() function. We can also calculate the variance and proportion of variance explained by each principal component. Plot the proportion of variance explained by each component using the plot() function.

To perform k-means clustering, we start with a dataset in matrix format that is split into two equal halves. We use the kmeans() function in R to cluster the data. Then plot the observations, coloring each according to its cluster assignment. We repeat this process using k=3, and observe the tot.withinss value (which measures the total within-cluster sum of squares). Based on our observations, we draw conclusions about the effectiveness of the clustering algorithm.

To perform hierarchical clustering we use hclust() function. We can perform average, single, and complete linkage. We then use the plot() function to generate dendrograms. Dendrogram are tree-based diagrams that visualize the clustering results. We apply this process to a three-dimensional dataset using correlation-based distance. It measures similarity between observations.

Appendix B: Results

Below result is a plot obtained from PCA

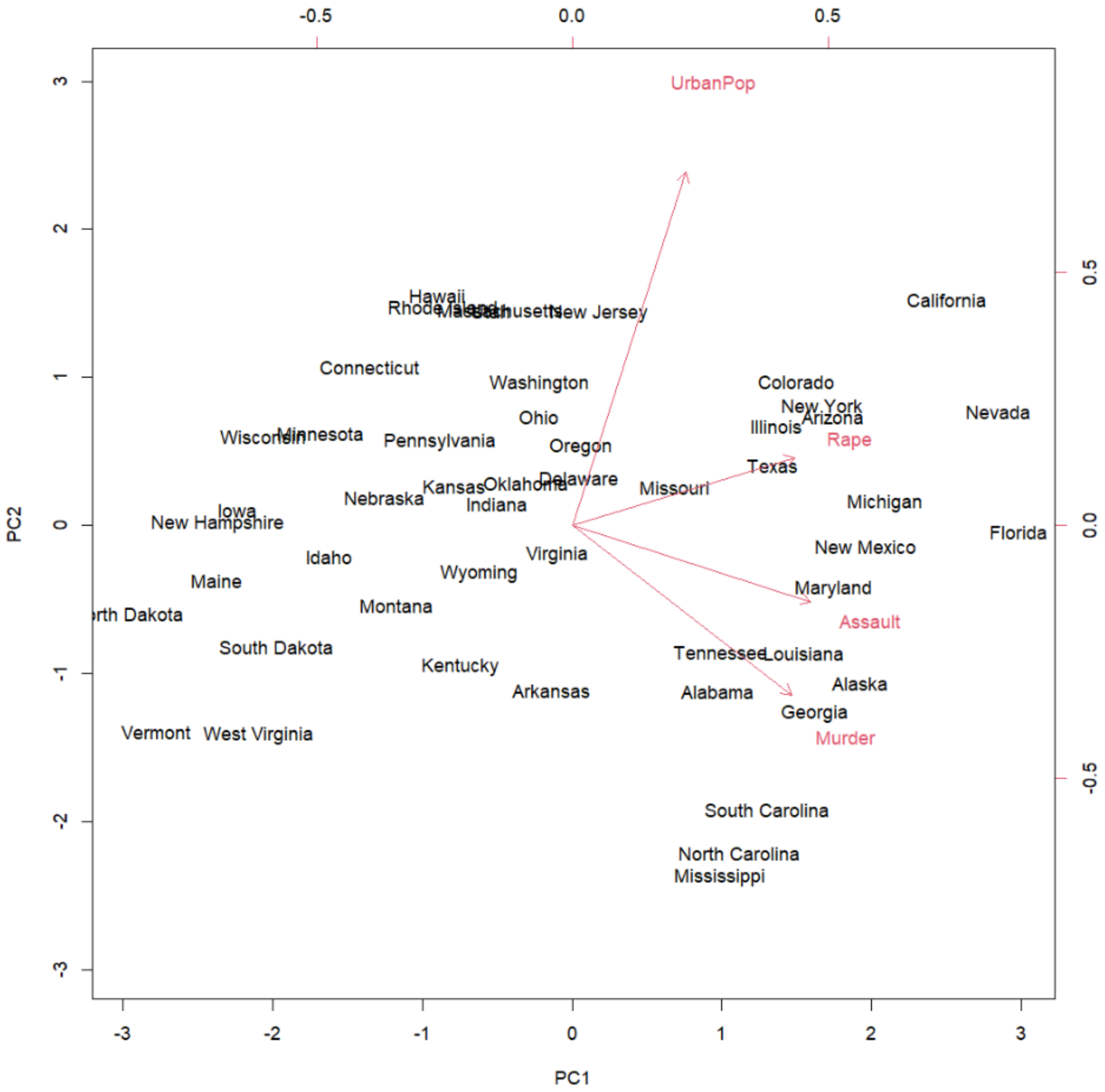
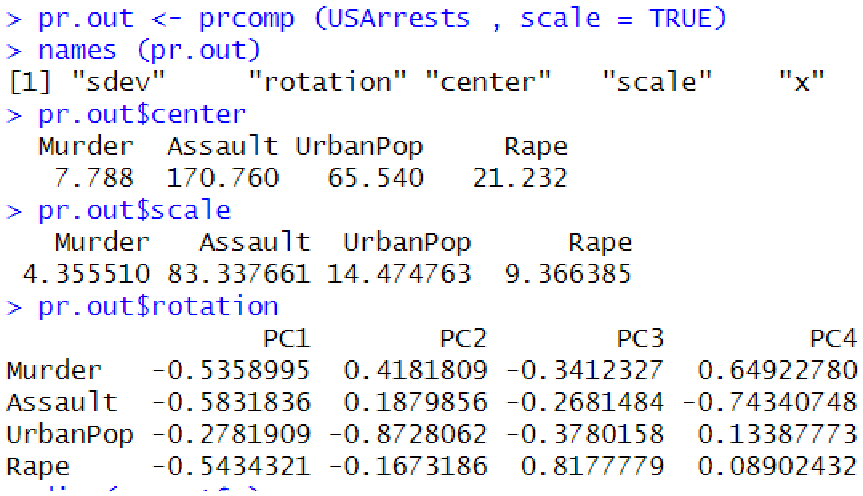


Figure 1. Biplot for first two principal components

Next we can see result of prcomp()



```
> pr.out <- prcomp (USArrests , scale = TRUE)
> names (pr.out)
[1] "sdev"     "rotation" "center"   "scale"    "x"
> pr.out$center
  Murder  Assault UrbanPop     Rape
   7.788  170.760   65.540   21.232
> pr.out$scale
   Murder   Assault  UrbanPop      Rape
 4.355510 83.337661 14.474763  9.366385
> pr.out$rotation
                PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

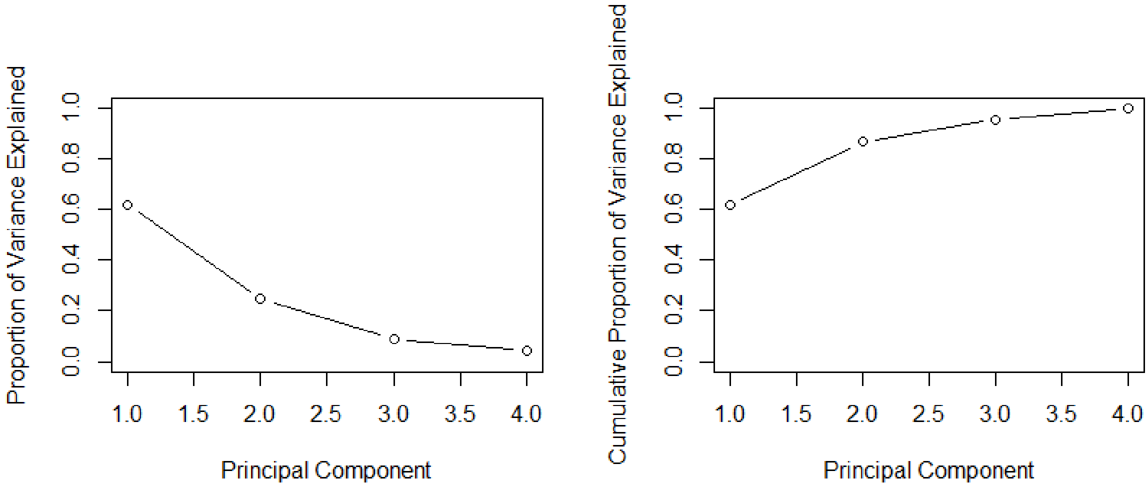


Figure 2. Variance plots

Below are the plots from k-means clustering for k=2, k=3.

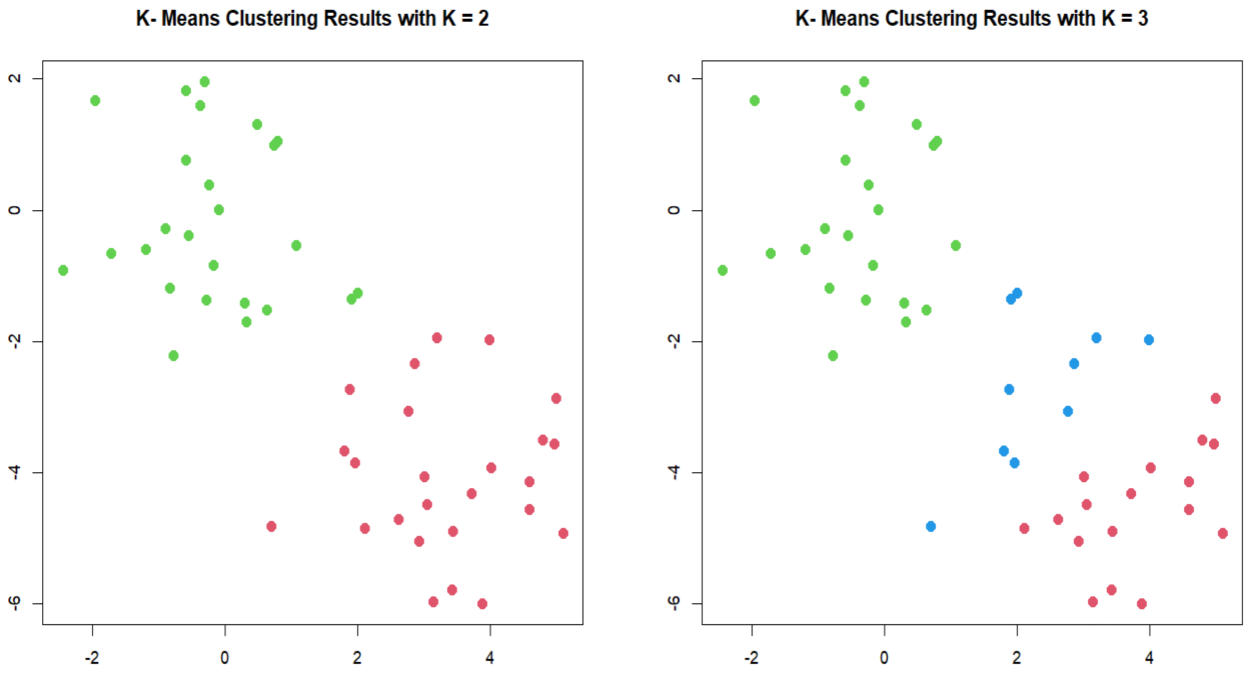
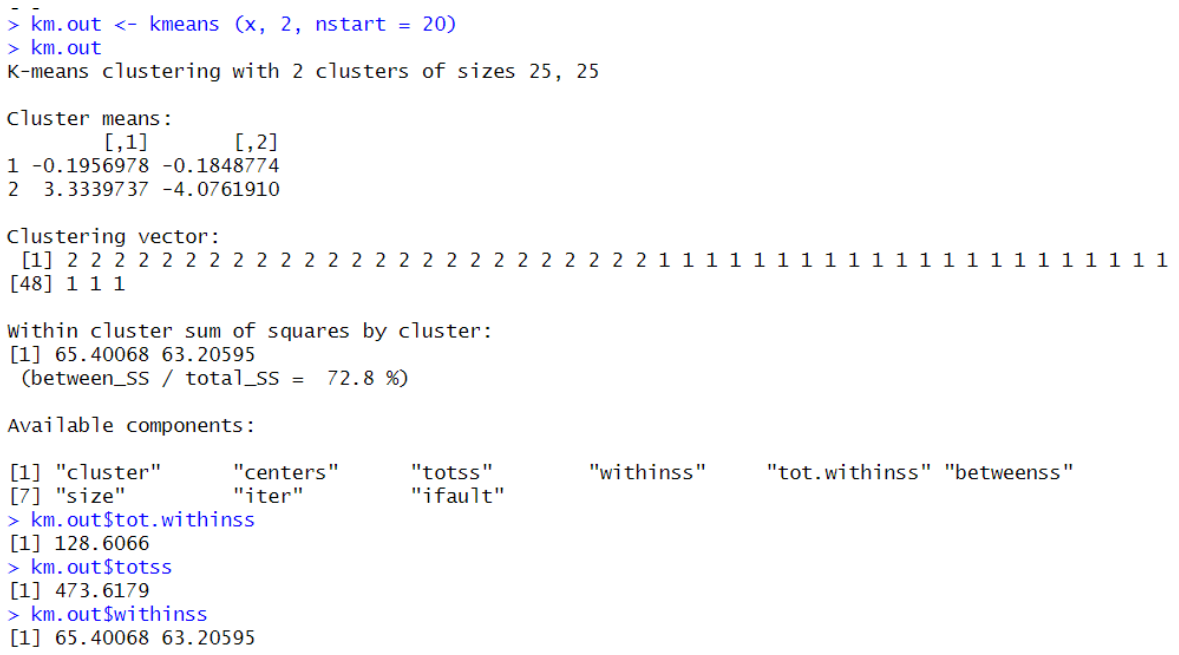
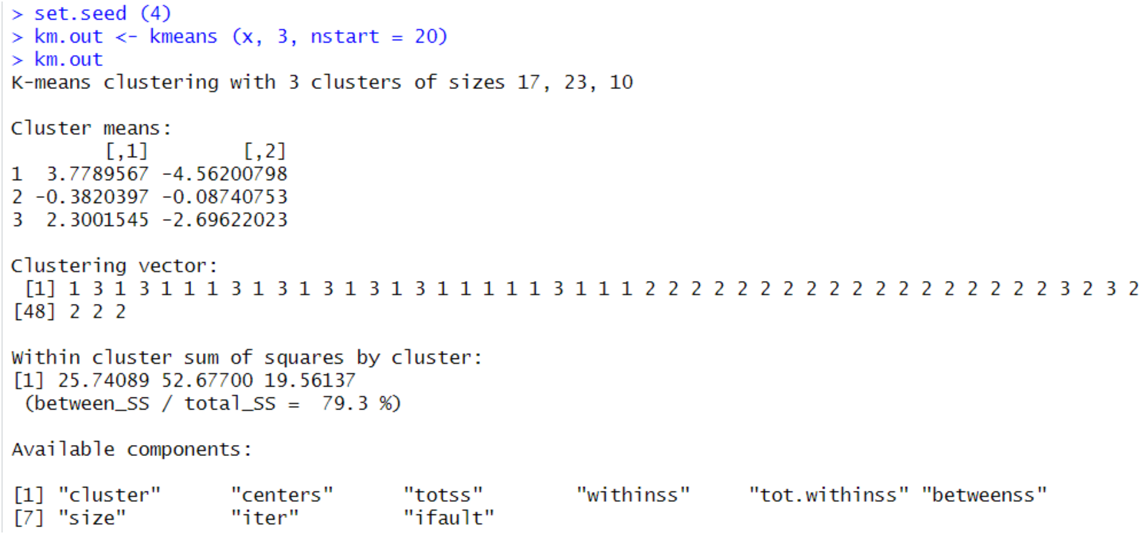


Figure 3. k-means clustering for different k values.

Next we see various results from k-means clustering, for k=2 & k=3.



```
> km.out <- kmeans (x, 2, nstart = 20)
> km.out
K-means clustering with 2 clusters of sizes 25, 25

Cluster means:
        [,1]       [,2]
1 -0.1956978 -0.1848774
2  3.3339737 -4.0761910

Clustering vector:
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[48] 1 1 1

Within cluster sum of squares by cluster:
[1] 65.40068 63.20595
 (between_SS / total_SS =  72.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> km.out$tot.withinss
[1] 128.6066
> km.out$totss
[1] 473.6179
> km.out$withinss
[1] 65.40068 63.20595
```



```
> set.seed (4)
> km.out <- kmeans (x, 3, nstart = 20)
> km.out
K-means clustering with 3 clusters of sizes 17, 23, 10

Cluster means:
        [,1]        [,2]
1  3.7789567 -4.56200798
2 -0.3820397 -0.08740753
3  2.3001545 -2.69622023

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2
[48] 2 2 2

Within cluster sum of squares by cluster:
[1] 25.74089 52.67700 19.56137
 (between_SS / total_SS =  79.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

Below we see the various results of hierarchical clustering,

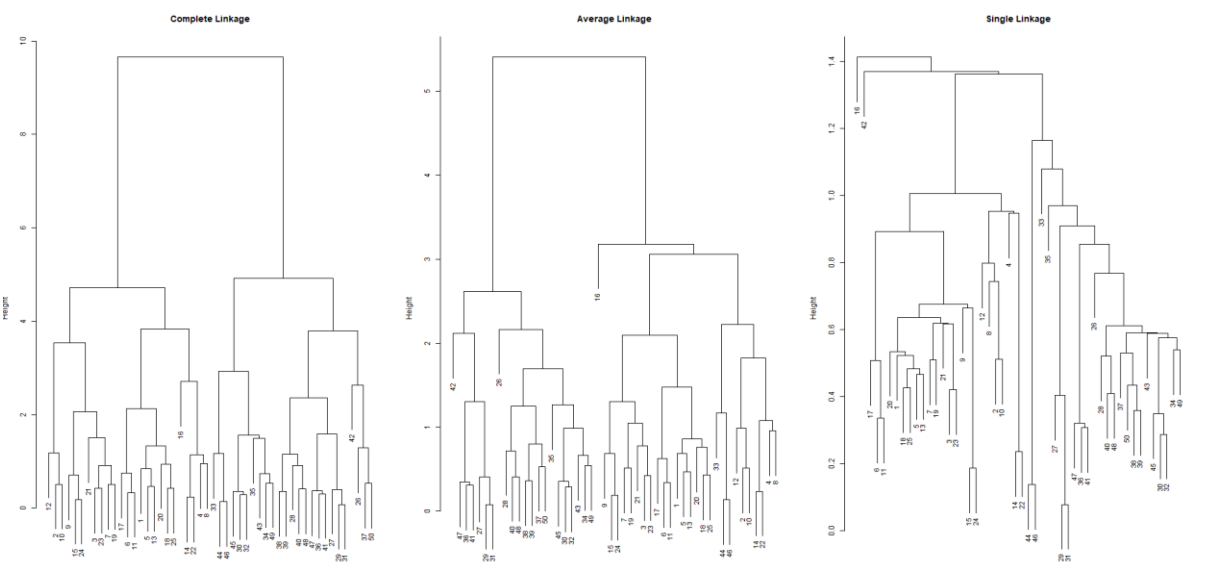


Figure 4. Hierarchical clustering performing on complete, average and single linkage

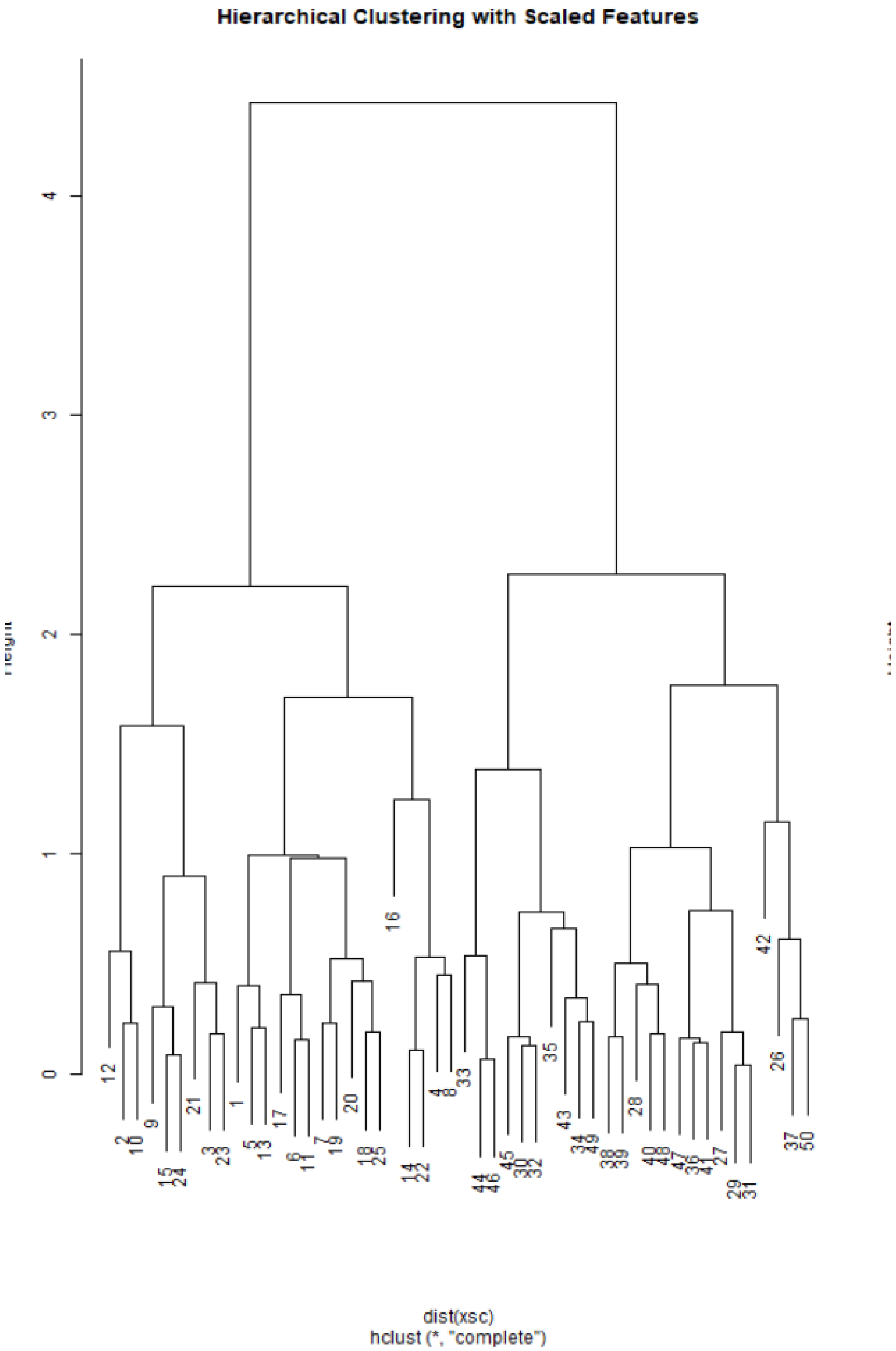


Figure 5. Hierarchical clustering with scaled features

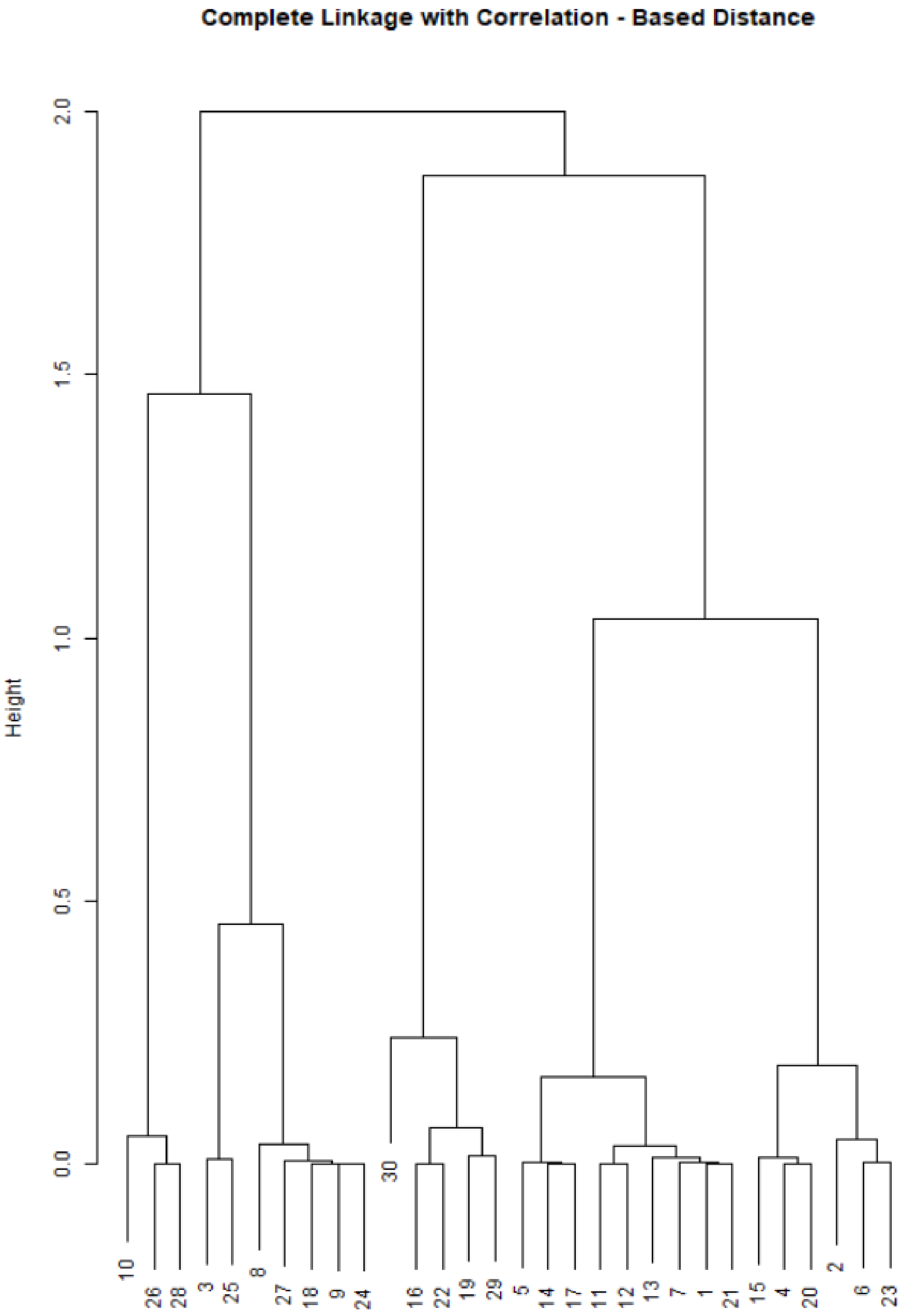


Figure 6. Complete linkage with correlation-based distance.

Appendix C: Code

We used R language and performed our code in R-Studio,

Code for PCA:

```
states <- row.names(USArrests)

states

names (USArrests)

apply (USArrests , 2, mean)

apply (USArrests , 2, var)

pr.out <- prcomp (USArrests , scale = TRUE)

names (pr.out)

pr.out$center

pr.out$scale

pr.out$rotation

dim (pr.out$x)

biplot (pr.out , scale = 0)


pr.out$rotation = -pr.out$rotation

pr.out$x = -pr.out$x

biplot (pr.out , scale = 0)

pr.out$sdev


#variance explained by each principal component

pr.var <- pr.out$sdev^2

pr.var


#proportion of variance

pve <- pr.var / sum (pr.var)

pve
```

```
par (mfrow = c(1, 2))
plot (pve , xlab = " Principal Component ",
        ylab = " Proportion of Variance Explained ", ylim = c(0, 1),
        type = "b")
plot ( cumsum (pve), xlab = " Principal Component ",
         ylab = " Cumulative Proportion of Variance Explained ",
         ylim = c(0, 1), type = "b")
```

Code for k-means clustering:

```
set.seed (2)
x <- matrix ( rnorm (50 * 2), ncol = 2)
x[1:25, 1] <- x[1:25, 1] + 3
x[1:25, 2] <- x[1:25, 2] - 4

km.out <- kmeans(x, 2, nstart = 20)
km.out
km.out$cluster
km.out$tot.withinss
km.out$totss
km.out$withinss

par (mfrow = c(1, 2))
plot (x, col = (km.out$cluster + 1),
      main = "K- Means Clustering Results with K = 2",
      xlab = "", ylab = "", pch = 20, cex = 2)

set.seed (4)
km.out <- kmeans (x, 3, nstart = 20)
km.out
```

```
plot (x, col = (km.out$cluster + 1),
      main = "K- Means Clustering Results with K = 3",
      xlab = "", ylab = "", pch = 20, cex = 2)
km.out$tot.withinss
km.out$withinss
```

<u>Code for hierarchical clustering</u>:

```
hc.complete <- hclust ( dist (x), method = "complete")
hc.average <- hclust ( dist (x), method = "average")
hc.single <- hclust ( dist (x), method = "single")

par (mfrow = c(1, 3))
#complete linkage
plot (hc.complete, main = " Complete Linkage ",
      xlab = "", sub = "", cex = .9)
#average linkage
plot (hc.average , main = " Average Linkage ",
      xlab = "", sub = "", cex = .9)
#single linkage
plot (hc.single, main = " Single Linkage ",
      xlab = "", sub = "", cex = .9)

cutree (hc.complete, 2)
cutree (hc.average , 2)
cutree (hc.single, 2)
cutree (hc.single, 4)

xsc <- scale (x)
plot ( hclust ( dist (xsc), method = "complete") ,
         main = " Hierarchical Clustering with Scaled Features ")
```

```
x <- matrix ( rnorm (30 * 3), ncol = 3)
dd <- as.dist (1 - cor (t(x)))
plot ( hclust (dd, method = "complete") ,
      main = " Complete Linkage with Correlation - Based Distance ",
      xlab = "", sub = "")
```