K_MEANS CLUSTERING USING IRIS DATASET

Langat Erick

To perform K-means clustering on the iris dataset in R, you would first need to load the dataset into R. The iris dataset is built-in to R, so you can load it using the following code:

```
library(tidyverse)
data(iris)#load the data
```

Next, you would need to select the variables that you want to use for the clustering. In this case, we would use the petal length and width variables. You can create a new data frame containing just those variables with the following code:

```
iris_data <- iris[, c("Petal.Length", "Petal.Width")]
iris_data %>% head()
```

```
##   Petal.Length Petal.Width
## 1          1.4         0.2
## 2          1.4         0.2
## 3          1.3         0.2
## 4          1.5         0.2
## 5          1.4         0.2
## 6          1.7         0.4
```

Once you have your data frame, you can run the K-means clustering algorithm using the **`kmeans()`** function in R. You'll need to specify the number of clusters you want to create. Since we know there are three species of iris in the dataset, we'll set **`k = 3`**:

```
set.seed(123)  # set random seed for reproducibility
kmeans_model <- kmeans(iris_data, centers = 3, nstart = 10)
kmeans_model
```

```
## K-means clustering with 3 clusters of sizes 52, 48, 50
##
## Cluster means:
##   Petal.Length Petal.Width
## 1     4.269231    1.342308
## 2     5.595833    2.037500
## 3     1.462000    0.246000
##
## Clustering vector:
##   [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 3
##  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2
## [112] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [149] 2 2
##
## Within cluster sum of squares by cluster:
## [1] 13.05769 16.29167  2.02200
```

```
##  (between_SS / total_SS =  94.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

The **kmeans()** function takes the dataset to be clustered (**iris_data** in this case), the number of clusters (**centers = 3**), and the number of random starting configurations to use (**nstart = 10**) as input. The **set.seed()** function is used to ensure that the results are reproducible across different runs.

Once the clustering has been performed, we can view some basic information about the resulting clusters using the **summary()** function:

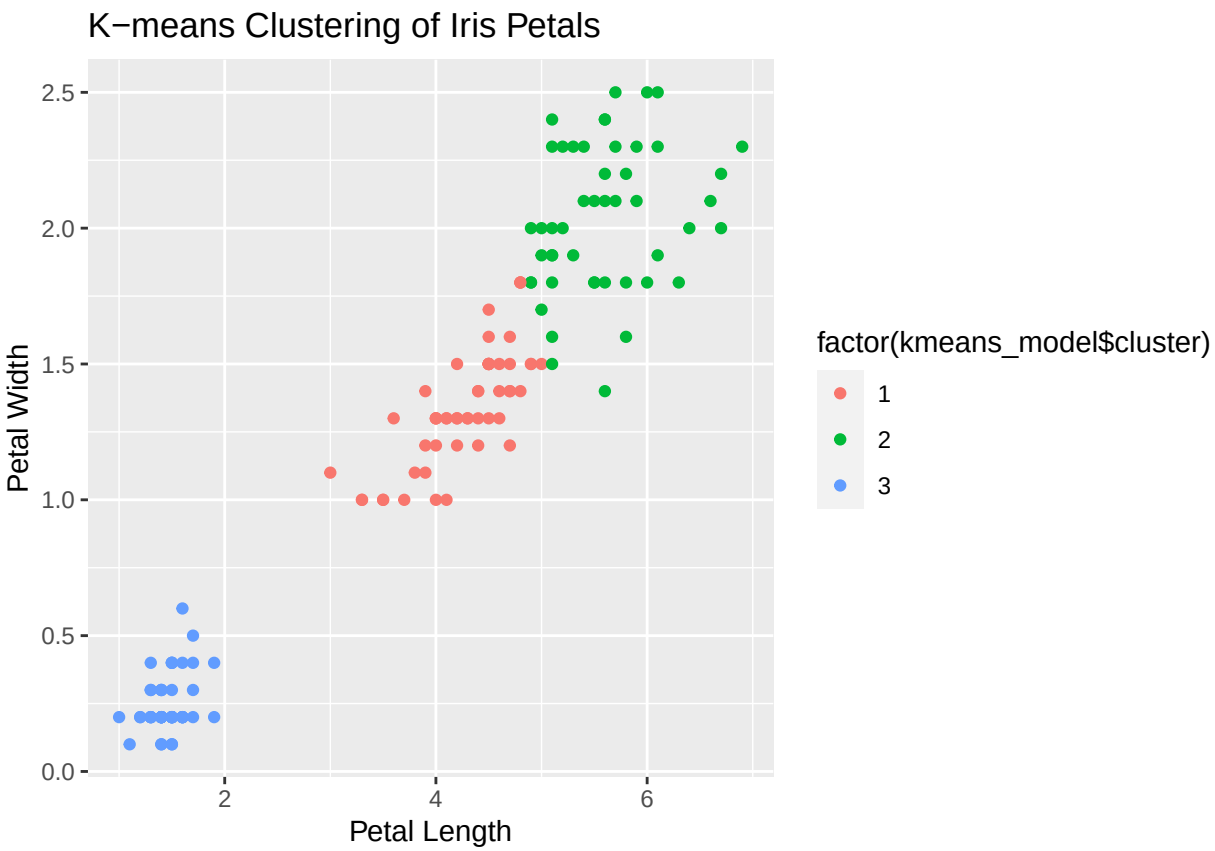
```
summary(kmeans_model)
```

```
##              Length Class  Mode
## cluster      150    -none- numeric
## centers        6    -none- numeric
## totss          1    -none- numeric
## withinss       3    -none- numeric
## tot.withinss   1    -none- numeric
## betweenss      1    -none- numeric
## size           3    -none- numeric
## iter           1    -none- numeric
## ifault         1    -none- numeric
```

This will display information about the cluster centers, the number of observations in each cluster, and the within-cluster sum of squares (which measures the degree of similarity between the data points in each cluster).

Finally, we can visualize the clusters using a scatterplot. Here's some example code to create a scatterplot with points colored according to their assigned cluster:

```
library(ggplot2)
ggplot(data = iris_data, aes(x = Petal.Length, y = Petal.Width, color = factor(kmeans_model$cluster))) +
  geom_point() +
  labs(title = "K-means Clustering of Iris Petals", x = "Petal Length", y = "Petal Width")
```



This code creates a scatterplot using the `ggplot2` package, with the x-axis representing petal length, the y-axis representing petal width, and points colored according to their assigned cluster. The `labs()` function is used to add a title and axis labels to the plot.

Here's a simplified explanation of the K-means clustering results for the iris dataset in R:

We performed K-means clustering on the iris dataset using two variables, petal length and petal width. We chose to divide the data into 3 clusters, which corresponds to the three different species of iris in the dataset. The algorithm grouped the flowers into three clusters based on their petal length and width measurements. The summary function provided us with information about the cluster centers, the number of observations in each cluster, and how similar the data points were within each cluster.

The scatterplot shows the clustering results graphically, with each point representing a flower and colored according to its assigned cluster. The plot shows that the algorithm was able to separate the flowers into three distinct groups, which correspond to the three different iris species in the dataset.

Overall, the K-means clustering results indicate that the petal length and width measurements are useful in distinguishing between the different species of iris in the dataset.