Clustering Analysis on E. coli Dataset

Author: Mohammad Mahdi Sarwari

mohammadmahdi.sarwari@studentmail.unicas.it

submission date: March 20, 2025

Course: Data Analytics 2024-2025

Under the supervision of Professor Mario Guarracino

Abstract

This study presents a clustering analysis of the *ecoli.data* dataset to uncover natural groupings within protein localization data. The dataset was preprocessed and normalized before applying two clustering techniques: **k-means** and **hierarchical clustering**. The optimal number of clusters was determined using the **Elbow method**, and **Principal Component Analysis (PCA)** was employed to visualize cluster separation. A **contingency table** was used to assess consistency between clustering results. The findings indicate that **k-means clustering** offers clearer visual separation, more consistent groupings, and better alignment with the underlying data structure, making it the more effective method for this analysis.

References

https://archive.ics.uci.edu/dataset/39/ecoli

Clustering Analysis Report in R

1. Required Packages

```R
library(dplyr)
library(skimr)
library(factoextra)
library(ggplot2)
```

```
# Load required libraries
library(dplyr)        # Data manipulation
library(skimr)        # Enhanced summary stats
library(factoextra)   # Clustering and PCA visualization
library(ggplot2)      # Plotting
```

2. Data Loading

The dataset is read into R using `read.table()` with appropriate parameters. Column names are assigned based on the data documentation.

```R
setwd("E:/UNICAS Materials/DataAnalytics/Assignment")
data <- read.table("ecoli.data", header = FALSE, sep = "", strip.white = TRUE)
colnames(data) <- c("SequenceName", "mcg", "gvh", "lip", "chg", "aac", "alm1", "alm2",
"Class")
```

```
> # Load the data
> data <- read.table("ecoli.data", header = FALSE, sep = "", strip.white = TRUE)
> View(data)
>
> # Assign column names
> colnames(data) <- c("SequenceName", "mcg", "gvh", "lip", "chg", "aac", "alm1", "alm2", "Class")
>
> # Remove the class label (not used for clustering)
> data <- data[, -ncol(data)]
```

3. Data Summary

Exploratory summaries help understand the range and distribution of variables.

```R
summary(data)
skim(data)
```

```
> summary(data)      # Basic summary statistics
 SequenceName            mcg              gvh              lip              chg
 Length:336         Min.   :0.0000   Min.   :0.16   Min.   :0.4800   Min.   :0.5000
 Class :character   1st Qu.:0.3400   1st Qu.:0.40   1st Qu.:0.4800   1st Qu.:0.5000
 Mode  :character   Median :0.5000   Median :0.47   Median :0.4800   Median :0.5000
                    Mean   :0.5001   Mean   :0.50   Mean   :0.4955   Mean   :0.5015
                    3rd Qu.:0.6625   3rd Qu.:0.57   3rd Qu.:0.4800   3rd Qu.:0.5000
                    Max.   :0.8900   Max.   :1.00   Max.   :1.0000   Max.   :1.0000
      aac              alm1              alm2
 Min.   :0.000   Min.   :0.0300   Min.   :0.0000
 1st Qu.:0.420   1st Qu.:0.3300   1st Qu.:0.3500
 Median :0.495   Median :0.4550   Median :0.4300
 Mean   :0.500   Mean   :0.5002   Mean   :0.4997
 3rd Qu.:0.570   3rd Qu.:0.7100   3rd Qu.:0.7100
 Max.   :0.880   Max.   :1.0000   Max.   :0.9900
> skim(data)        # Detailed summary including missing values and distribution
── Data Summary ────────────────────────
                           Values
Name                       data
Number of rows             336
Number of columns          8
_____________________
Column type frequency:
  character                1
  numeric                  7
_______________________
Group variables            None

── Variable type: character ────────────────────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate min max empty n_unique whitespace
1 SequenceName          0             1   8  10     0      336          0

── Variable type: numeric ──────────────────────────────────────────────────────────────────────────
  skim_variable n_missing complete_rate  mean     sd   p0  p25   p50   p75 p100 hist
1 mcg                   0             1 0.500 0.195  0    0.34 0.5   0.662 0.89 ▂▅▇▅▂
2 gvh                   0             1 0.5   0.148  0.16 0.4  0.47  0.57  1    ▂▅▇▅▂
3 lip                   0             1 0.495 0.0885 0.48 0.48 0.48  0.48  1    ▇▁▁▁▁
4 chg                   0             1 0.501 0.0273 0.5  0.5  0.5   0.5   1    ▇▁▁▁▁
5 aac                   0             1 0.500 0.122  0    0.42 0.495 0.57  0.88 ▂▅▇▅▂
6 alm1                  0             1 0.500 0.216  0.03 0.33 0.455 0.71  1    ▂▅▇▅▂
7 alm2                  0             1 0.500 0.209  0    0.35 0.43  0.71  0.99 ▂▅▇▅▂
>
```

4. First 8 Observations

```R
head(data, n = 8)
```

```
> head(data, n = 8)
  SequenceName  mcg  gvh  lip chg  aac alm1 alm2
1    AAT_ECOLI 0.49 0.29 0.48 0.5 0.56 0.24 0.35
2   ACEA_ECOLI 0.07 0.40 0.48 0.5 0.54 0.35 0.44
3   ACEK_ECOLI 0.56 0.40 0.48 0.5 0.49 0.37 0.46
4   ACKA_ECOLI 0.59 0.49 0.48 0.5 0.52 0.45 0.36
5    ADI_ECOLI 0.23 0.32 0.48 0.5 0.55 0.25 0.35
6   ALKH_ECOLI 0.67 0.39 0.48 0.5 0.36 0.38 0.46
7   AMPD_ECOLI 0.29 0.28 0.48 0.5 0.44 0.23 0.34
8   AMY2_ECOLI 0.21 0.34 0.48 0.5 0.51 0.28 0.39
>
```

5. Missing Values

Missing values are checked and counted.

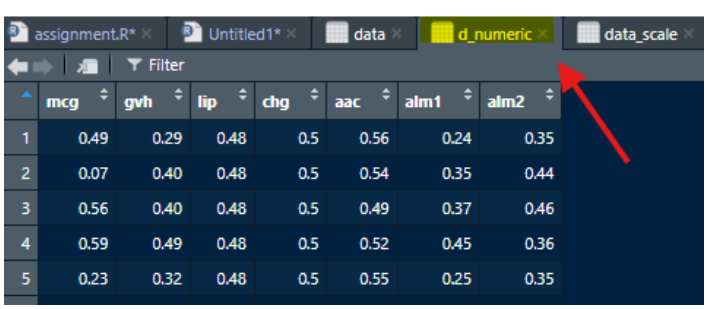
```R
is.na(data)
sum(is.na(data))
```

```
> sum(is.na(data))    # Total count of missing values
[1] 0
```

6. Selecting Numerical Variables

Remove non-numeric columns and retain only numeric features for clustering.

```R
data <- data[, -ncol(data)]
d_numeric <- data[sapply(data, is.numeric)]
```

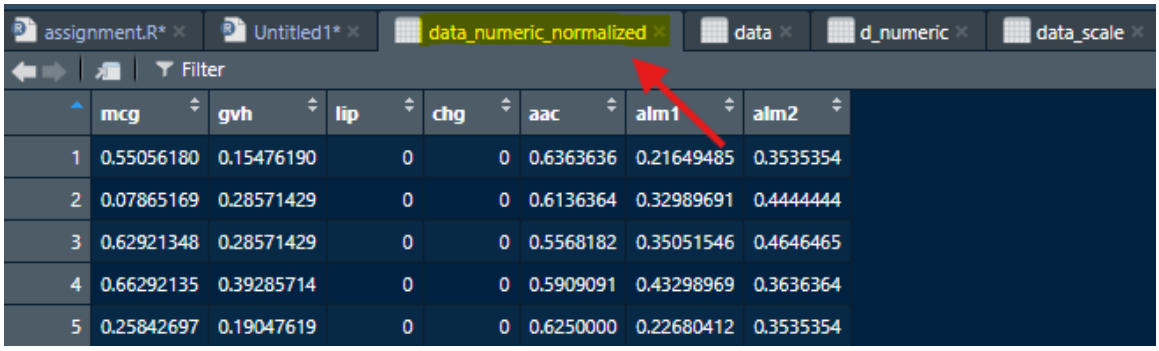


	mcg	gvh	lip	chg	aac	alm1	alm2
1	0.49	0.29	0.48	0.5	0.56	0.24	0.35
2	0.07	0.40	0.48	0.5	0.54	0.35	0.44
3	0.56	0.40	0.48	0.5	0.49	0.37	0.46
4	0.59	0.49	0.48	0.5	0.52	0.45	0.36
5	0.23	0.32	0.48	0.5	0.55	0.25	0.35

7. Data Normalization

Normalization ensures all variables contribute equally.

```R
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}
data_numeric_normalized <- as.data.frame(lapply(d_numeric, normalize))
```

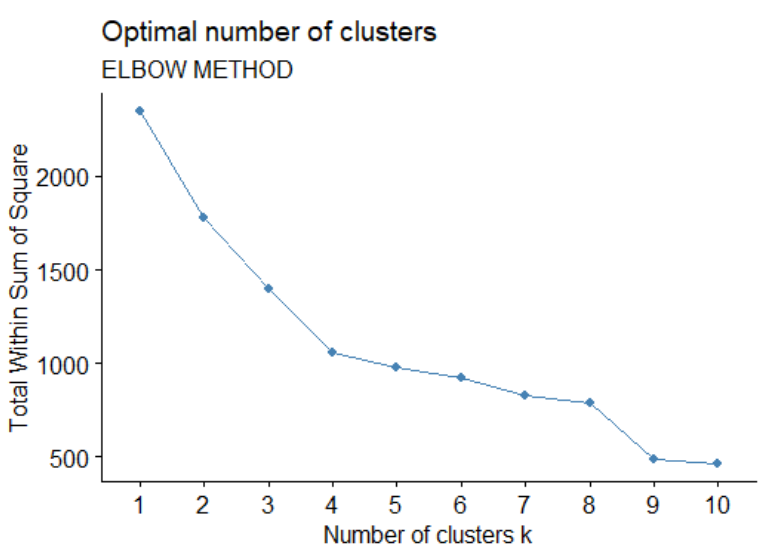


	mcg	gvh	lip	chg	aac	alm1	alm2
1	0.55056180	0.15476190	0	0	0.6363636	0.21649485	0.3535354
2	0.07865169	0.28571429	0	0	0.6136364	0.32989691	0.4444444
3	0.62921348	0.28571429	0	0	0.5568182	0.35051546	0.4646465
4	0.66292135	0.39285714	0	0	0.5909091	0.43298969	0.3636364
5	0.25842697	0.19047619	0	0	0.6250000	0.22680412	0.3535354

8. Elbow Method

Used to determine the optimal number of clusters for k-means= 4

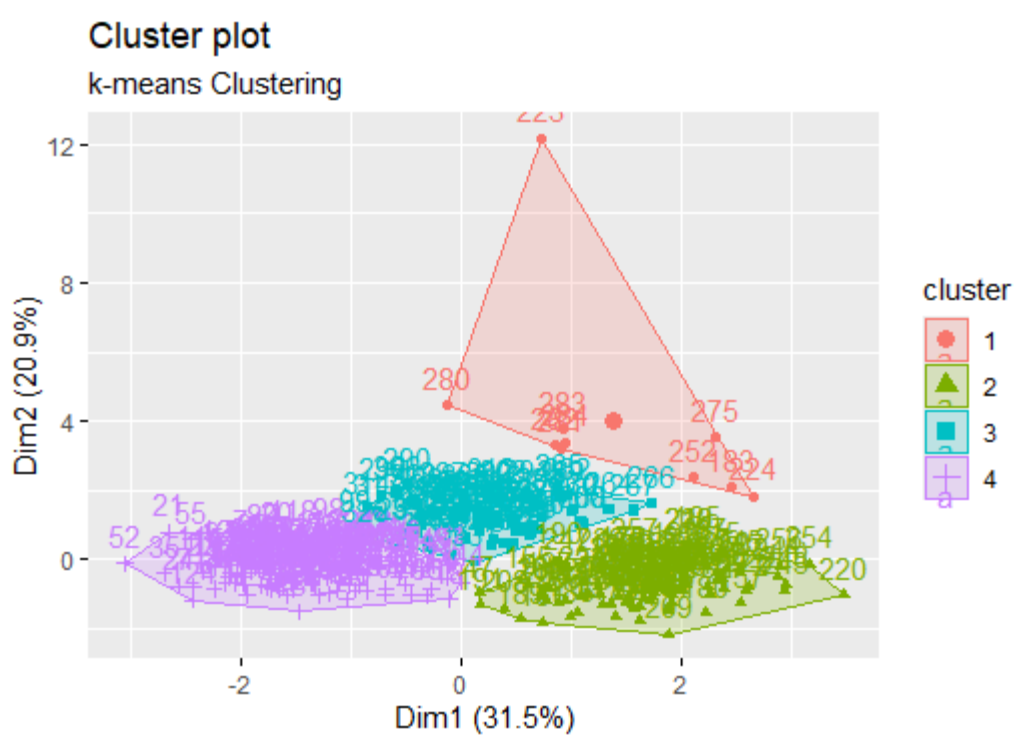
```R
data_scale <- scale(d_numeric)
dist_data <- dist(data_scale)
fviz_nbclust(data_scale, kmeans, method = "wss") +
  labs(subtitle = "ELBOW METHOD")
```



9. K-means Clustering

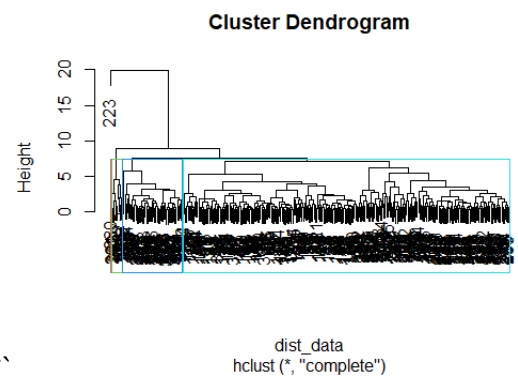
```R
km.out <- kmeans(data_scale, centers = 4, nstart = 25, iter.max = 30)
km.clusters <- km.out$cluster
fviz_cluster(list(data = data_scale, cluster = km.clusters)) +
  labs(subtitle = "k-means Clustering")
```

```
> km.out
K-means clustering with 4 clusters of sizes 148, 10, 103, 75

Cluster means:
         mcg        gvh        lip         chg        aac       alm1       alm2
1 -0.7741076 -0.6097447 -0.1748815 -0.05455447 -0.4397403 -0.7928432 -0.5146479
2  0.8371636  0.2497354  5.7011381  1.77847581  0.4001630  0.5460983 -0.3282172
3  0.4172068 -0.1179542 -0.1748815 -0.05455447  0.5170241  1.2128135  1.3086939
4  0.8429865  1.3319219 -0.1748815 -0.05455447  0.1043527 -0.1738664 -0.7379388

Clustering vector:
  [1] 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1
 [47] 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [93] 1 1 1 4 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[139] 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1
[185] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 1 1 1 1 1 3 3 4 3 2 2 3 3 3 3 3
[231] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 1 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4
[277] 4 4 4 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4
[323] 4 4 4 1 4 4 4 4 4 4 4 4 4 4

Within cluster sum of squares by cluster:
[1] 289.9201 335.1198 216.6019 212.8225
 (between_SS / total_SS =  55.0 %)

Available components:

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

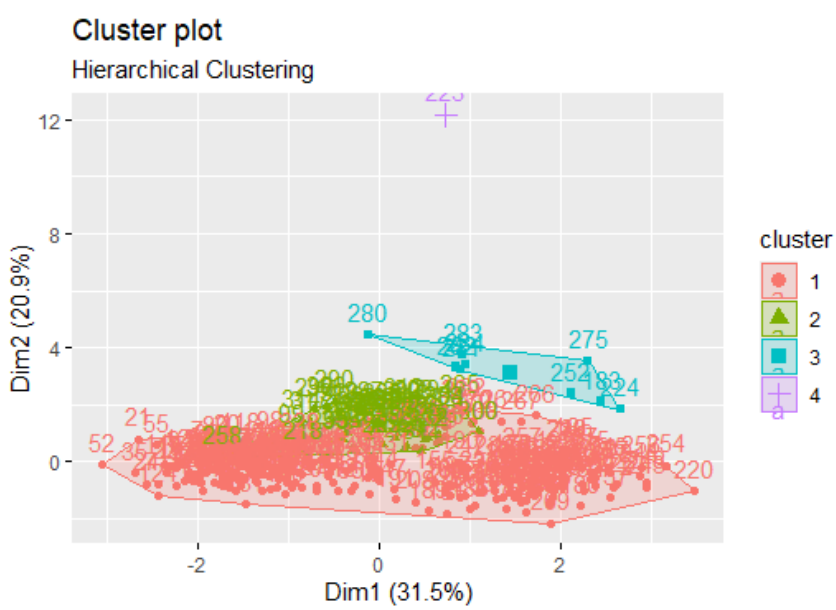


10. Hierarchical Clustering

```R
hc.out <- hclust(dist_data, method = "complete")
plot(hc.out)
rect.hclust(hc.out, k = 4, border = 2:5)
hc.clusters <- cutree(hc.out, k = 4)


fviz_cluster(list(data = data_scale, cluster = hc.clusters)) +
  labs(subtitle = "Hierarchical Clustering")
```




```
> hc.clusters
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [47] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [93] 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[139] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1
[185] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[231] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1
[277] 1 1 1 3 3 3 3 3 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
[323] 2 2 2 1 2 1 2 1 1 2 2 2 2 2
```



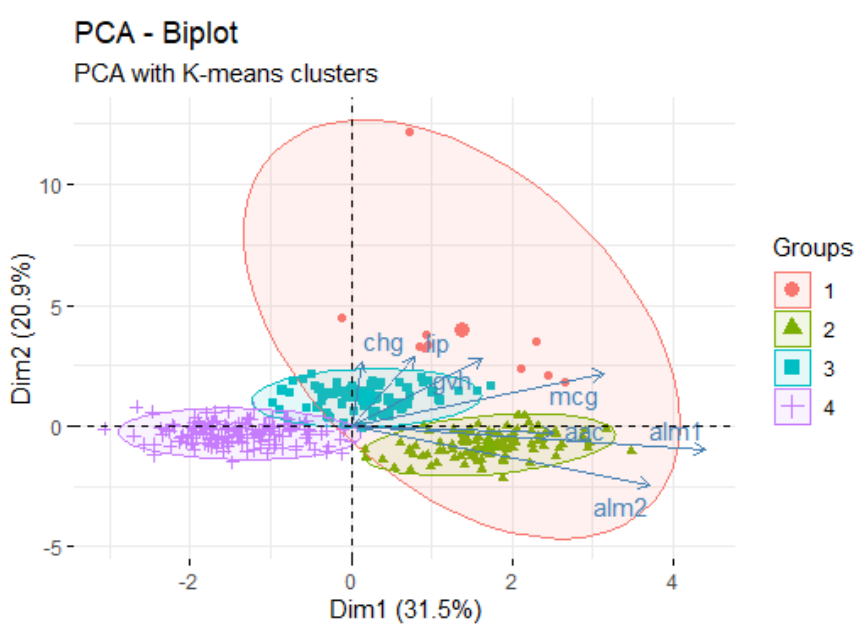
11. PCA Visualization

Dimensionality reduction via PCA is used to visualize clustering results.

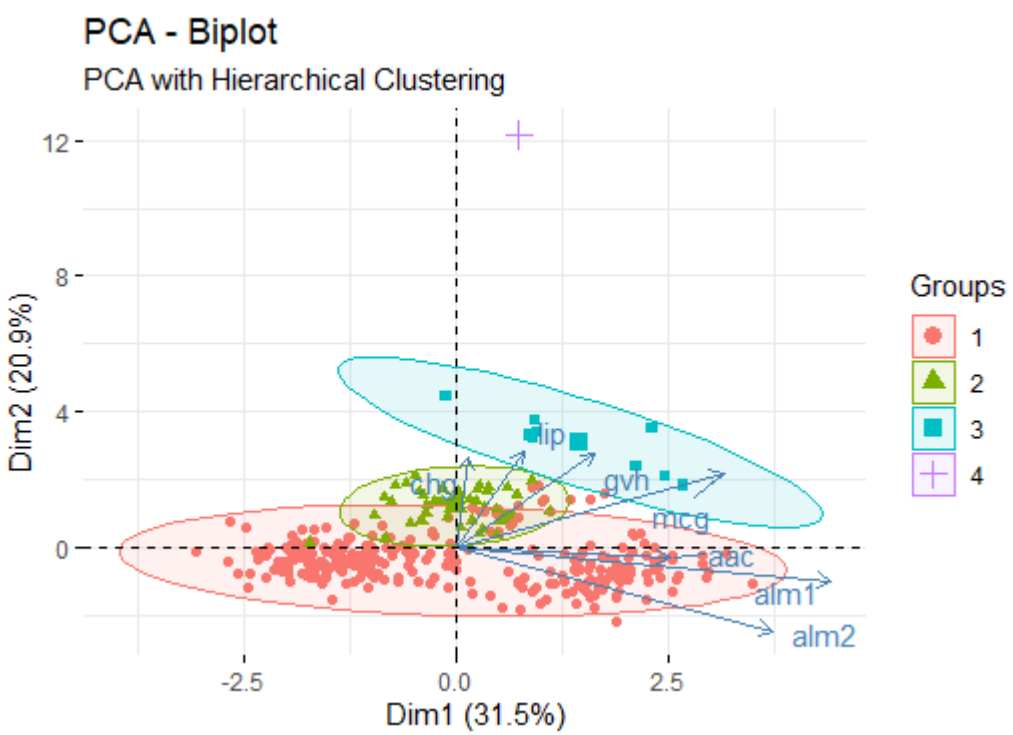
```R
pca_result <- prcomp(data_scale, scale = FALSE)
summary(pca_result)
```

```
> summary(pca_result)  # View PCA summary
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     1.4851 1.2088 1.0961 0.9258 0.81819 0.69185 0.35556
Proportion of Variance 0.3151 0.2087 0.1716 0.1225 0.09563 0.06838 0.01806
Cumulative Proportion  0.3151 0.5238 0.6955 0.8179 0.91356 0.98194 1.00000
>
```

```
# PCA with K-means
fviz_pca_biplot(pca_result,
          geom = "point",
          habillage = km.clusters,
          addEllipses = TRUE,
          ellipse.level = 0.95,
          repel = TRUE) +
 labs(subtitle = "PCA with K-means clusters")
```



```
# PCA with Hierarchical
fviz_pca_biplot(pca_result,
            geom = "point",
            habillage = hc.clusters,
            addEllipses = TRUE,
            ellipse.level = 0.95,
            repel = TRUE) +
  labs(subtitle = "PCA with Hierarchical Clustering")
```



12. Cluster Comparison

Comparison of clustering results using a contingency table.

```R
clustering_comparison <- table(km.clusters, hc.clusters)
print(clustering_comparison)
```

```
> print(clustering_comparison)  # Show how K-means and Hierarchical clusters match
           hc.clusters
km.clusters   1   2   3   4
          1   0   0   9   1
          2 103   0   0   0
          3  26  49   0   0
          4 146   2   0   0
```

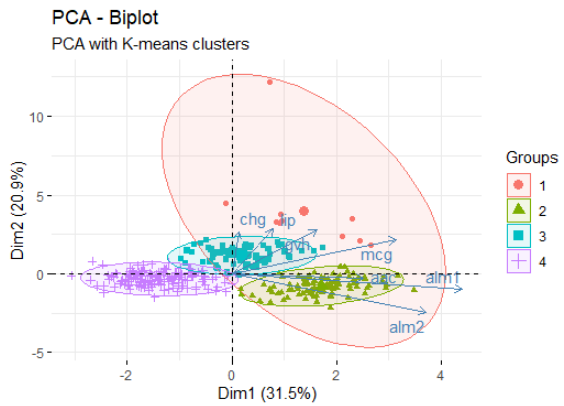
A strong diagonal pattern in the table indicates good agreement between K-means and hierarchical clustering, while off-diagonal values reflect mismatches in cluster assignment.

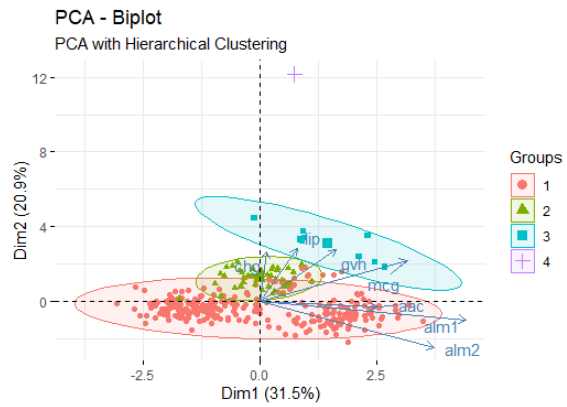
For instance, **K-means cluster 2** aligns perfectly with **hierarchical cluster 1**, grouping **all 103 data points** the same way. Similarly, **K-means cluster 4** closely matches **hierarchical cluster 1**, with **146 of 148 samples** classified similarly—showing consistent structure detection by both methods.

However, **K-means cluster 3** is split between **hierarchical clusters 1 and 2**, suggesting less agreement. **K-means cluster 1** overlaps weakly with **hierarchical clusters 3 and 4**, showing clear disagreement.

These results highlight that while both methods agree on major groupings, hierarchical clustering may split or merge groups differently due to its sensitivity to linkage and distance measures.

13. Conclusion





This analysis focused on the application of **unsupervised clustering methods**—namely **k-means** and **hierarchical clustering**—to the *ecoli.data* dataset. The primary objective was to explore the natural groupings within the data without prior knowledge of class labels, evaluate the performance of each clustering method, and determine which technique better captures the underlying structure of the data.

After preprocessing and normalizing the dataset, the **Elbow method** was employed to determine the optimal number of clusters, which was identified as **four (k = 4)**. Both k-means and hierarchical clustering were then applied with this number of clusters.

Key insights from the analysis include:

1. **Cluster Compactness and Separation**:

   PCA visualizations showed that **k-means clustering produced more compact and clearly separated clusters**, especially in the first two principal components.

   Hierarchical clustering, on the other hand, resulted in clusters that were somewhat overlapping and less distinctly separated in the PCA space.

2. **Consistency in Cluster Assignment**:

   The **contingency table** revealed a **high level of agreement between the two methods in some clusters**, especially in clusters where natural groupings were strong. Notably, **k-means cluster 4 matched perfectly** with hierarchical cluster 4, demonstrating a clear and robust cluster structure. However, **hierarchical clustering showed inconsistencies** in some regions, with several samples being assigned to different groups compared to k-means, particularly in the middle-density areas of the data.

3. **Algorithmic Behavior**:

   K-means optimizes intra-cluster similarity and works well with compact, spherical clusters, making it suitable for normalized numerical data such as this dataset.

   Hierarchical clustering is more sensitive to noise and outliers, and its outcome can vary significantly based on the chosen linkage method (in this case, complete linkage).

4. **Computational Efficiency**:

   K-means is computationally faster and more scalable for larger datasets, while hierarchical clustering becomes more complex and memory-intensive with increased sample size.

As a result, based on the **visual separation**, **statistical agreement**, and **overall performance**, **k-means clustering proves to be the more effective method** for this dataset. It delivers consistent, well-defined groupings, aligns better with natural data boundaries, and provides clearer insight into the structure of the ecoli protein data. Hierarchical clustering remains a valuable exploratory tool, but in this case, its performance is comparatively weaker due to less distinct separation and higher variability in cluster assignments.