K-Means Clustering of Iris Data

Michaella

R Markdown

K-Means Clustering

K-means clustering is an unsupervised machine learning technique used for partitioning a dataset into distinct groups or clusters based on similarities among data points, with the objective of minimizing the variance within each cluster. It assigns each data point to the cluster whose centroid (representative point) is closest, commonly employed for data segmentation and pattern discovery.

This Iris Data Exploration and Clustering project features the following steps:

- Step 1: Load necessary libraries (ggplot2, GGally, ggfortify) and import the Iris dataset.
- Step 2: Visualize data relationships and distributions.
- Step 3: Perform k-means clustering on both unscaled and scaled data.
- Step 4: Determine optimal cluster numbers with elbow plots.
- Step 5: Compare clustering results and visualize clusters.

The “iris.txt” dataset comprises 150 data points, each characterized by four predictor variables related to the dimensions of sepals and petals of flowers. Additionally, there is one categorical response variable indicating the flower type. This dataset is accessible via the R library “datasets” using the “iris” command once the library is loaded. Alternatively, it can be obtained from the UCI Machine Learning Repository at this link: https://archive.ics.uci.edu/ml/datasets/Iris .

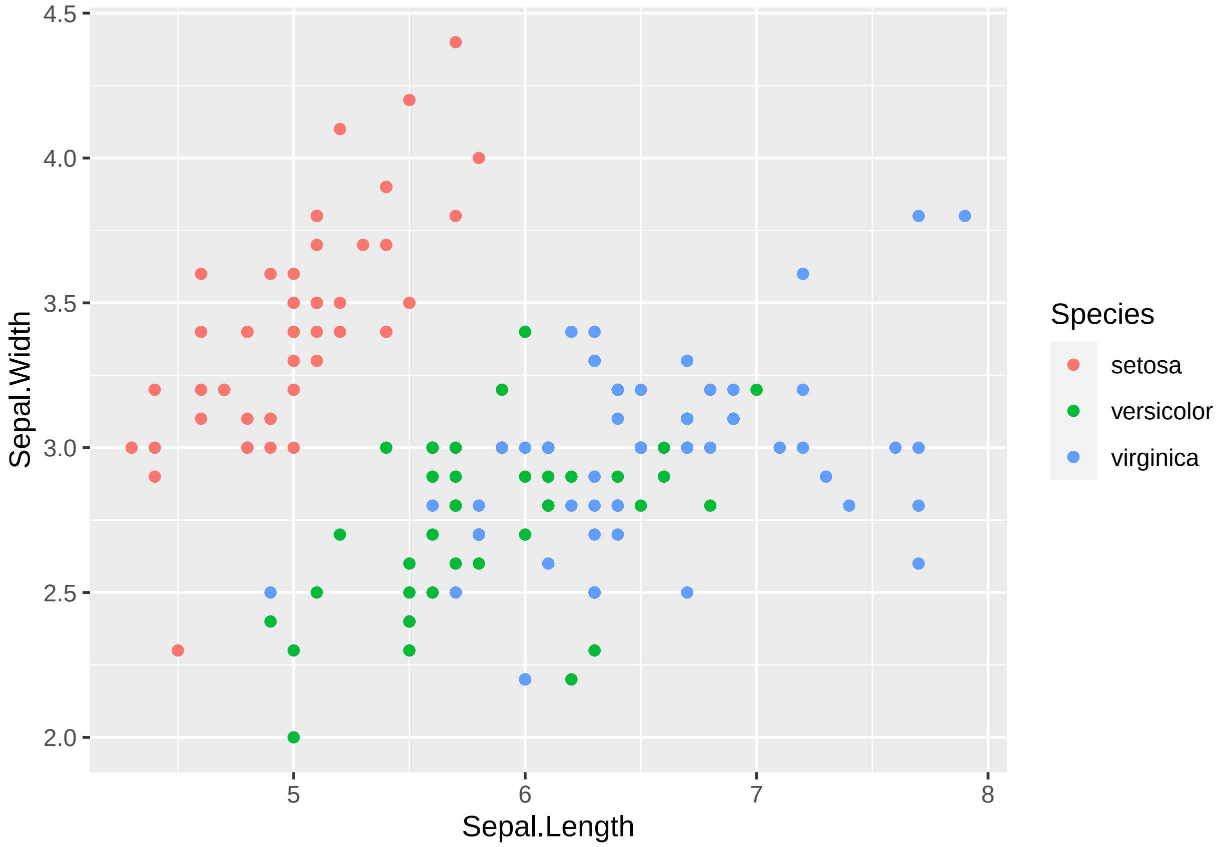
```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggfortify)

iris_data <- read.table("/Users/michaella/Downloads/iris/iris.txt", stringsAsFactors = FALSE, header =

head(iris_data)
```

```
##   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
```

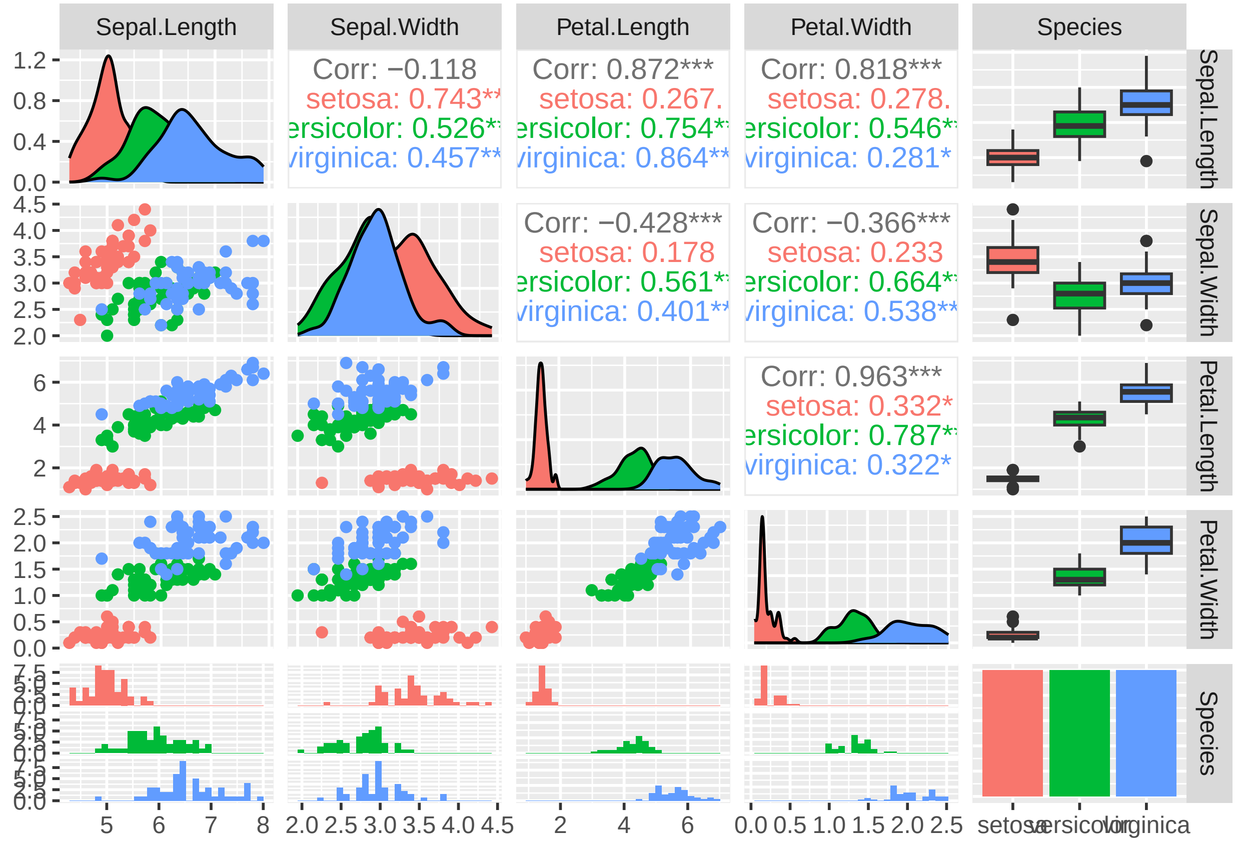
```
ggplot(iris_data, aes(Sepal.Length, Sepal.Width, color=Species)) + geom_point()
```



```
ggpairs(iris_data,
        ggplot2:: aes(colour=Species))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
#kmeans() on UNSCALED data
cluster_1 <- kmeans(iris_data[,1:4], centers=2, nstart =5)
cluster_2 <- kmeans(iris_data[, 1:4], centers =3, nstart= 10)

# Total within sum of squares to make elbow diagram
cluster_1$tot.withinss
```

```
## [1] 152.348
```

```
cluster_2$tot.withinss
```

```
## [1] 78.85144
```

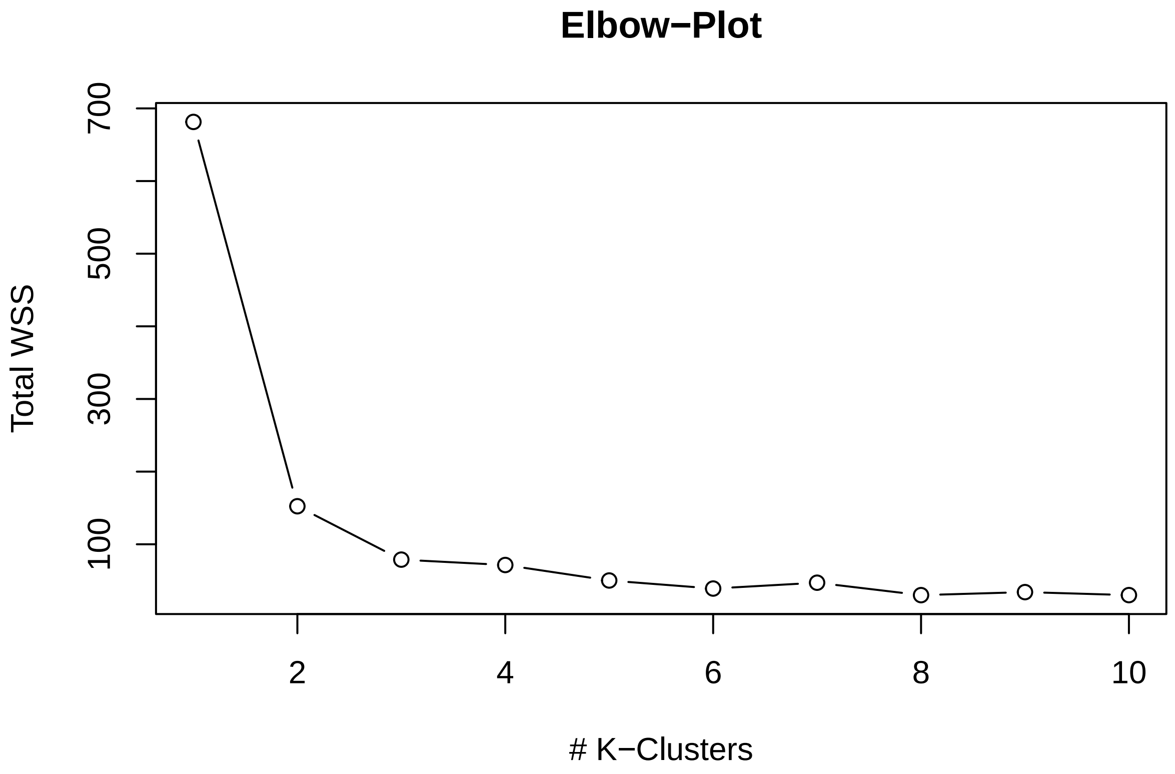
```
# Table to compare clusters to True Labels
table(cluster_1$cluster, iris_data$Species)
```

```
##
##     setosa versicolor virginica
##   1     50          3         0
##   2      0         47        50
```

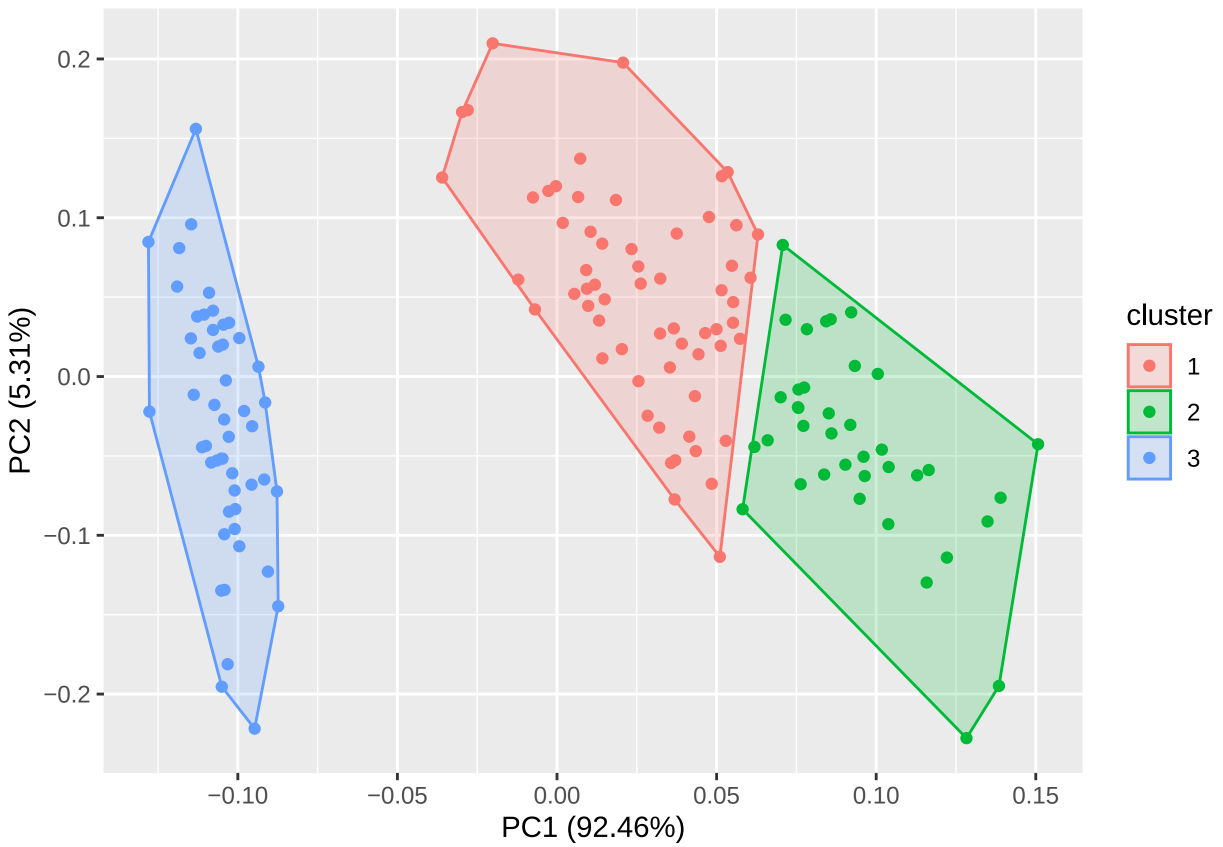
```
table(cluster_2$cluster, iris_data$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         48        14
##   2      0          2        36
##   3     50          0         0
```

```
#######################################
#           UNSCALED                  #
#######################################

# Vector to store within-cluster sum of squares
wss <- vector()

# Make a suggested value for k
iris_k <- 1:10

#K-means Clustering
for (k in iris_k) {
  iris_kmeans <- kmeans(iris_data[, 1:4], centers= k)
  wss[k] <- iris_kmeans$tot.withinss

}

# Create elbow plot
plot(
  iris_k,
  type="b",
  wss,
  xlab = "# K-Clusters",
  ylab = "Total WSS",
  main = "Elbow-Plot"

)
```



```
autoplot(cluster_2, iris_data, frame=TRUE)
```



Based on the Elbow-plot, k=3 is likely a good cluster value.

The first line standardizes the Iris dataset by scaling the first four columns using the scale() function. Standardization is essential to bring all variables to a similar scale, which is particularly important for k-means clustering. Out of curiosity, I tried it with Scaled Data but with poor results. The scaled data particularly caused the model to struggle identifying between versicolor and virginica.

```
########################
#  SCALED DATA             #
#######################

# Standardize the data
scaled_iris_data <- as.data.frame(scale(iris_data[, 1:4]))

#kmeans() on SCALED data
scaled_cluster_1 <- kmeans(scaled_iris_data[,1:4], centers=2, nstart =5)
scaled_cluster_2 <- kmeans(scaled_iris_data[, 1:4], centers =3, nstart= 10)

# Total within sum of squares to make elbow diagram
scaled_cluster_1$tot.withinss
```

```
## [1] 220.8793
```

```
scaled_cluster_2$tot.withinss
```

```
## [1] 138.8884
```

Our values are higher compared to the the unscaled data. A higher WSS value suggests that the clusters may not be well-defined or that the data points within each cluster are widely scattered. It's often desirable to find an optimal number of clusters (k) that minimizes the WSS while maintaining reasonably compact and cohesive clusters, which typically indicates a better partitioning of the data.

This is why we will make an elbow plot to select an appropriate k by looking for a point in the WSS plot where the reduction in WSS starts to slow down significantly (i.e. where the elbow kink appears).

On the following, I have removed the first cluster (k = 2) because I have established that k=3 is a better cluster value.

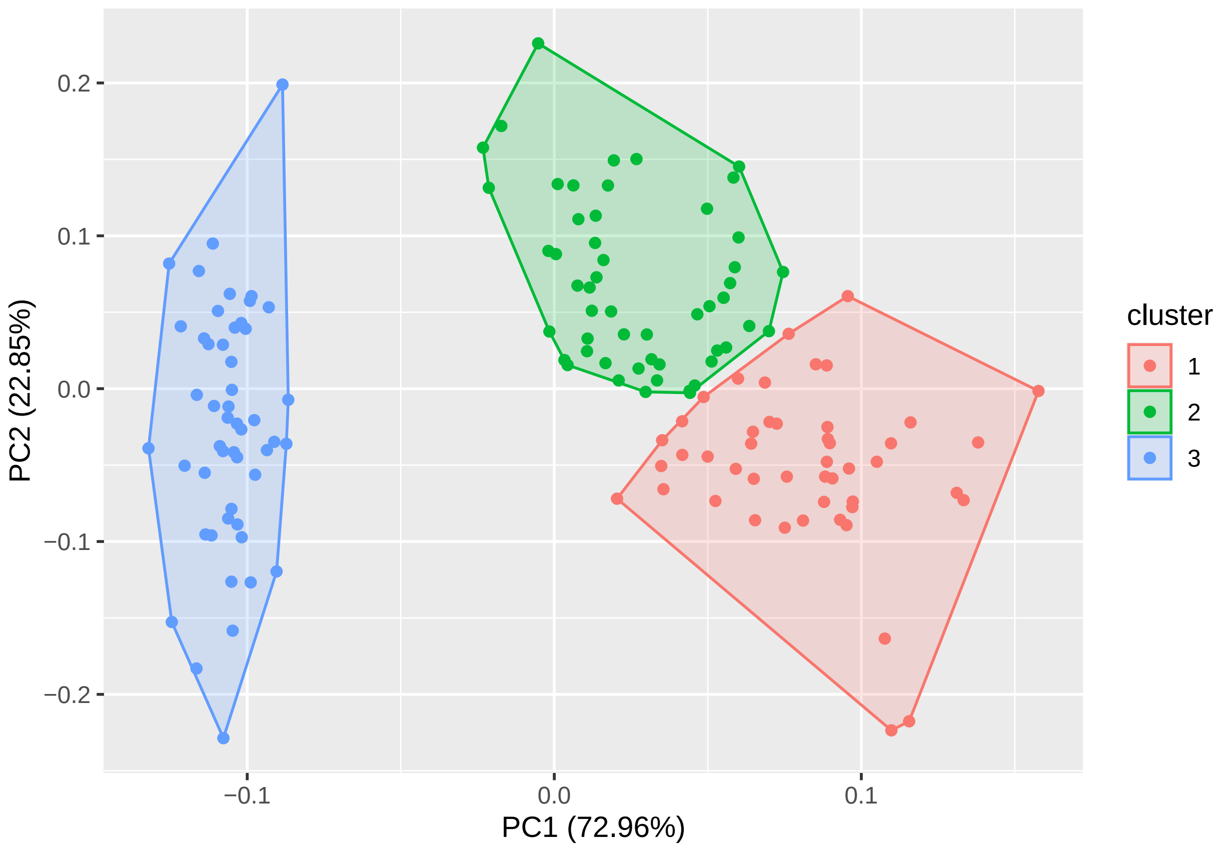
```
# Table to compare scaled and unscaled
table(scaled_cluster_2$cluster, iris_data$Species)
```

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

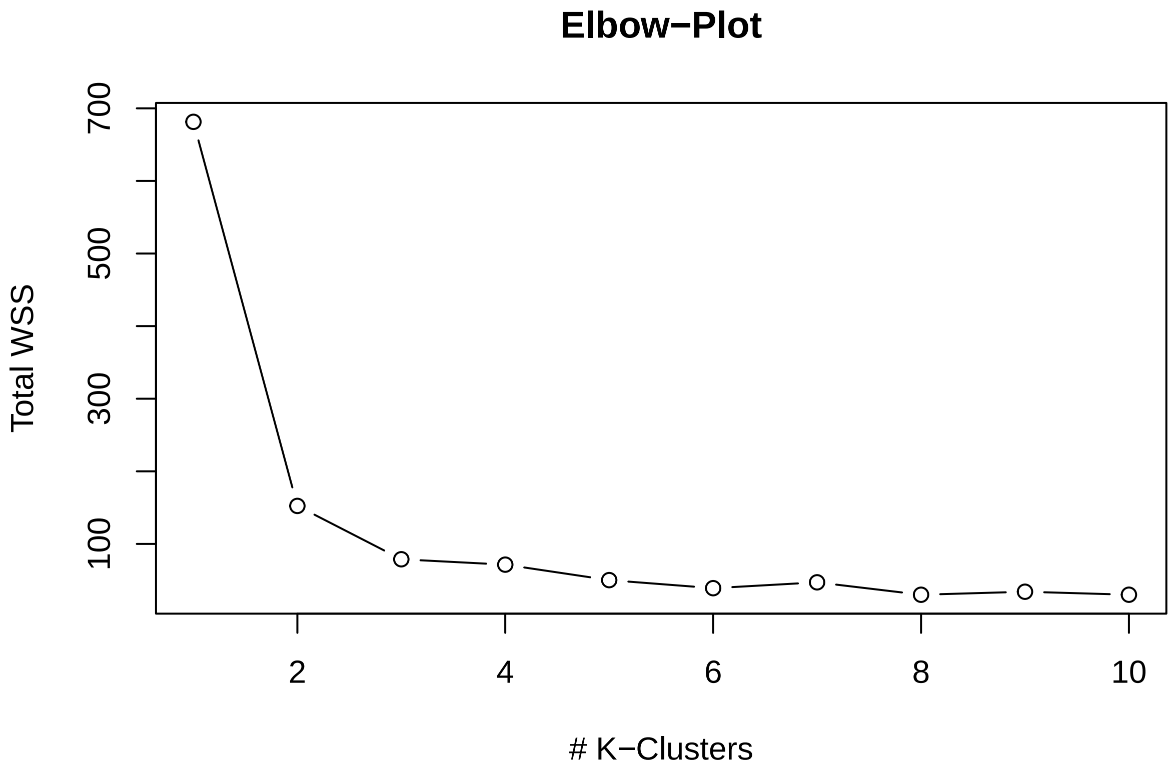
```
table(cluster_2$cluster, iris_data$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         48        14
##   2      0          2        36
##   3     50          0         0
```

```
# Vector to store within-cluster sum of squares
scaled_wss <- vector()

# Make a suggested value for k
iris_k <- 1:10

#K-means Clustering
for (k in iris_k) {
  scaled_iris_kmeans <- kmeans(scaled_iris_data[, 1:4], centers= k)
  scaled_wss[k] <- scaled_iris_kmeans$tot.withinss

}
autoplot(scaled_cluster_2, scaled_iris_data, frame=TRUE)
```



```
# Create elbow plot
plot(
  iris_k,
  type="b",
  wss,
  xlab = "# K-Clusters",
  ylab = "Total WSS",
  main = "Elbow-Plot"

)
```



The elbow plot produced the same output, but the cluster plot (autoplot) clusters are different. It also confirms that k = 3 was the ideal value for this cluster.