39-TilakPoudel-ClassPresentation

Tilak Poudel

2025-05-10

The slides must include a) code used to generate 500 random data with five variables b) code used to fix the random seed with roll number c) code used to fit the assigned supervised and unsupervised statistical models d) codes used to get model accuracy indices and e) interpretations of the outputs from these codes

Assigned models:

1. Roll 1 - 10: Fitting Logistic regression and KNN classification models and comparison of the model assumptions and accuracy indices to select the best model for the data based on training and testing datasets
2. Roll 11 - 20: Fitting Decision Tree and Random Forest models and comparison of the model assumptions and accuracy indices to select the best model for the data based on training and testing datasets
3. Roll 21 - 30: Fitting dimension (variables) reduction techniques i.e. PCA and MDS and selecting the best model for this data with careful interpretations of the bi-plots
4. Roll 31 - 40: Fitting dimension (cases) reduction techniques i.e. HCA and k-means and selecting the best model for this data with careful interpretations of the cluster plots

Step (a): Generate 500 random observations with 5 variables

```
set.seed(39)

# Generate the data
data <- data.frame(
  X1 = rnorm(500, mean = 10, sd = 2),
  X2 = runif(500, min = 0, max = 100),
  X3 = rnorm(500, mean = 50, sd = 10),
  X4 = runif(500, min = 2, max=8),
  X5 = sample(1:3, 500, replace = TRUE)
)

# View first few rows
head(data)
```

```
##          X1        X2       X3       X4 X5
## 1  9.628869  8.010958 44.85668 6.625935  3
## 2  7.541515 32.069833 46.53322 2.767781  3
## 3  9.145594 92.334041 44.12986 4.383337  3
## 4  8.808036 17.268322 42.05698 2.716910  1
```

```
## 5 10.934647 83.514670 54.58758 2.845628  1
## 6 10.843278 89.084333 43.61641 6.517983  2
```

```
print(dim(data))
```

```
## [1] 500   5
```

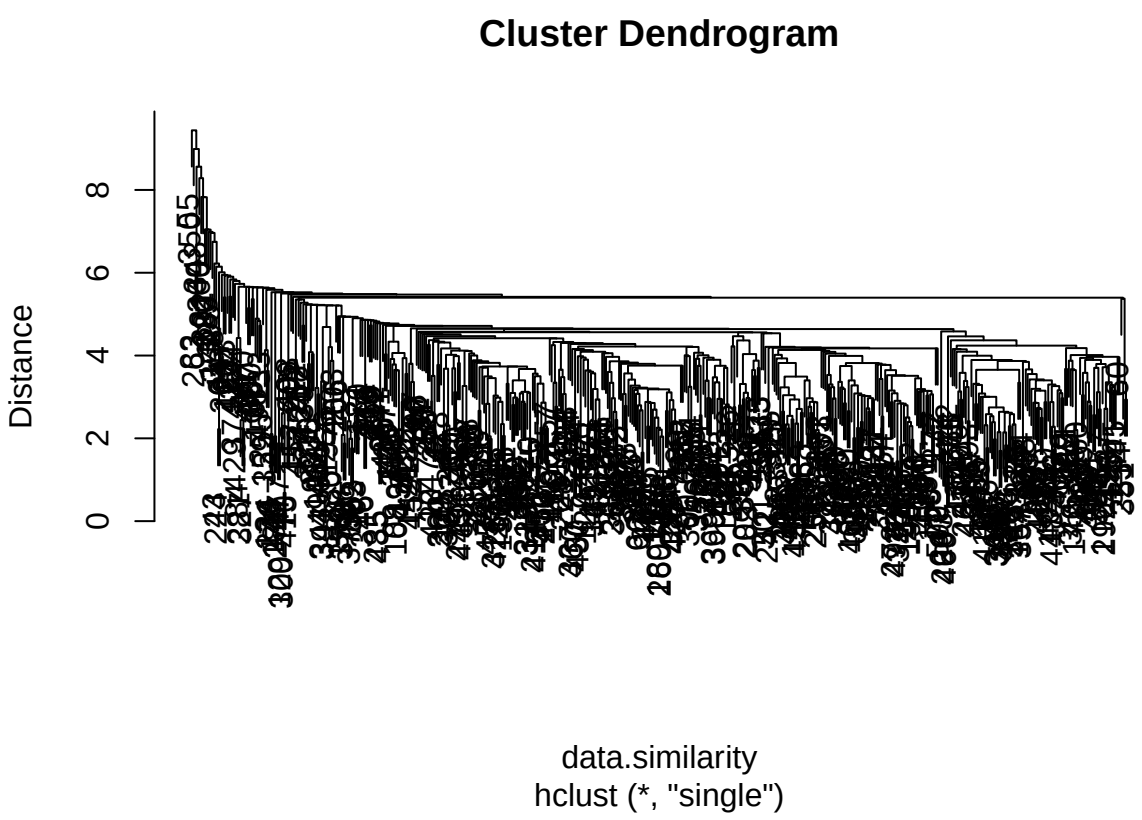
HCA (Hierarchical Cluster Analysis)

Hiererchical clustering with single linkage

```
# data <- scale(data[, 1:4])  # Use only independent variables

data.similarity <- dist(data)
hirar.1 <- hclust(data.similarity, method='single')
hirar.1
```

```
##
## Call:
## hclust(d = data.similarity, method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 500
```

```
# Plot
plot(
  hirar.1,
  labels=rownames(data),
  ylab="Distance"
)
```

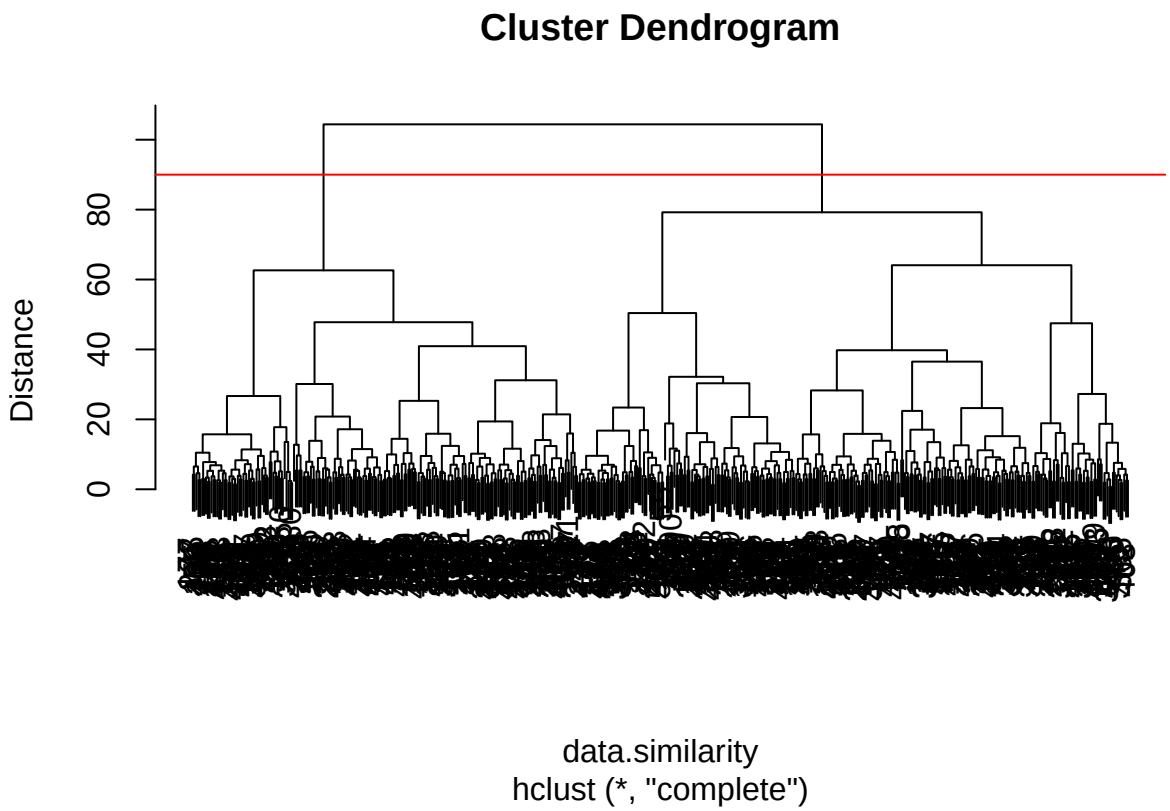


Hiererchical clustering with complete linkage

```
hirar.2 <- hclust(data.similarity, method='complete')
hirar.2
```

```
##
## Call:
## hclust(d = data.similarity, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 500
```

```
# Plot
plot(
  hirar.2,
  labels=rownames(data),
  ylab="Distance"
)

abline(h = 90, col = 'red')
```

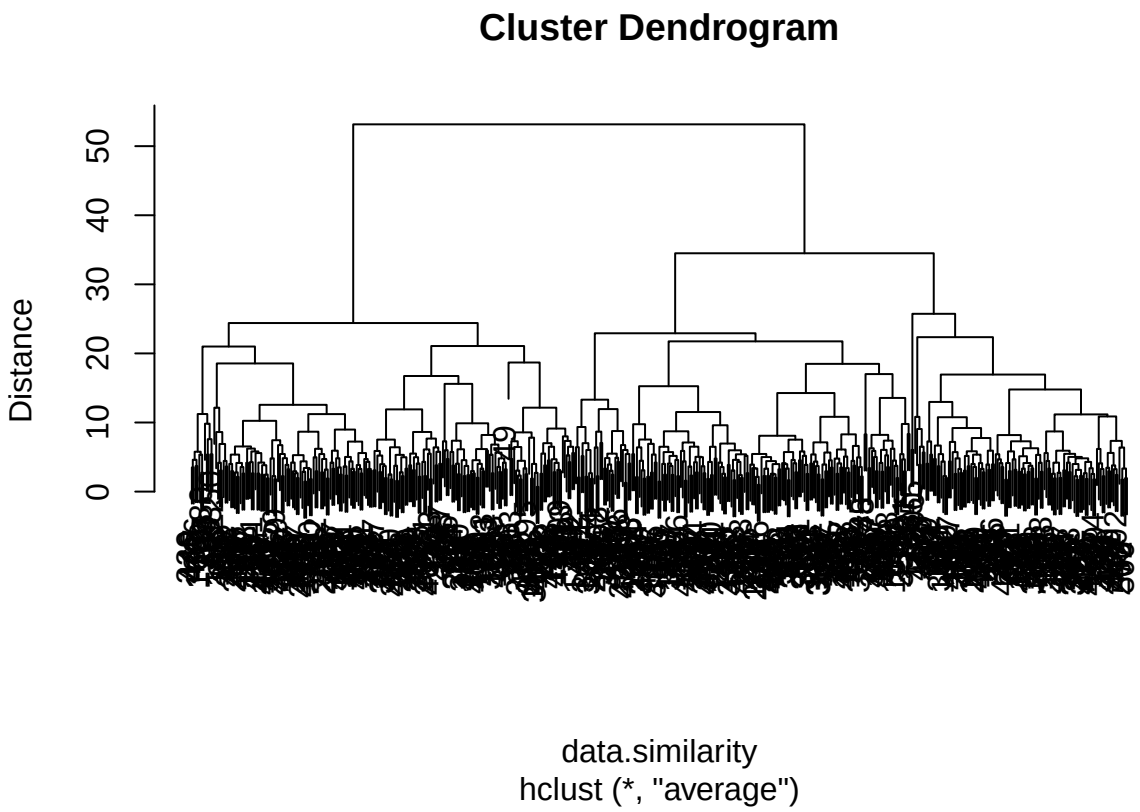


Hiererchical clustering with average linkage

```
hirar.3 <- hclust(data.similarity, method='average')
hirar.3
```

```
##
## Call:
## hclust(d = data.similarity, method = "average")
##
## Cluster method   : average
## Distance         : euclidean
## Number of objects: 500
```

```
# Plot
plot(
  hirar.3,
  labels=rownames(data),
  ylab="Distance"
)
```

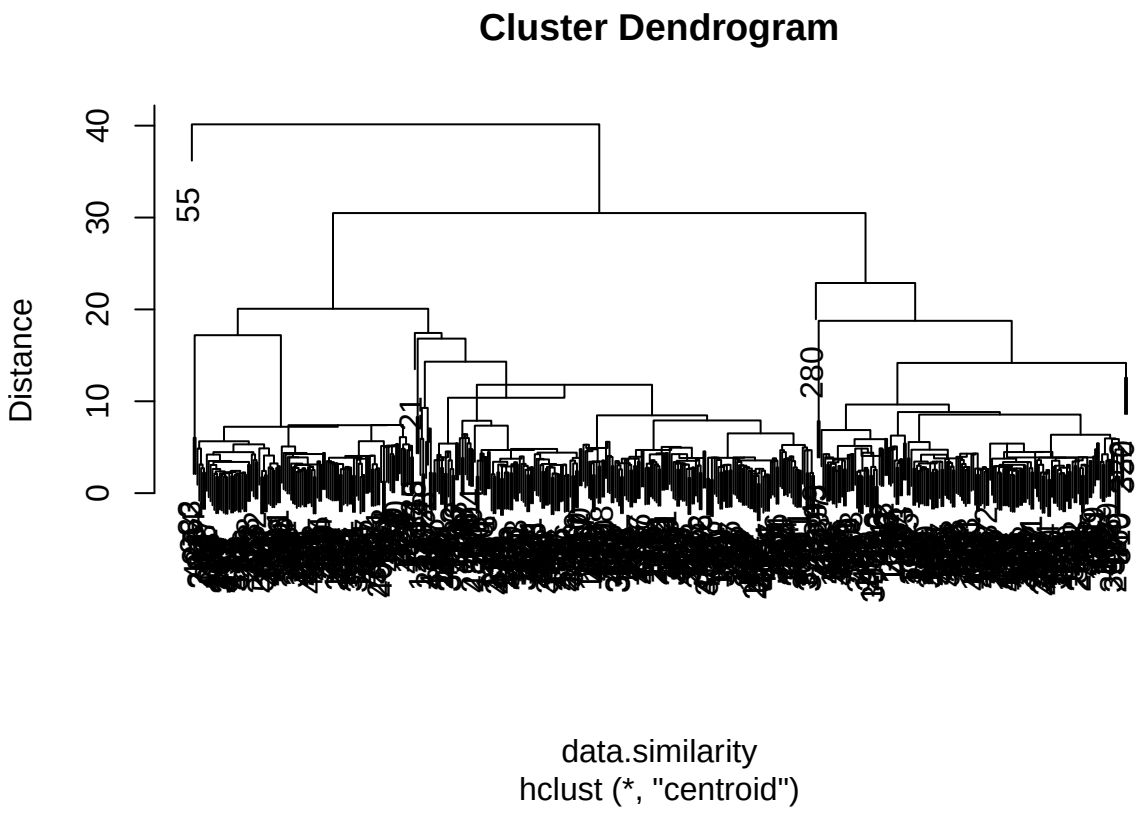


Hiererchical clustering with centroid linkage

```
hirar.4 <- hclust(data.similarity, method='centroid')
hirar.4
```

```
##
## Call:
## hclust(d = data.similarity, method = "centroid")
##
## Cluster method   : centroid
## Distance         : euclidean
## Number of objects: 500
```

```
# Plot
plot(
  hirar.4,
  labels=rownames(data),
  ylab="Distance"
)
```



Selecting the number of clusters (k)

Observing the dendogram with various linkage, we can take similarity complete and cut at distance 90 to get 2 clusters.

Fit the model: Clustering(K-mean)

```
# Fit the k-mean clustering model on the data
# Scale if needed (optional here as all variables have same scale)
# data <- scale(data)

set.seed(39)
kmeans_model <- kmeans(data, centers=2, nstart = 20)
kmeans_model
```

```
## K-means clustering with 2 clusters of sizes 248, 252
##
## Cluster means:
##          X1       X2       X3       X4       X5
## 1  9.858237 22.98047 50.68765 5.053990 1.991935
## 2 10.033920 73.60512 49.89743 5.166196 2.079365
##
```

```
## Clustering vector:
##   [1] 1 1 2 1 2 2 2 2 2 2 2 1 2 1 1 1 2 2 2 1 1 1 2 2 1 1 1 1 1 1 2 1 1 2 1 1 2
##  [38] 1 1 1 2 1 2 2 1 2 2 2 1 1 1 1 2 2 2 1 2 1 2 2 2 1 1 1 2 2 1 2 1 1 2 2 2 1
##  [75] 1 1 1 2 1 1 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 1 1 1 2 2 1 1 1 2 2 1 1 2 1 2 1
## [112] 2 1 1 2 2 2 1 2 1 1 1 2 2 1 2 1 1 1 2 2 2 2 1 2 2 1 1 1 1 1 2 1 2 1 1 1 1
## [149] 1 1 2 2 1 1 1 1 2 2 1 2 1 1 1 2 1 2 2 1 1 2 2 2 1 2 2 2 1 1 1 1 1 2 2 2 2
## [186] 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 1 2 2 2 2 1 1 1 2 2 1 1 2 2 2 1 2 1 2 1
## [223] 1 2 2 1 2 2 1 1 2 1 2 1 1 1 2 1 2 2 1 1 2 2 2 1 2 2 2 1 2 1 1 1 1 2 2 1 2
## [260] 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 2 1 1 2 2 1 2 1 2 1 2 2 1
## [297] 1 2 1 1 1 2 2 1 2 2 2 2 1 1 1 2 2 1 1 2 1 2 2 2 1 2 1 2 2 1 2 2 1 2 1 1 1
## [334] 2 1 1 2 1 1 2 2 2 1 1 1 2 2 2 2 1 1 1 2 1 2 2 2 2 1 2 2 1 1 1 1 2 1 2 1 2
## [371] 2 1 2 2 1 2 2 1 2 1 2 1 2 2 2 2 2 2 1 2 2 2 2 1 1 1 1 1 2 1 2 2 1 1 2 1 2
## [408] 1 1 1 1 1 1 2 2 1 1 2 1 2 1 1 2 1 2 1 2 2 1 2 1 1 2 1 1 1 2 1 1 1 1 2 2 1 1
## [445] 1 1 2 2 2 2 1 1 1 2 1 1 2 2 2 2 1 2 1 1 1 2 1 2 2 2 2 2 1 2 1 1 1 2 2 2 2
## [482] 1 2 2 1 2 2 1 2 1 1 1 2 1 1 1 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 77062.14 84340.07
##  (between_SS / total_SS =  66.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# Check the clusters
#kmeans_model$cluster

# Add cluster labels
#data$kmeans_cluster <- factor(kmeans_model$cluster)
```

The data set are partitioned into 2 clusters.

Within-Cluster Sum of Squares (WSS). WSS measures how compact each cluster is. Lower values means points are closer to their cluster center. [1] 77062.14 84340.07 Cluster 2 is tighter (better fit) than the other WSS values are fairly close, which seems to be good.

Between-Cluster vs Total Sum of Squares (between_SS / total_SS = 66.5 %)

- This is the proportion of variance explained by the clustering.
- 66.5% of the total variance is between clusters.
- Higher % (closer to 100%)specifies clusters are well separated.

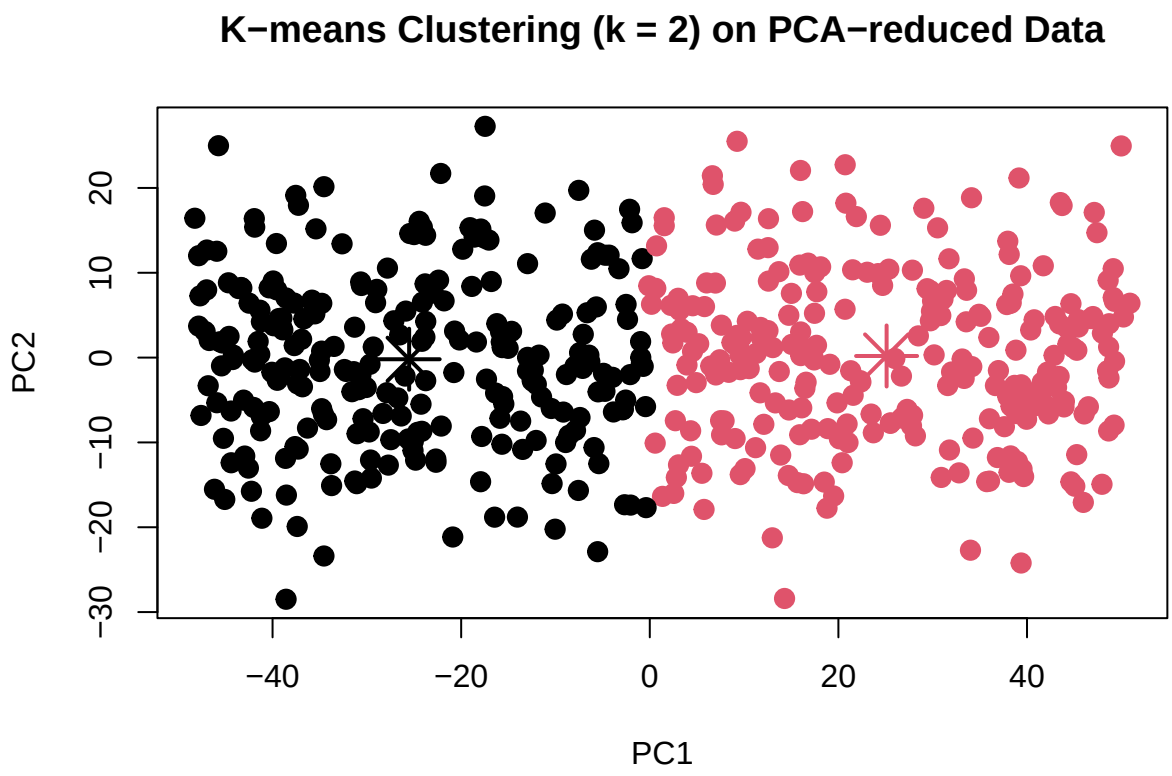
Plot the clusters

```
summary(data)
```

```
##        X1               X2               X3              X4
##  Min.   : 4.888   Min.   : 0.6483   Min.   :21.57   Min.   :2.020
##  1st Qu.: 8.550   1st Qu.:22.6250   1st Qu.:43.03   1st Qu.:3.582
```

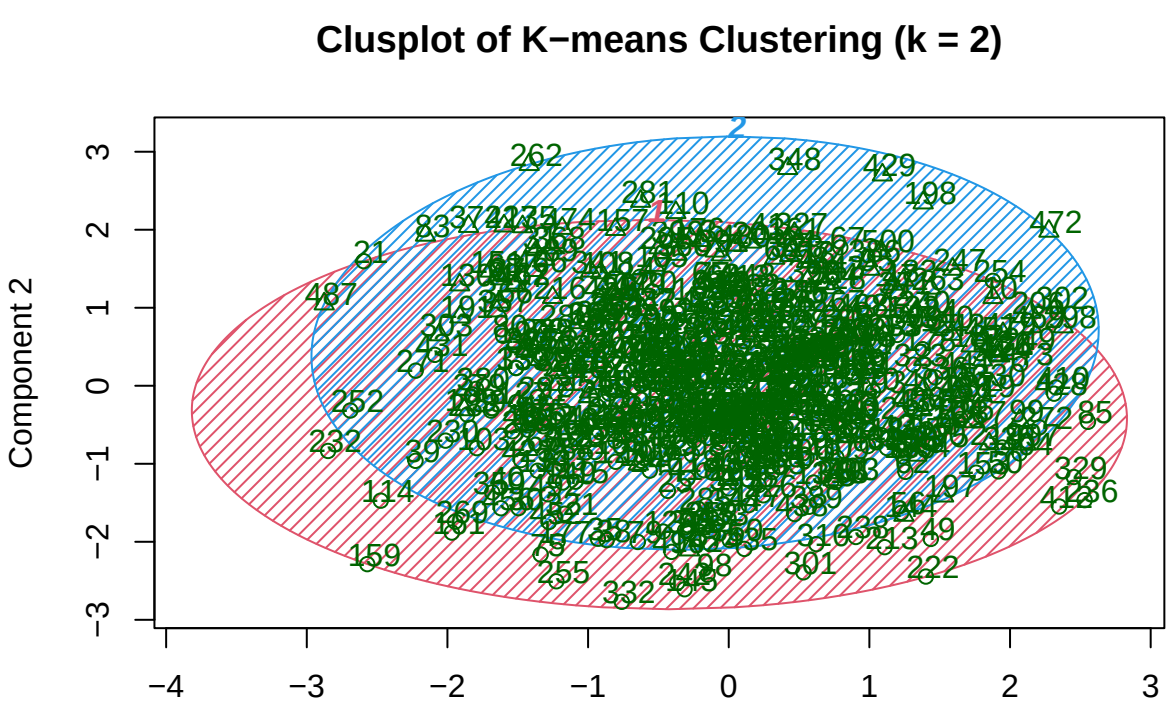
```
##  Median : 9.896   Median :48.8405   Median :50.23   Median :5.329
##  Mean   : 9.947   Mean   :48.4953   Mean   :50.29   Mean   :5.111
##  3rd Qu.:11.327   3rd Qu.:73.3148   3rd Qu.:56.95   3rd Qu.:6.559
##  Max.   :14.563   Max.   :99.5379   Max.   :77.93   Max.   :7.994
##        X5
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :2.036
##  3rd Qu.:3.000
##  Max.   :3.000
```

```r
library(cluster)

# Reduce data to 2D with PCA for visualization.
# Since we have 5 variables it is not possible to visualize
pca_result <- prcomp(data, scale = FALSE)
pca_data <- pca_result$x[, 1:2]  # first two PCs

# Plot cluster
plot(
  pca_data,
  col = kmeans_model$cluster,
  main = "K-means Clustering (k = 2) on PCA-reduced Data",
  xlab = "PC1",
  ylab = "PC2",
  pch = 20,
  cex = 2
)

# Add cluster centers
points(
  aggregate(pca_data, by = list(kmeans_model$cluster), FUN = mean)[, 2:3],
  col = 1:3,
  pch = 8,
  cex = 3,
  lwd = 2
)
```



Cusplot

```
clusplot(
  data,
  kmeans_model$cluster,
  color = TRUE,
  shade = TRUE,
  labels = 2,
  lines = 0,
  main = "Clusplot of K-means Clustering (k = 2)"
)
```



Evaluate with WSS and silhoutte

```
# Silhouette
library(cluster)
sil <- silhouette(kmeans_model$cluster, dist(data))
mean(sil[, 3])  # Average silhouette width
```

```
## [1] 0.5304015
```