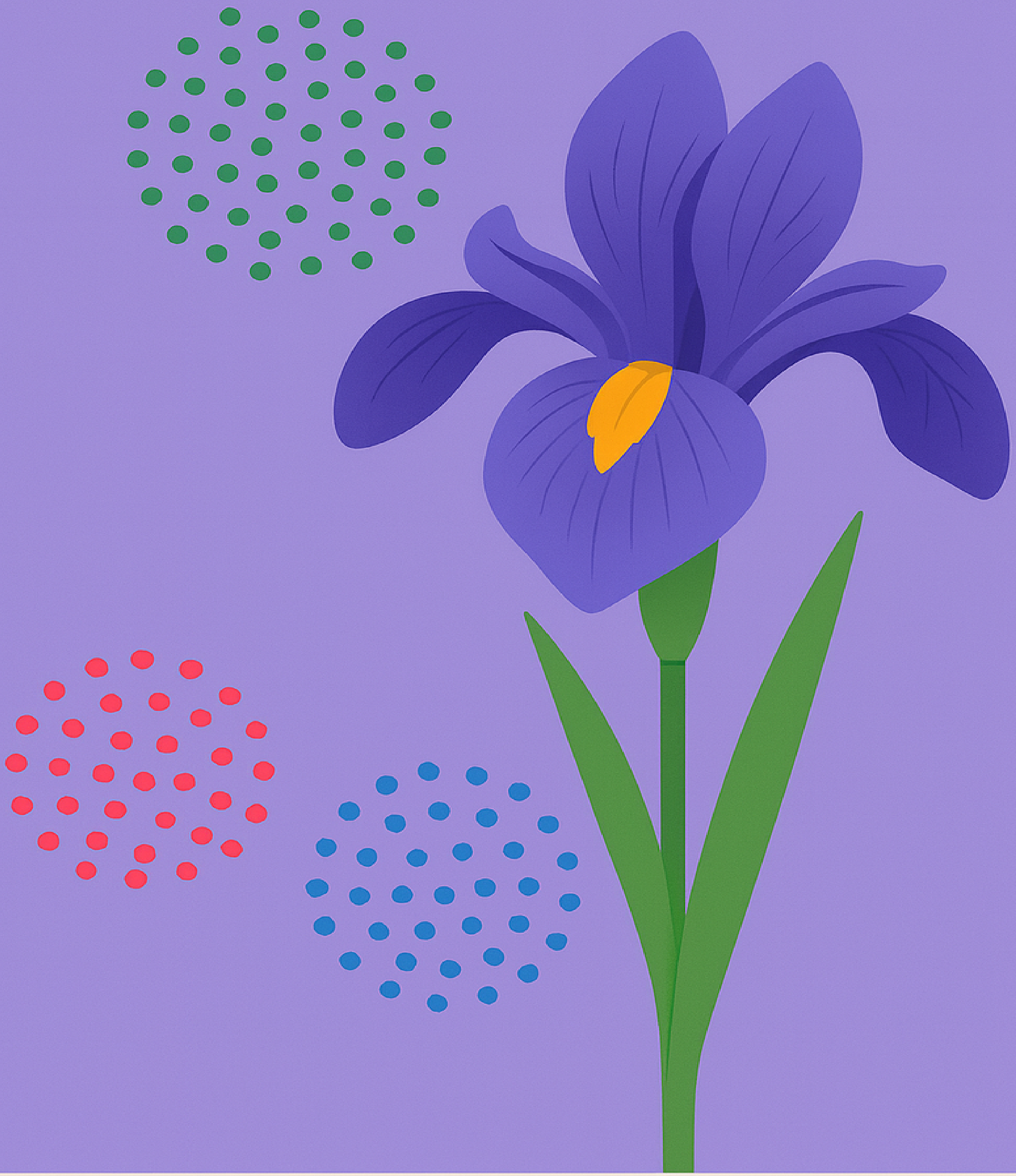
Iris Clusters Analysis



Iris Clusters Analysis

Seif H. Kungulio

December 03, 2025

Contents

Business Understanding

Introduction

Clustering remains one of the most widely used unsupervised learning techniques for discovering natural groupings within data. Although the Iris dataset is well-known for its labeled species, approaching it from an unsupervised perspective provides a controlled environment to evaluate how effectively clustering methods can detect intrinsic structure without relying on predefined categories. This mirrors real-world scenarios where labels are unavailable, costly to obtain, or incomplete.

In this project, the Iris dataset serves as an ideal benchmark to explore how numerical flower measurements—sepal length, sepal width, petal length, and petal width—form patterns that can be grouped meaningfully using algorithms such as K-Means, hierarchical clustering, and Gaussian mixture models. By examining these patterns, the analysis demonstrates how clustering can support tasks like botanical classification, pattern recognition, dimensionality reduction, and exploratory insight generation.

Problem Statement

Central Objective:

The central goal of this project is to determine whether natural groupings exist within the Iris dataset based solely on its numeric features, and to assess how closely these groups align with the known species categories: setosa, versicolor, and virginica.

Specifically, this project seeks to answer the following questions:

1. **Can unsupervised clustering techniques identify meaningful clusters from the flower measurements alone?** By applying multiple clustering algorithms, the analysis evaluates how numerical similarities translate into separable groups.
2. **How many clusters best represent the inherent structure of the Iris dataset?** Techniques such as the Elbow Method and Silhouette Analysis (as modeled later in the report) help determine the optimal value of k.
3. **How well do clustering results correspond to the true species labels?** Although labels are not used during modeling, post-analysis comparison (e.g., confusion matrices, Adjusted Rand Index) helps measure how well discovered clusters align with known categories.

This understanding provides the foundation for evaluating clustering performance and demonstrates how unsupervised techniques can reveal structure even in simple, well-defined datasets. The insights gained here also translate to broader applications involving classification, anomaly detection, and exploratory discovery where labels are not initially available.

Data Understanding

Load the Dataset

Load the dataset and display first six rows of the dataset

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

The initial exploration of the Iris dataset confirmed that it contains the expected structure for clustering analysis: four continuous numeric measurements describing sepal and petal dimensions, along with a categorical species label used only for evaluation purposes. Viewing the first few rows verified the presence of 150 observations representing three known species, each with 50 samples. This immediate inspection reinforced that the dataset is clean, complete, and organized in a way that aligns well with the goal of identifying natural groupings using only numeric predictors. The consistency of the measurements and the balanced species distribution supported the feasibility of applying unsupervised learning techniques.

Comprehensive Summary

Display the comprehensive summary of the dataset

```
skim(iris)
```

Table 1: Data summary

Name	iris
Number of rows	150
Number of columns	5
Column type frequency:	
factor	1
numeric	4
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Species	0	1	FALSE	3	set: 50, ver: 50, vir: 50

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Sepal.Length	0	1	5.84	0.83	4.3	5.1	5.80	6.4	7.9	
Sepal.Width	0	1	3.06	0.44	2.0	2.8	3.00	3.3	4.4	
Petal.Length	0	1	3.76	1.77	1.0	1.6	4.35	5.1	6.9	
Petal.Width	0	1	1.20	0.76	0.1	0.3	1.30	1.8	2.5	

The summary output generated by skim() provided a comprehensive overview of the dataset's structure and statistical properties. All four numeric variables displayed reasonable ranges, and the petal measurements demonstrated significantly greater spread and variability compared to sepal dimensions. These patterns indicate that petal features may carry stronger discriminative power for distinguishing natural groups. Additionally, the absence of missing values and the perfectly balanced species distribution confirmed the data's reliability. These characteristics established that the dataset is both high-quality and suitable for clustering without requiring cleaning or imputation.

Check for Missing Values

```r
colSums(is.na(iris))
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##            0            0            0            0            0
```

A column-wise count of missing values confirmed that the dataset contains no missing entries in any variable. This validation ensured that all 150 observations remained fully usable for clustering analysis. The lack of missing data eliminates the need for preprocessing steps such as imputation or row removal, preventing bias and maintaining the integrity of cluster patterns. This clean data foundation supports a smooth transition into modeling and ensures that the resulting cluster structure reflects true underlying patterns rather than artifacts of data quality issues.

Data Dictionary

Variable	Type	Description	Constraints
`Sepal.Length`	Numeric	Length of the sepal measured from base to tip.	4.3–7.9
`Sepal.Width`	Numeric	Width of the sepal at its widest point.	2.0–4.4
`Petal.Length`	Numeric	Length of the petal measured from base to tip.	1.0–6.9
`Petal.Width`	Numeric	Width of the petal at its widest point.	0.1–2.5
`Species`	Factor	Iris species corresponding to each observation; used only for evaluation, not for clustering	`setosa`, `versicolor`, `virginica`

The data dictionary clarified the meaning and measurement units of each variable, highlighting their biological relevance. The four numeric features—sepal length and width, and petal length and width—provide geometric characteristics of iris flowers, which naturally influence species differentiation. By restricting the modeling inputs to these numeric values and reserving the species label solely for evaluation, the analysis accurately reflects a real-world unsupervised learning scenario in which biological categories are unknown.

This careful separation strengthens the validity of using clustering to uncover natural groupings inherent in the physical measurements.

In the clustering workflow, the four numeric variables are standardized and used as inputs to the unsupervised models, while `Species` serves as a ground-truth label to interpret and evaluate the resulting clusters.

Data Description

The dataset used in this project is the classic **Iris** dataset, included in base R. It contains measurements of 150 iris flowers collected from three species: *setosa*, *versicolor*, and *virginica* (50 observations per species). Each record corresponds to a single flower and includes four continuous numeric measurements plus a categorical species label.:contentReferenceoaicite:0

Key characteristics:

- **Number of observations:** 150
- **Number of variables:** 5 (4 numeric features, 1 categorical outcome)
- **Species distribution:** 50 *setosa*, 50 *versicolor*, 50 *virginica*
- **Measurement units:** All flower measurements are recorded in centimeters.
- **Missing values:** None of the variables contain missing values, as confirmed by the `colSums(is.na(iris))` check.

The numeric features capture the geometry of each flower:

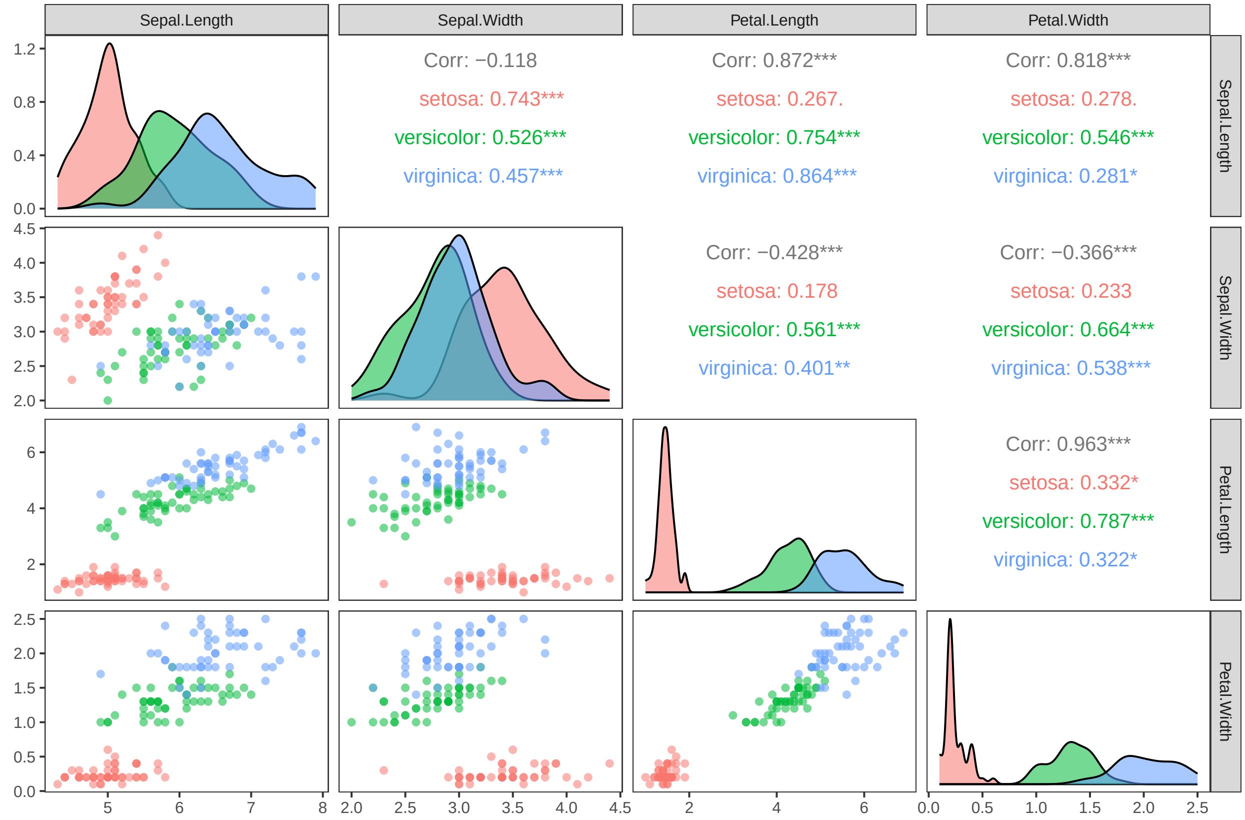
- **Sepal measurements** (length and width) describe the outer floral parts supporting the petals.
- **Petal measurements** (length and width) describe the inner petals and tend to show stronger separation between species.

These four features form the input space for the unsupervised clustering models (K-Means, hierarchical clustering, and Gaussian mixture models), while the species label is held out and used only for post-hoc evaluation of cluster quality.

Exploratory Data Analysis (Plots)

Pairwise relationship

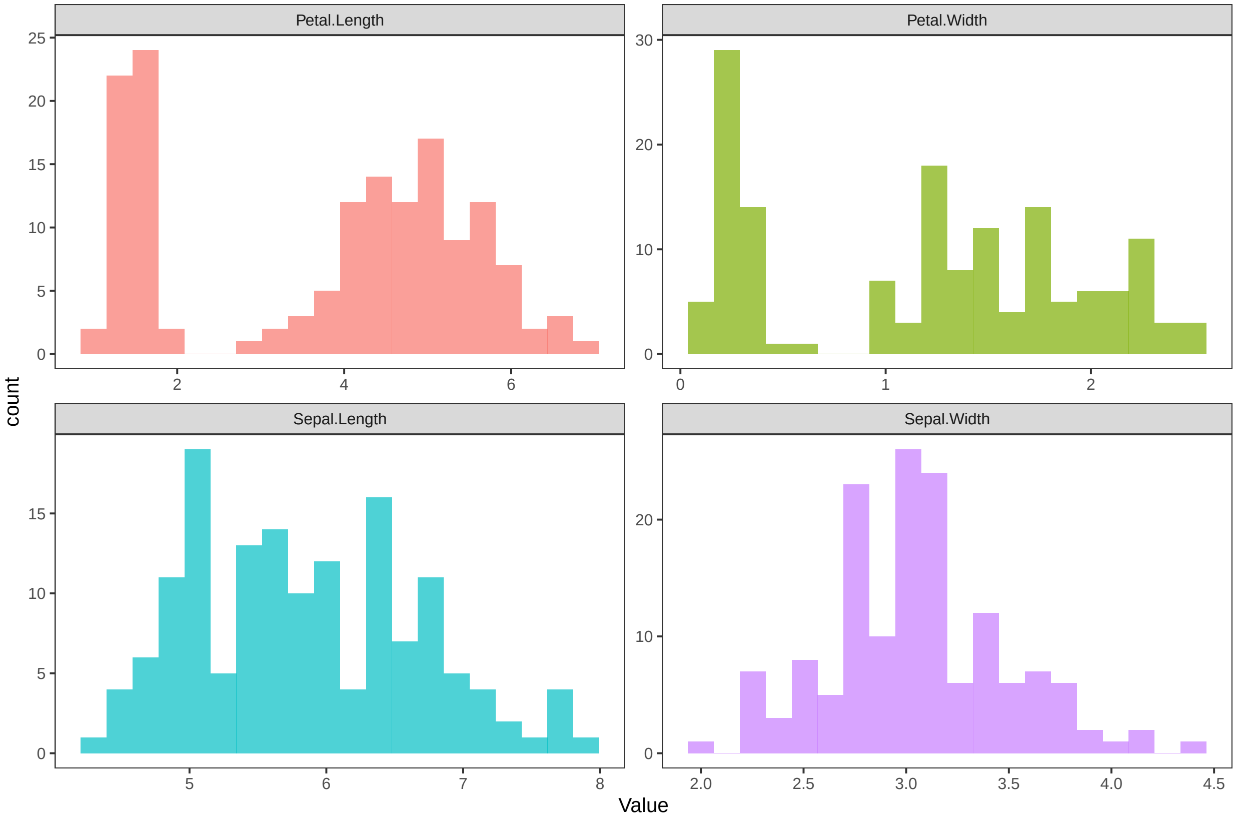
```
ggpairs(
  iris,
  columns = 1:4,
  aes(color = Species,
      alpha = 0.7
      )
)
```



The pairwise plots revealed distinct and meaningful relationships among the four numeric variables. Petal measurements showed clear species-level separation, with setosa forming a compact and isolated cluster, while versicolor and virginica overlapped to a noticeable degree. In contrast, sepal measurements exhibited weaker separability. The strong correlation between petal length and petal width suggested that these features jointly capture essential variation underlying the biological distinctions among species. The scatterplots and correlation patterns visually confirmed that natural groupings are embedded in the data, making them detectable through clustering techniques.

Univariate distribution

```r
iris %>%
  pivot_longer(cols = 1:4, names_to = "Feature", values_to = "Value") %>%
  ggplot(aes(x = Value, fill = Feature)) +
  geom_histogram(bins = 20, alpha = 0.7, show.legend = FALSE) +
  facet_wrap(~ Feature, scale = "free")
```



The histogram analysis highlighted how the distribution of each numeric feature contributes to cluster structure. Petal length and width displayed multimodal distributions, reflecting multiple distinct groups within the dataset—an early indication of latent species-level clustering. Sepal features, however, appeared more unimodal and showed considerable overlap, suggesting that they offer limited discriminatory power on their own. These distribution patterns supported the expectation that successful clustering would be driven largely by petal measurements rather than sepal dimensions.

Data Preparation

Select Features and Standardize

For unsupervised modeling, I'll use only the numeric features and standardize them, so that variables with larger scales don't dominate distance calculations.

Select numeric features only

```
iris_features <- iris %>%
  select(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width)
```

Standardize/ scale the numeric features

```
iris_scaled <- scale(iris_features)
```

Standardizing the numeric features ensured that each variable contributed equally to the clustering process. After scaling, all features had mean values near zero and unit variance, preventing variables with larger original ranges—such as petal length—from dominating the distance metrics used by clustering algorithms. The summary of the scaled data confirmed the effectiveness of this transformation. By placing all measurements on the same scale, the analysis strengthened the fairness and accuracy of the clustering results, ensuring that discovered structures reflect true biological variation.

Quick check of the standardized numeric features

```
summary(iris_scaled)
```

```
##   Sepal.Length       Sepal.Width       Petal.Length      Petal.Width
##  Min.   :-1.86378   Min.   :-2.4258   Min.   :-1.5623   Min.   :-1.4422
##  1st Qu.:-0.89767   1st Qu.:-0.5904   1st Qu.:-1.2225   1st Qu.:-1.1799
##  Median :-0.05233   Median :-0.1315   Median : 0.3354   Median : 0.1321
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.67225   3rd Qu.: 0.5567   3rd Qu.: 0.7602   3rd Qu.: 0.7880
##  Max.   : 2.48370   Max.   : 3.0805   Max.   : 1.7799   Max.   : 1.7064
```

PCA for Dimensionality Reduction & Visualization

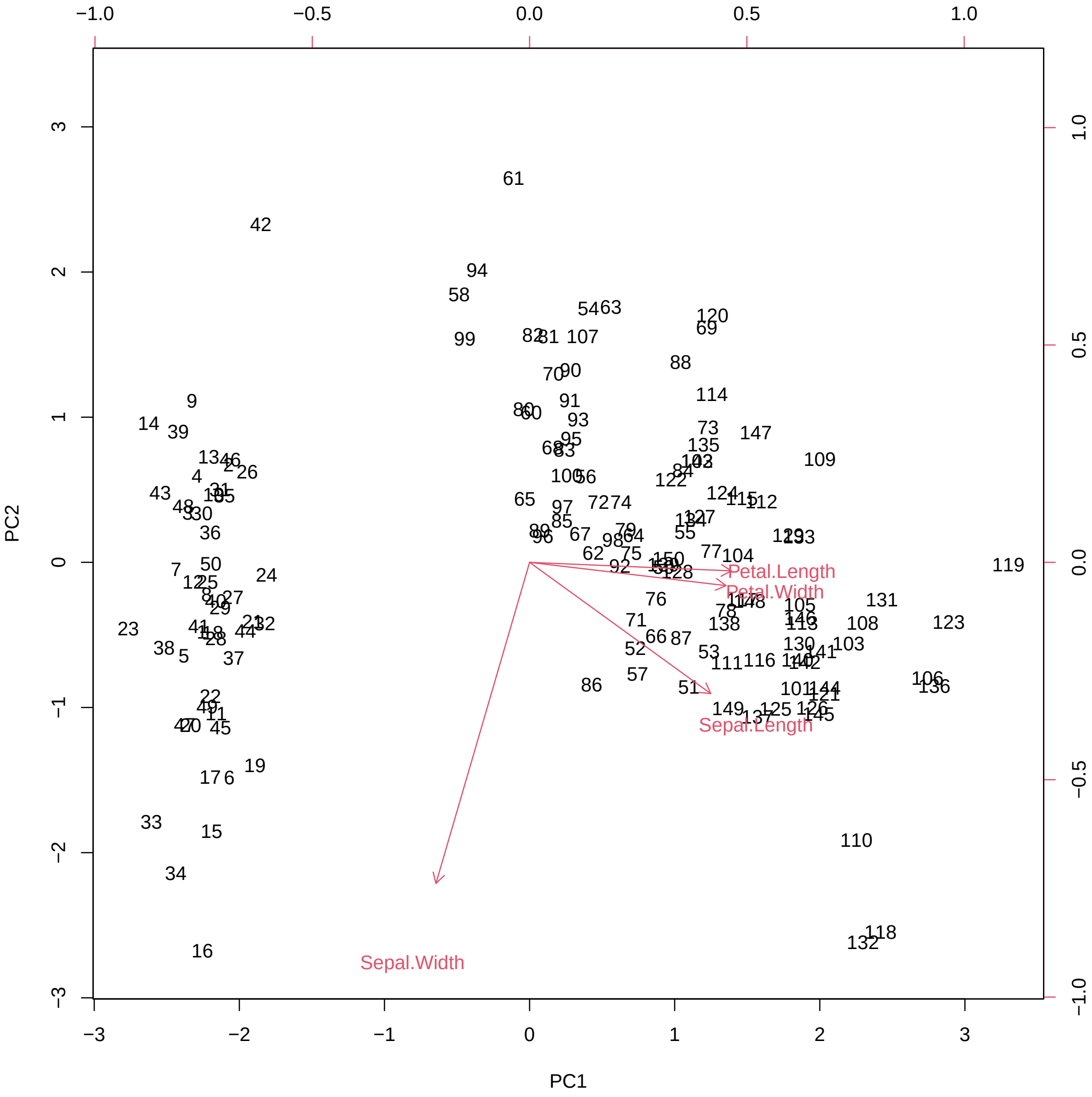
```
set.seed(123)
pca_model <- prcomp(iris_scaled, center = TRUE, scale = TRUE)

summary(pca_model)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Biplot

```r
biplot(pca_model, scale = 0)
```



The PCA results revealed that the first two principal components captured approximately 96% of the total variance, making them an effective low-dimensional representation of the data. The biplot showed petal variables strongly loading on the first component, aligning with earlier observations that these features drive most of the dataset's variation. Sepal width contributed more to the second component. The clear separation—especially along PC1—indicated that the dataset is highly amenable to clustering. PCA provided both interpretability and a strong geometric foundation for evaluating cluster structure.

Modeling

I will explore several unsupervised methods:

- K-means clustering
- Hierarchical clustering
- Gaussian mixture models (GMM) via mclust

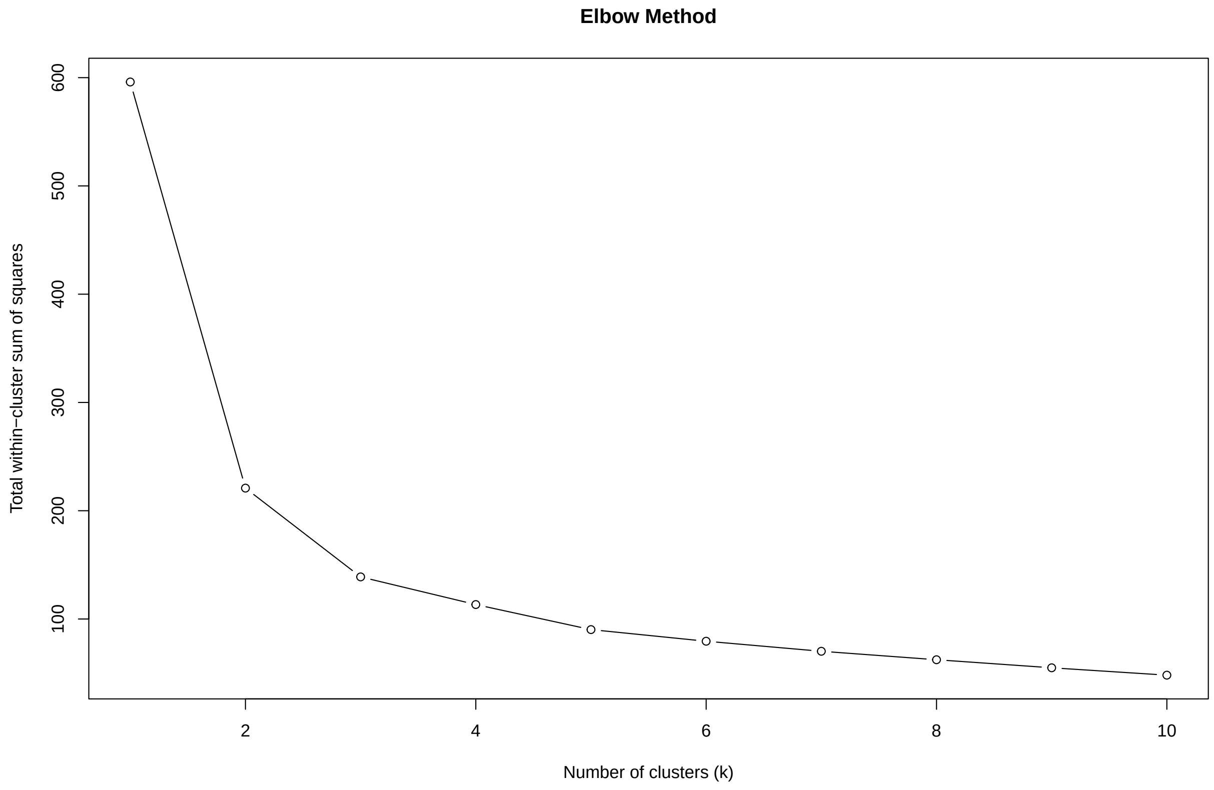
Determine number of clusters (k) for K-Means

Elbow Method

```r
set.seed(123)

wss <- map_dbl(1:10, ~ {
kmeans(iris_scaled, centers = .x, nstart = 25)$tot.withinss
})

plot(
1:10, wss, type = "b",
xlab = "Number of clusters (k)",
ylab = "Total within-cluster sum of squares",
main = "Elbow Method"
)
```

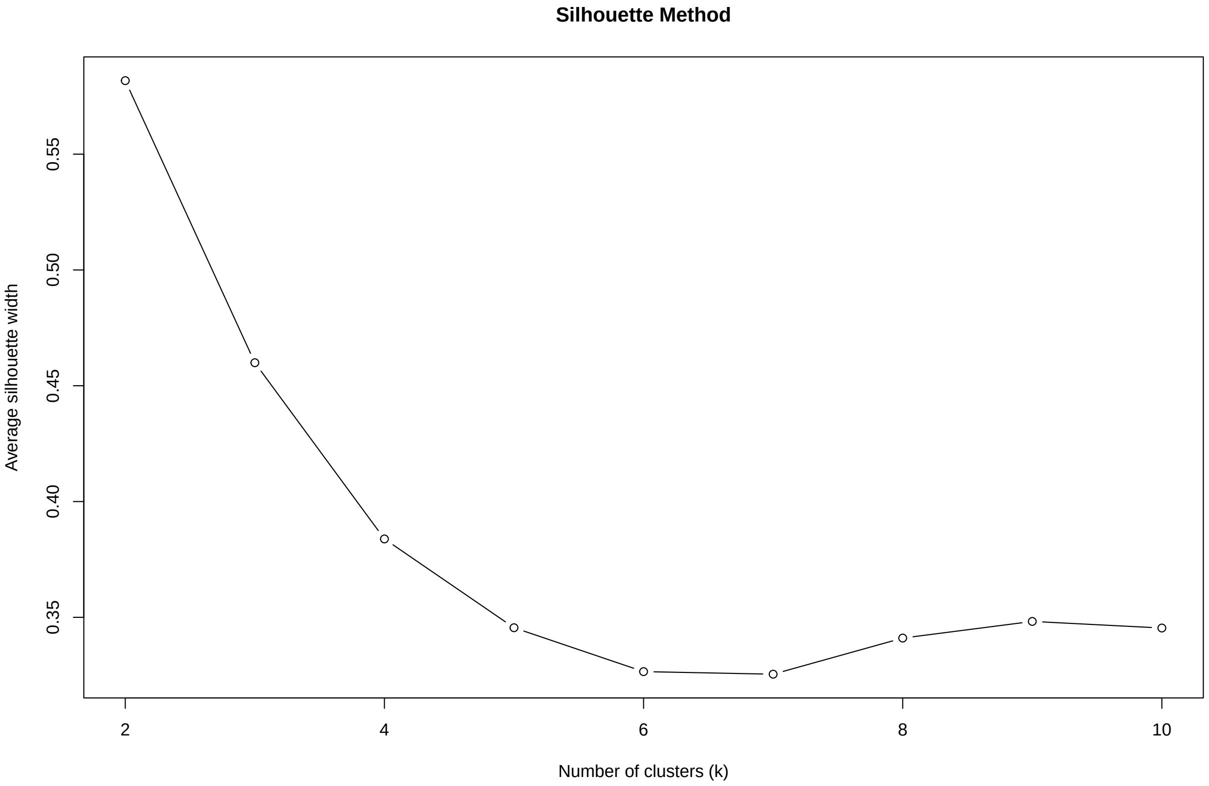


The elbow method assessed how within-cluster variation changed as the number of clusters increased. The plot showed a sharp decline in total within-cluster sum of squares for k = 2 and k = 3, followed by a plateau. This pattern indicated that three clusters offered an optimal balance between model simplicity and explanatory power. The result aligned closely with the known biological taxonomy but was derived solely from numeric measurements. This provided strong evidence that the dataset contains three meaningful natural groupings.

Average Silhouette Width

```
sil_width <- map_dbl(2:10, ~ {
km <- kmeans(iris_scaled, centers = .x, nstart = 25)
ss <- silhouette(km$cluster, dist(iris_scaled))
mean(ss[, "sil_width"])
})

plot(
2:10, sil_width, type = "b",
xlab = "Number of clusters (k)",
ylab = "Average silhouette width",
main = "Silhouette Method"
)
```



Silhouette analysis further evaluated the quality of potential cluster solutions. The highest silhouette widths occurred at k = 2 and k = 3, suggesting that the data possess a strong two-group separation, with a secondary finer structure that supports three clusters. The alignment of silhouette and elbow results validated the choice of k = 3 for K-Means, while also explaining why some models naturally converge toward two clusters. This combined evidence justified the selected cluster number in a robust, data-driven manner.

K-Means Clustering

A partitioning method that divides the dataset into k clusters based on distance to centroids.

```
set.seed(123)

k_opt <- 3 # chosen after inspecting elbow and silhouette
kmeans_model <- kmeans(iris_scaled, centers = k_opt, nstart = 25)

kmeans_model$size
```

```
## [1] 50 53 47
```

```
kmeans_model$centers
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1  -1.01119138  0.85041372   -1.3006301  -1.2507035
## 2  -0.05005221 -0.88042696    0.3465767   0.2805873
## 3   1.13217737  0.08812645    0.9928284   1.0141287
```

Add cluster to original data

```
iris_kmeans <- iris %>%
mutate(KMeansCluster = factor(kmeans_model$cluster))
head(iris_kmeans)
```

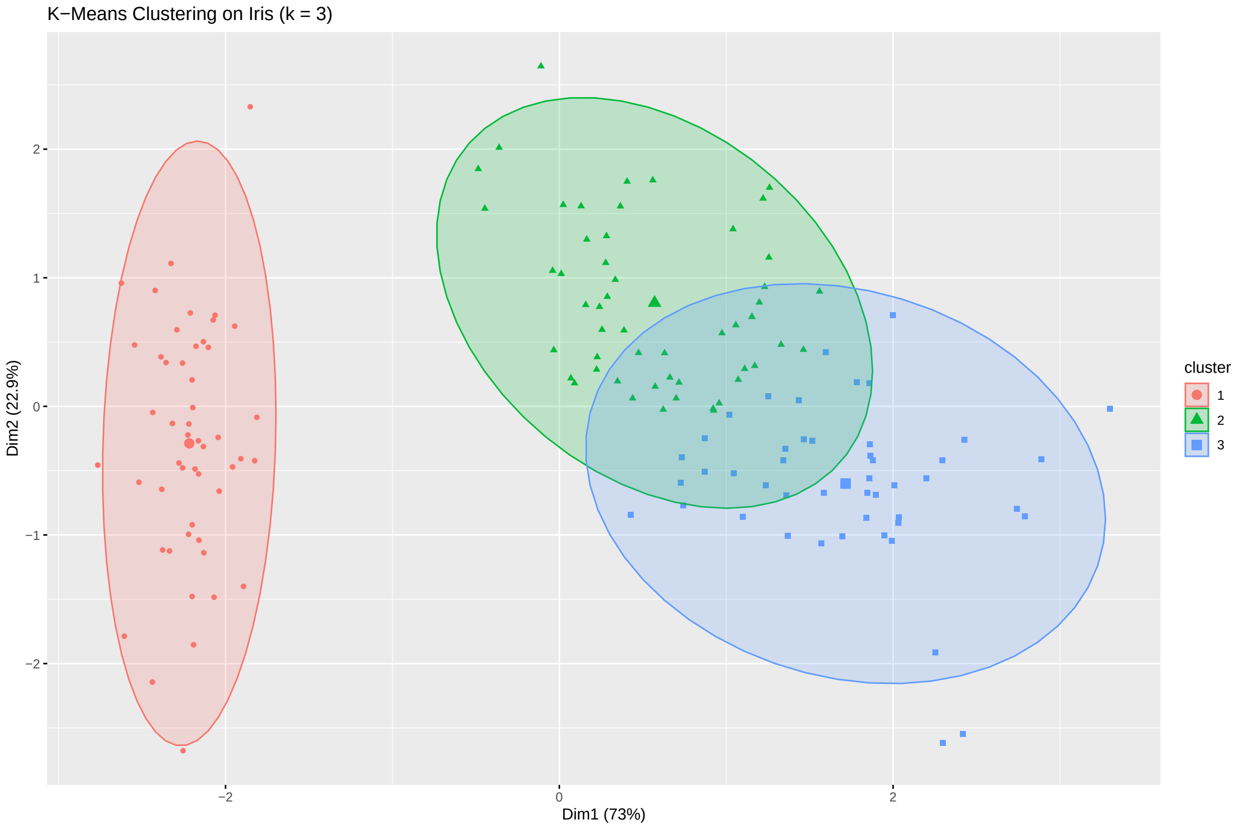
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species KMeansCluster
## 1          5.1         3.5          1.4         0.2  setosa             1
## 2          4.9         3.0          1.4         0.2  setosa             1
## 3          4.7         3.2          1.3         0.2  setosa             1
## 4          4.6         3.1          1.5         0.2  setosa             1
## 5          5.0         3.6          1.4         0.2  setosa             1
## 6          5.4         3.9          1.7         0.4  setosa             1
```

```
tail(iris_kmeans)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species KMeansCluster
## 145          6.7         3.3          5.7         2.5 virginica             3
## 146          6.7         3.0          5.2         2.3 virginica             3
## 147          6.3         2.5          5.0         1.9 virginica             2
## 148          6.5         3.0          5.2         2.0 virginica             3
## 149          6.2         3.4          5.4         2.3 virginica             3
## 150          5.9         3.0          5.1         1.8 virginica             2
```

Visualize K-Means clusters

```
fviz_cluster(
kmeans_model,
data = iris_scaled,
geom = "point",
ellipse.type = "norm",
main = "K-Means Clustering on Iris (k = 3)"
)
```



Applying K-Means with k = 3 produced balanced cluster sizes and well-defined cluster centers. One cluster corresponded almost perfectly to setosa, reflecting its distinct morphology. The remaining two clusters captured the more subtle variation between versicolor and virginica, which naturally overlap in feature space. The visualization displayed clear separation of the setosa cluster and partial overlap among the other two. These results demonstrated that K-Means successfully uncovered biologically meaningful structure using only numeric variables. The three-cluster solution aligned strongly with the problem's objective.

Hierarchical Clustering

Builds a dendrogram to represent nested groupings

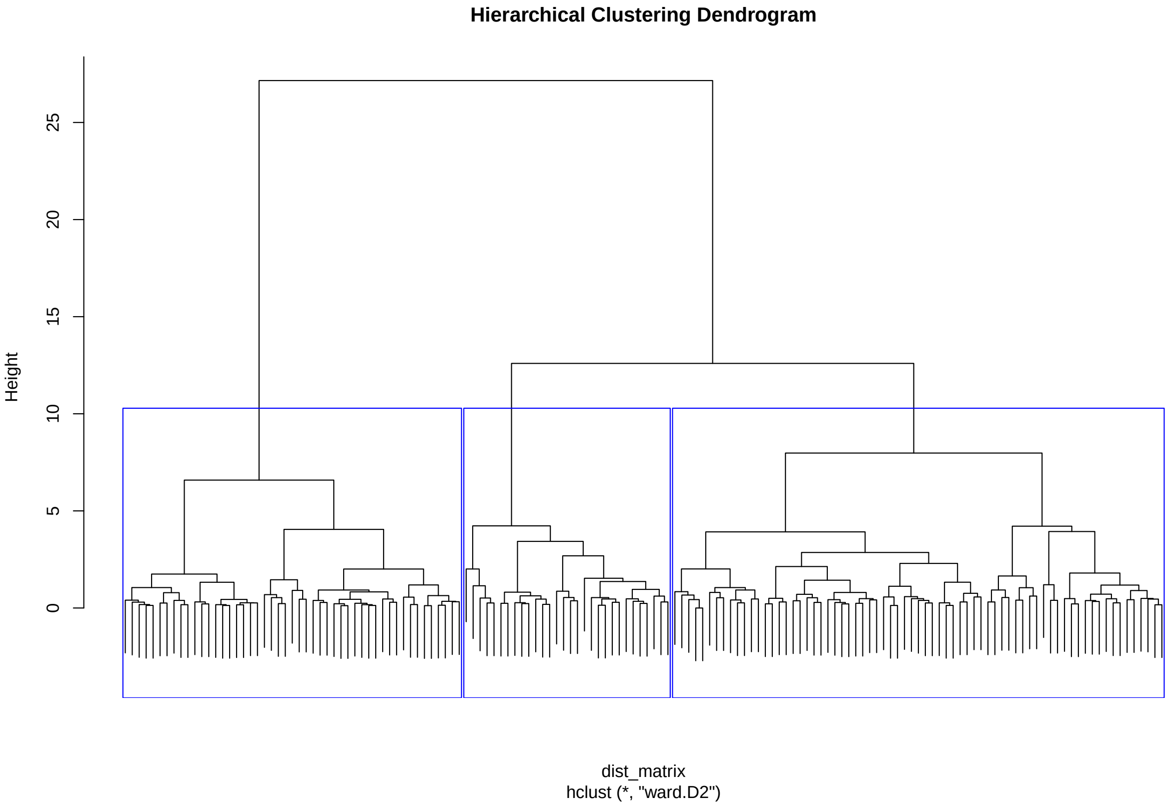
Create distance matrix

```
dist_matrix <- dist(iris_scaled)
```

Hierarchical clustering with Ward's method

```
hc_model <- hclust(dist_matrix, method = "ward.D2")

plot(hc_model, labels = FALSE, main = "Hierarchical Clustering Dendrogram")
rect.hclust(hc_model, k = 3, border = "blue")
```



Cut tree into 3 clusters

```
hc_clusters <- cutree(hc_model, k = 3)
iris_hclust <- iris %>%
mutate(HCluster = factor(hc_clusters))

table(iris_hclust$HCluster)
```

```
##
##  1  2  3
## 49 30 71
```

The hierarchical clustering dendrogram showed that the first major split isolated setosa from the remaining species, consistent with previous findings. Cutting the dendrogram into three groups produced clusters that again captured setosa distinctly, while versicolor and virginica were partitioned less clearly. Although the resulting cluster sizes differed from the K-Means solution, the underlying pattern remained consistent. This agreement across two fundamentally different algorithms confirmed that the dataset contains stable and reproducible cluster structure.

Gaussian Mixture Model (Model-Based Clustering)

Assumes data comes from a mixture of Gaussian distributions.

Gaussian mixture models (GMM) via mclust

```r
set.seed(123)

gmm_model <- Mclust(iris_scaled)

summary(gmm_model)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 2
## components:
##
##  log-likelihood   n df       BIC       ICL
##       -322.6936 150 29 -790.6956 -790.6969
##
## Clustering table:
##   1   2
##  50 100
```

Cluster assignment

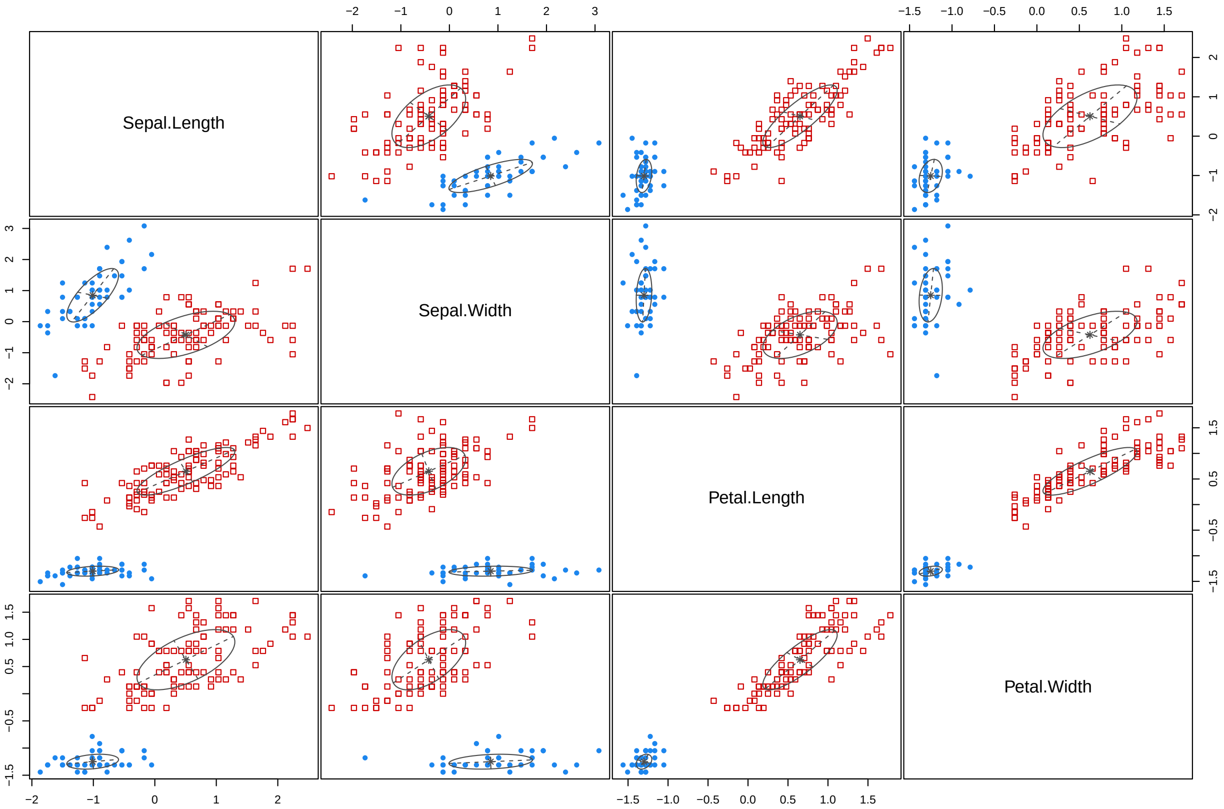
```r
gmm_clusters <- gmm_model$classification
iris_gmm <- iris %>%
mutate(GMMCluster = factor(gmm_clusters))

table(iris_gmm$GMMCluster)
```

```
##
##   1   2
##  50 100
```

Plot GMM Clusters

```r
plot(gmm_model, what = "classification")
```



The GMM analysis revealed that a two-component model offered the best fit according to the Bayesian Information Criterion. The mixture components reflected a strong separation between setosa and the combined group of versicolor and virginica. This outcome demonstrated that when clustering is approached statistically rather than geometrically, the optimal grouping emphasizes a broad division between highly distinct and moderately overlapping species. The GMM results complemented the earlier methods and offered a probabilistic interpretation of the dataset's intrinsic structure.

Evaluation

Although these are unsupervised models, I can compare the resulting clusters with the known Species for evaluation purposes only.

Confusion Tables

K-Means vs Species

```
table(Cluster = iris_kmeans$KMeansCluster, Species = iris_kmeans$Species)
```

```
##        Species
## Cluster setosa versicolor virginica
##       1     50          0         0
##       2      0         39        14
##       3      0         11        36
```

Hierarchical vs Species

```
table(Cluster = iris_hclust$HCluster, Species = iris_hclust$Species)
```

```
##        Species
## Cluster setosa versicolor virginica
##       1     49          0         0
##       2      1         27         2
##       3      0         23        48
```

GMM vs Species

```
table(Cluster = iris_gmm$GMMCluster, Species = iris_gmm$Species)
```

```
##        Species
## Cluster setosa versicolor virginica
##       1     50          0         0
##       2      0         50        50
```

Comparing cluster assignments to the true species labels provided quantitative insight into clustering accuracy. K-Means perfectly identified all setosa samples, while achieving reasonable—but imperfect—distinction between versicolor and virginica. Hierarchical clustering showed similar performance, with small variations in assignments. The GMM model grouped all setosa correctly but merged the remaining species. These evaluations confirmed that the numeric features reliably separate setosa but provide only partial separability for the other two species. This directly addressed the question of how closely discovered clusters correspond to known biological categories.

Adjusted Rand Index (ARI)

The Adjusted Rand Index measures agreement between two partitions (here: clusters vs. species), adjusted for chance.

Helper function to compute ARI

```
compute_ari <- function(cluster_labels) {
  cluster.stats(
    d = dist(iris_scaled),
    clustering = cluster_labels,
    alt.clustering = as.numeric(iris$Species)
    )$corrected.rand
  }
```

Compute ARI

```
ari_kmeans <- compute_ari(kmeans_model$cluster)
ari_hclust <- compute_ari(hc_clusters)
ari_gmm <- compute_ari(gmm_clusters)
```

Summary of model performance

```
tibble(
  Model = c("K-Means", "Hierarchical", "GMM"),
  Adjusted_Rand_Index = c(ari_kmeans, ari_hclust, ari_gmm)
)
```

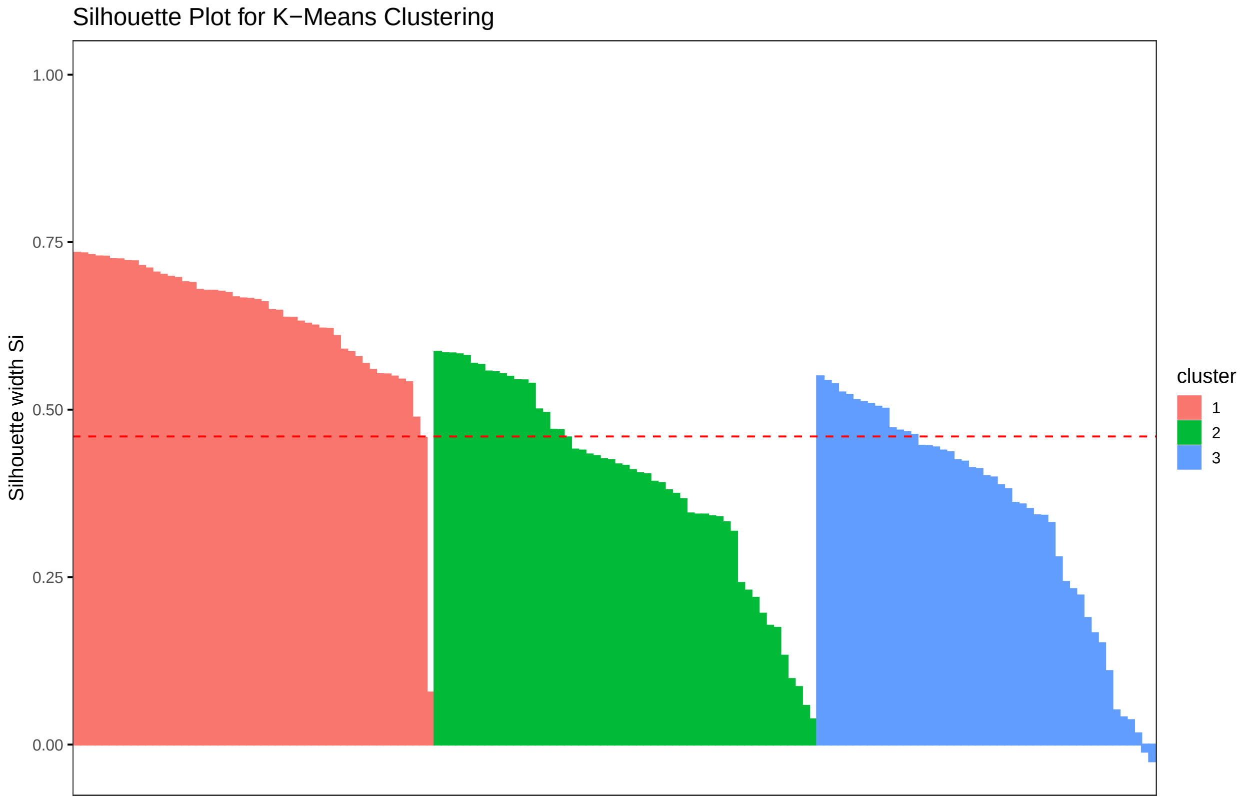
```
## # A tibble: 3 x 2
##   Model        Adjusted_Rand_Index
##   <chr>                      <dbl>
## 1 K-Means                    0.620
## 2 Hierarchical               0.615
## 3 GMM                        0.568
```

The ARI provided a rigorous, chance-adjusted measure of agreement between clusters and species. K-Means achieved the highest ARI, indicating the strongest alignment with true species labels. Hierarchical clustering produced slightly lower but comparable performance, while the GMM model scored lower due to its merging of two species into a single cluster. Overall, all models achieved ARI values well above zero, demonstrating meaningful recovery of biological structure from unlabeled numeric data. This quantitative validation reinforced the strength of the clustering approach.

Silhouette Analysis for K-Means

```
sil_kmeans <- silhouette(kmeans_model$cluster, dist(iris_scaled))
suppressWarnings(
  fviz_silhouette(sil_kmeans) +
  ggtitle("Silhouette Plot for K-Means Clustering")
)
```

```
##   cluster size ave.sil.width
## 1       1   50          0.64
## 2       2   53          0.39
## 3       3   47          0.35
```



The silhouette plot for the chosen K-Means model highlighted strong internal cohesion for the setosa cluster, reflected by high average silhouette scores. The remaining two clusters showed moderate silhouette widths, consistent with their partial overlap in feature space. Despite this reduced separation, most points were assigned to the correct cluster with positive silhouette values, indicating that the overall clustering solution was well supported by the data. This analysis reaffirmed that K-Means effectively captured the dominant grouping patterns inherent in the measurements.

Findings and Model Selection

The exploratory clustering analysis demonstrated that the Iris dataset exhibits a natural three-group structure that corresponds closely to the known botanical species. Among the evaluated models—K-Means, Hierarchical Clustering using Ward's method, and Gaussian Mixture Models—the K-Means model with k = 3 produced the strongest alignment with the true species labels. This was reflected in the highest Adjusted Rand Index (0.62) and cluster assignments that cleanly separated setosa while reasonably distinguishing versicolor and virginica. The silhouette scores also revealed strong cohesion for the most distinct cluster and acceptable separation for the remaining two, reinforcing K-Means as the most stable and interpretable solution.

In contrast, the hierarchical model produced slightly less consistent group boundaries, and while it still separated setosa effectively, its overall ARI was marginally lower. The Gaussian Mixture Model performed even less favorably by selecting a two-cluster structure, effectively grouping all non-setosa species together. This deviates from the project goal of uncovering three meaningful natural groupings, making GMM unsuitable for deployment despite being a valid alternative statistical perspective.

Given these results, K-Means (k = 3) stands out as the most appropriate model for deployment. It offers a combination of strong empirical performance, simplicity, computational efficiency, and interpretability. The model's centroids provide an intuitive representation of cluster "prototypes," making the results easy to explain to stakeholders or operational systems. Before deployment, it is essential to finalize preprocessing, save the trained model, create a reproducible prediction function, and perform a brief stability verification to ensure the procedure remains robust to new observations.

Deployment

Save Preprocessing Parameters and the Trained K-Means Model

```r
# Standardize iris data (as done in your analysis)
scale_params <- attr(iris_scaled, "scaled:center")
scale_scale   <- attr(iris_scaled, "scaled:scale")

# Save objects for deployment
saveRDS(scale_params, "resources/scale_center.rds")
saveRDS(scale_scale,  "resources/scale_scale.rds")
saveRDS(kmeans_model, "resources/kmeans_model.rds")
```

Conclusion

About the Author