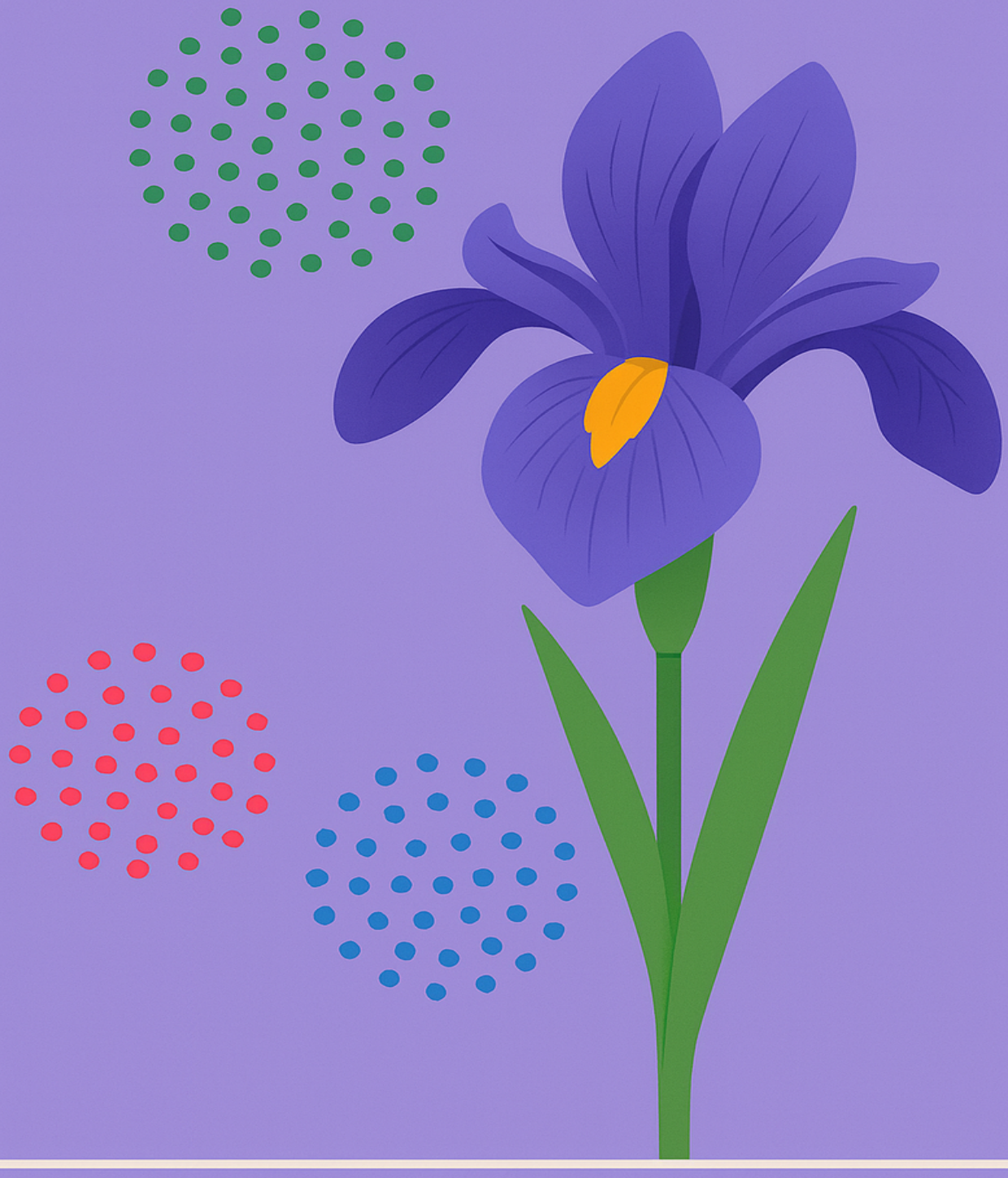
Iris Clusters Analysis



Iris Clusters Analysis

Seif H. Kungulio

December 03, 2025

Contents

Business Understanding

Introduction

Problem Statement

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
```

Statistical Summary

Display the statistical summary of the dataset

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
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	

Data Description

Data Dictionary

Check for Missing Values

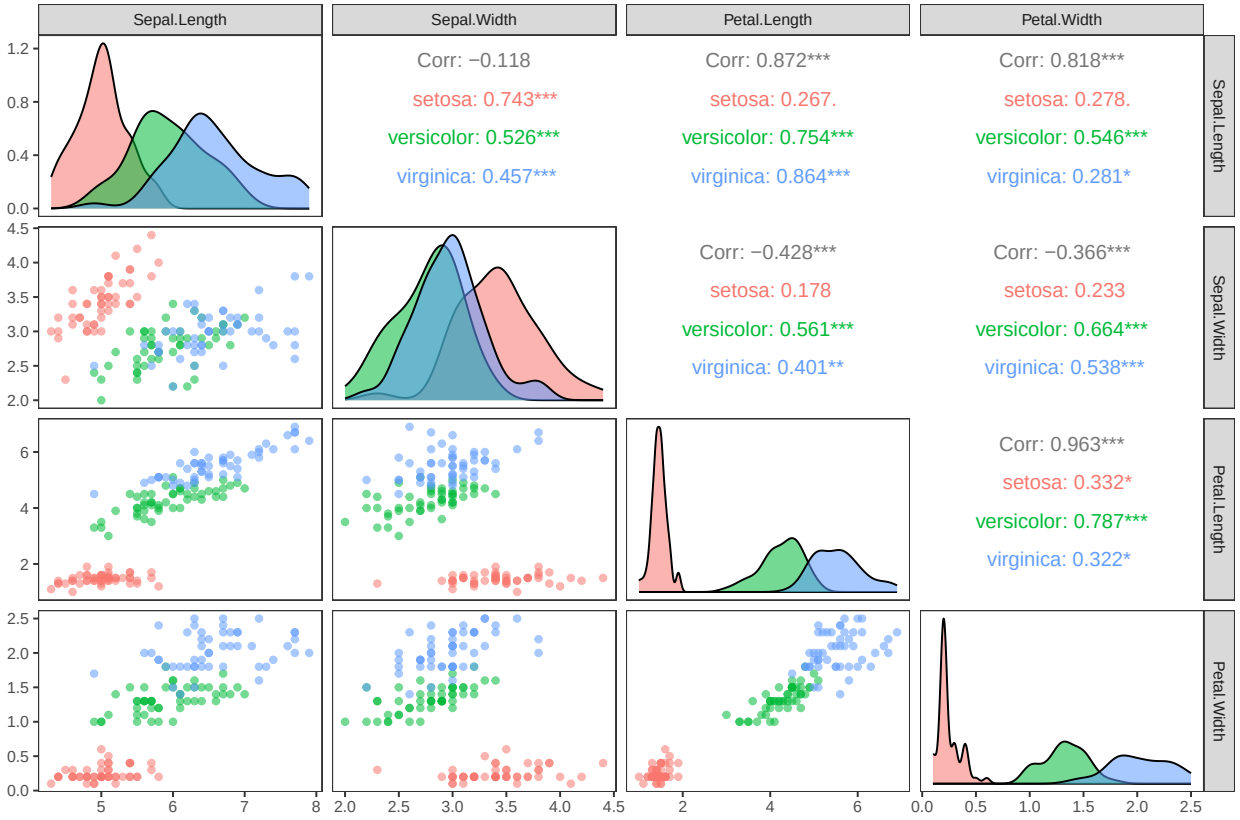
```
colSums(is.na(iris))
```

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

Exploratory Data Analysis (Plots)

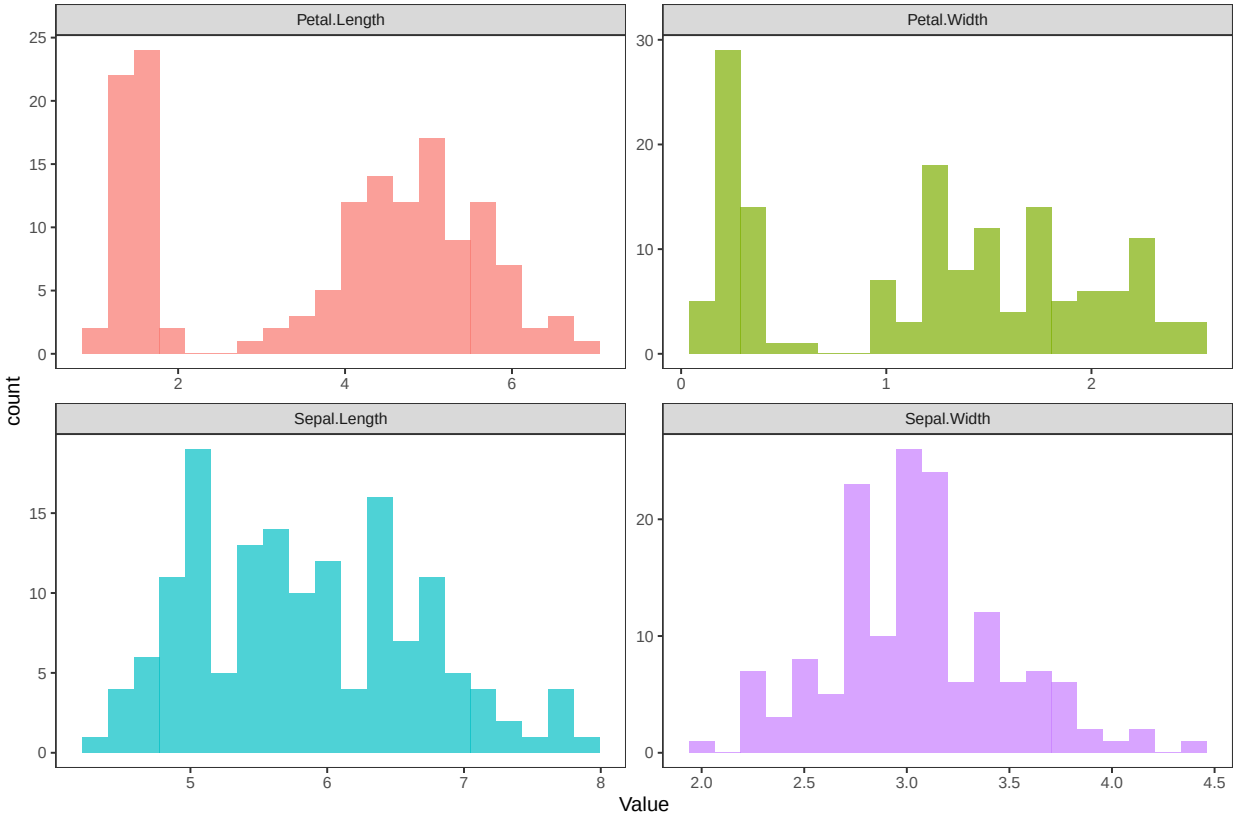
Pairwise relationship

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



Univariate distribution

```
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")
```


Petal.Length
Petal.Width
Sepal.Length
Sepal.Width
count
Value

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)
```

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
```

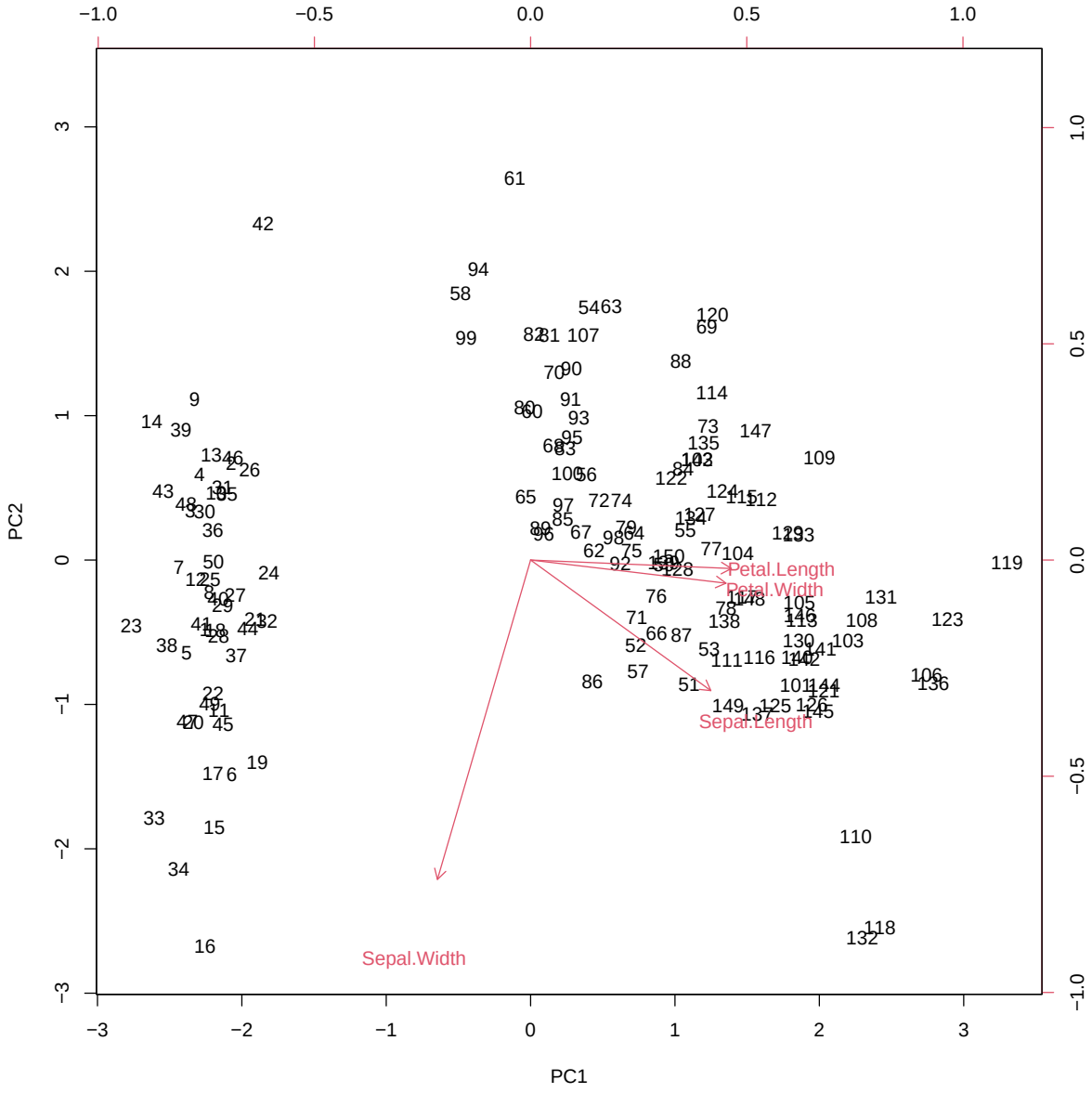
PCA for Dimensionality Reduction & Visualization

```
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

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


PC1
PC2
Petal.Length
Petal.Width
Sepal.Length
Sepal.Width

Modeling

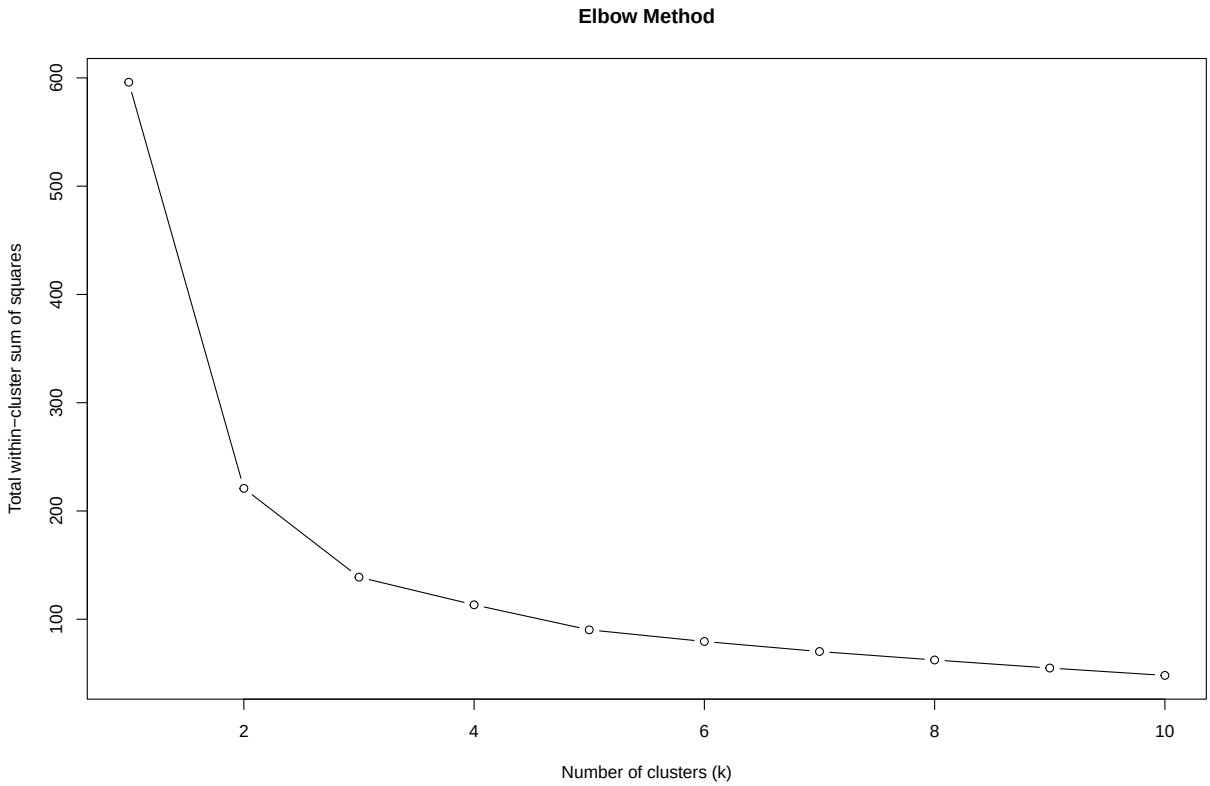
I will explore several unsupervised methods:

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

Determine number of clusters (k) for K-Means

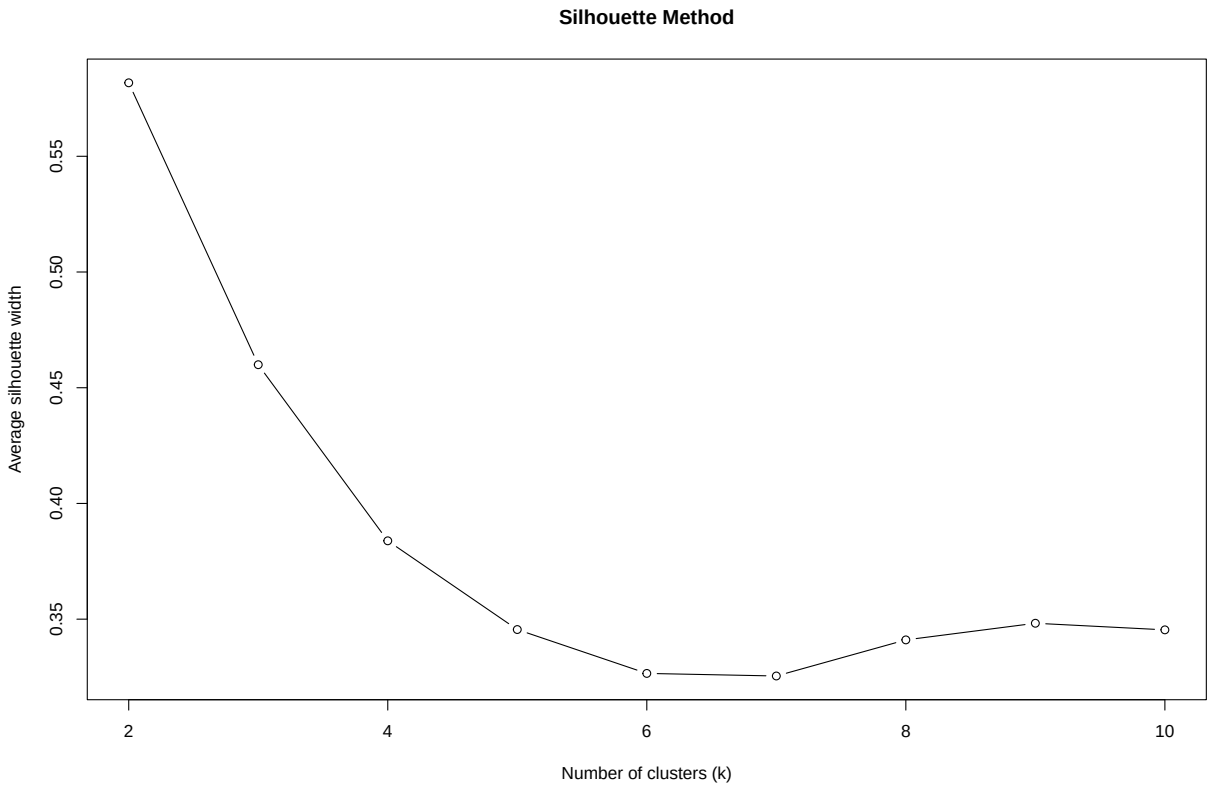
Elbow Method

```
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"
)
```



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"
)
```



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/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)
```

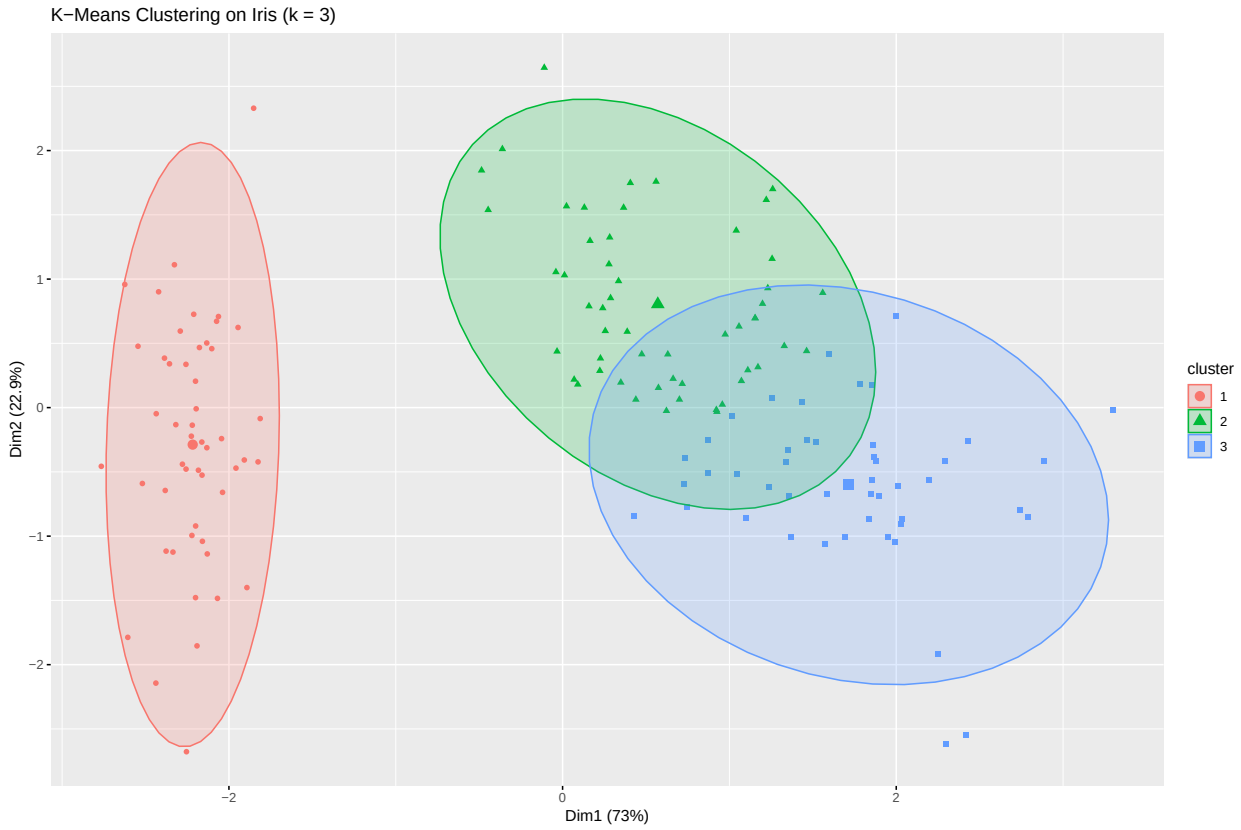
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species KMeansCluster
## 1          5.1         3.5          1.4         0.2  setosa             1
## 2          4.9         3.0          1.4         0.2  setosa             1
## 3          4.7         3.2          1.3         0.2  setosa             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)"
)
```



Hierarchical Clustering

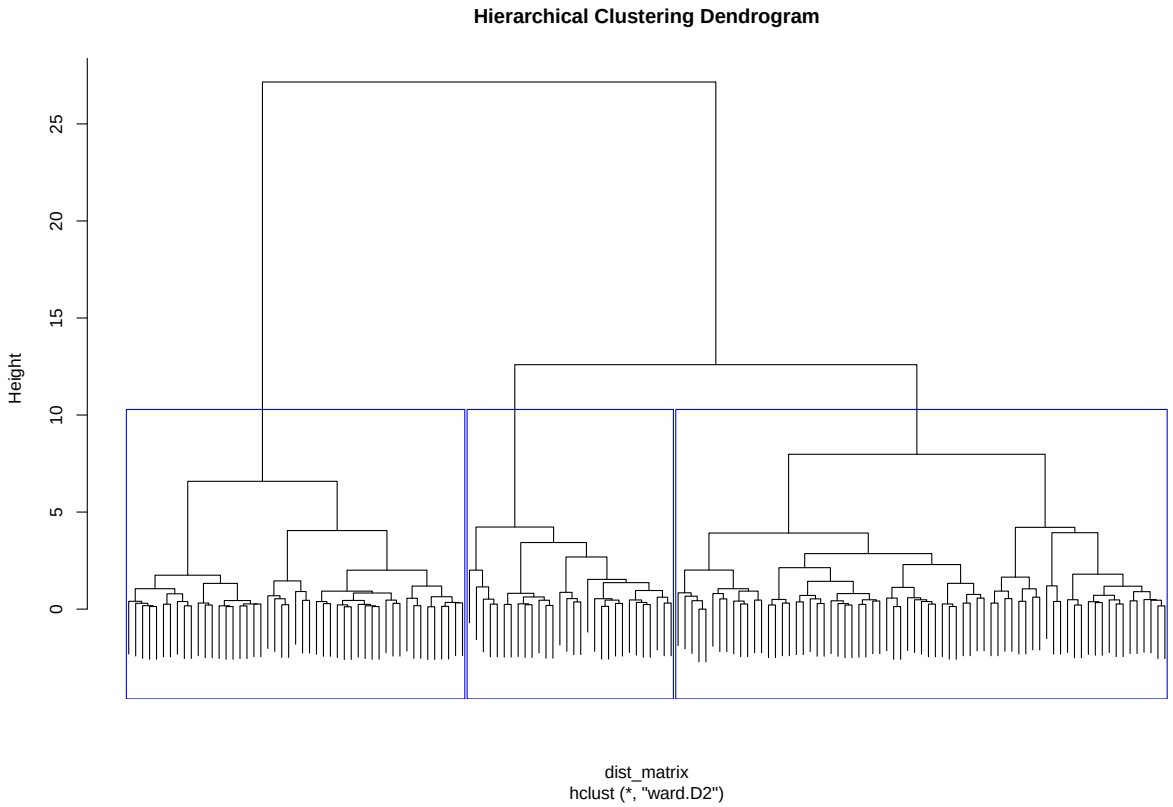
Builds a dendrogram to represent nested groupings

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
```

Gaussian Mixture Model (Model-Based Clustering)

Assumes data comes from a mixture of Gaussian distributions.

Gaussian mixture models (GMM) via mclust

```
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
```

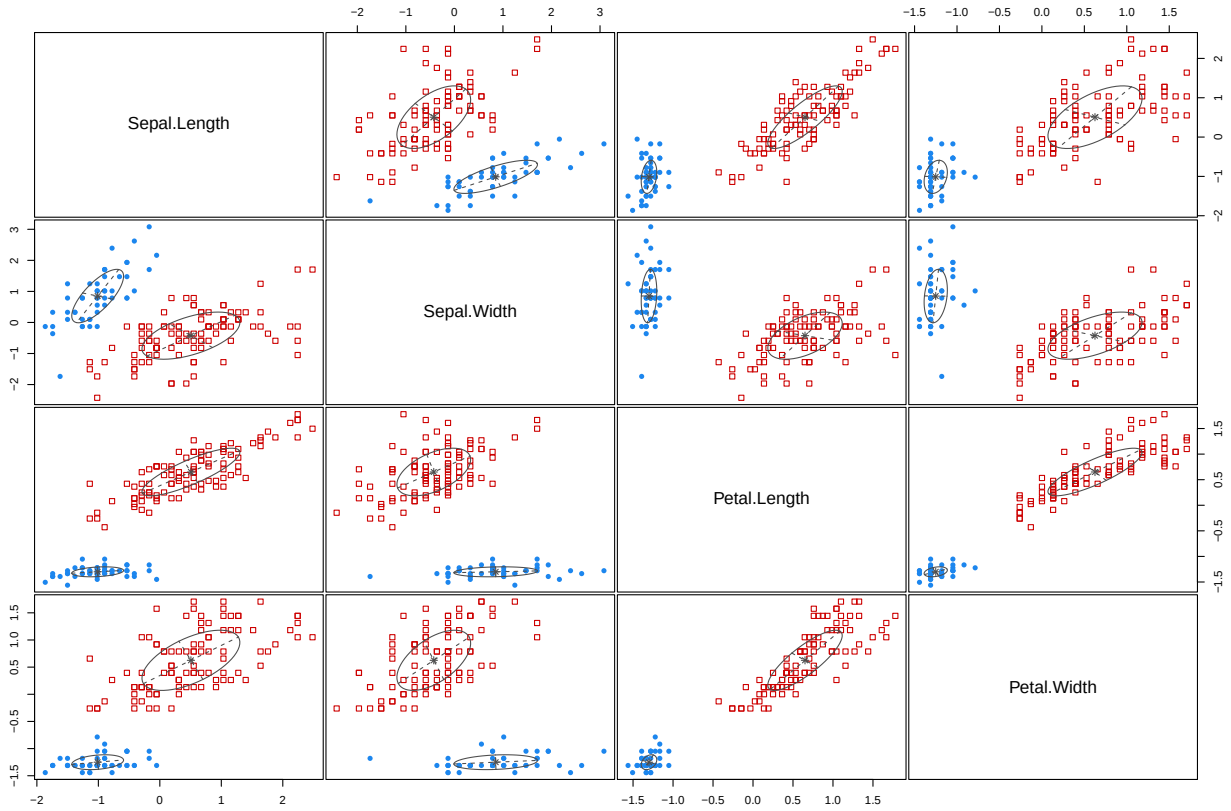
Cluister assignment

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

table(iris_gmm$GMMCluster)
```

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

Plot GMM Clusters

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



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
```

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)
```

Display models ARI

```
ari_kmeans
```

```
## [1] 0.6201352
```

```
ari_hclust
```

```
## [1] 0.615323
```

```
ari_gmm
```

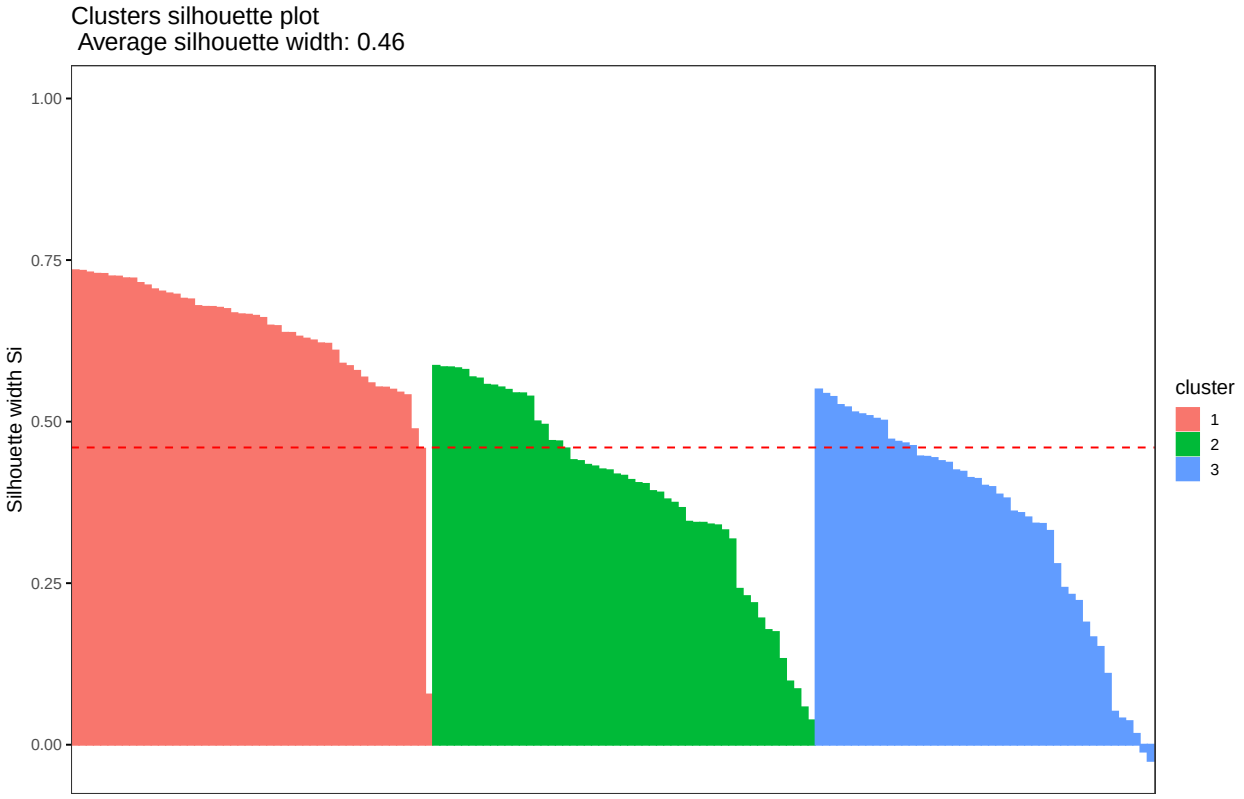
```
## [1] 0.5681159
```

Silhouette Analysis for K-Means

```
sil_kmeans <- silhouette(kmeans_model$cluster, dist(iris_scaled))
fviz_silhouette(sil_kmeans)
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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


Clusters silhouette plot
Average silhouette width: 0.46
Silhouette width Si
1.00
0.75
0.50
0.25
0.00
cluster
1
2
3

Deployment

Conclusion

About the Author