

Practical No. 10

Implementation and analysis of clustering algorithms like K-Means , Agglomerative.

K-Means Clustering:

Step 1: Install Required Packages

R

Copy code

```
install.packages("cluster")  
library(cluster)
```

Step 2: Generate Sample Data

R

Copy code

```
set.seed(123)  
data <- matrix(rnorm(300, mean = c(2, 8, 15), sd = c(3, 3, 3)), ncol = 3)
```

Step 3: Run K-Means Algorithm

R

Copy code

```
# Assuming you want to create 3 clusters  
kmeans_result <- kmeans(data, centers = 3)
```

```
# Display cluster assignments
```

```
kmeans_result$cluster
```

Step 4: Visualize Results

R

Copy code

```
# Visualize clusters  
plot(data, col = kmeans_result$cluster, main = "K-Means Clustering")  
points(kmeans_result$centers, col = 1:3, pch = 8, cex = 2)
```

Agglomerative Hierarchical Clustering:

Step 1: Install Required Packages

R

Copy code

```
install.packages("stats")
```

Step 2: Run Agglomerative Hierarchical Clustering Algorithm

R

Copy code

```
# Assuming you have the data matrix from the previous example  
dist_matrix <- dist(data)  
hierarchical_result <- hclust(dist_matrix, method = "ward.D2")
```

```
# Cut the dendrogram to create clusters
```

```
cut_tree_result <- cutree(hierarchical_result, k = 3)
```

```
# Display cluster assignments
```

```
cut_tree_result
```

Step 3: Visualize Results

R

Copy code

```
# Visualize dendrogram
```

```
plot(hierarchical_result, main = "Agglomerative Hierarchical Clustering  
Dendrogram", xlab = "", sub = "")
```

```
rect.hclust(hierarchical_result, k = 3, border = 2:4)
```

These are basic examples, and you can adapt them to your specific dataset and requirements. For a real-world scenario, you would typically load your dataset, preprocess it as needed, and then apply the clustering algorithm. Additionally, you may want to evaluate the clustering results using relevant metrics depending on your specific use case.

Always remember to tailor the code to your specific needs and explore the parameters of the clustering algorithms based on the characteristics of your data.

```
> library(stats)
> library(dplyr)
> library(ggplot2)
> library(ggfortify)
>
> View(iris)
> mydata <- select(iris, c(1, 2, 3, 4))
> View(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
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
14	4.3	3.0	1.1	0.1	setosa
9	4.4	2.9	1.4	0.2	setosa
39	4.4	3.0	1.3	0.2	setosa
43	4.4	3.2	1.3	0.2	setosa
42	4.5	2.3	1.3	0.3	setosa
4	4.6	3.1	1.5	0.2	setosa
7	4.6	3.4	1.4	0.3	setosa
23	4.6	3.6	1.0	0.2	setosa
48	4.6	3.2	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa

```

> wssplot <- function(data, nc=15, seed=1234){
+   wss <- (nrow(data)-1)*sum(apply(data,2,var))
+   for (i in 2:nc){
+     set.seed(seed)
+     wss[i] <- sum(kmeans(data, centers=i)$withinss)}
+   plot(1:nc, wss, type="b", xlab="Number of Clusters",
+        ylab="Within groups sum of squares")
+   wss
+ }
>
> wssplot(mydata)
[1] 681.37060 152.34795 78.85144 57.26562 46.46117 41.70442
40.66047 39.03110 34.20191 33.40363
[11] 32.53526 30.71360 29.34359 27.01944 19.61325
>

```

```

> KM = kmeans(mydata,3)
>
> autoplot(KM,mydata,frame=TRUE)
>
> KM$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1      6.850000      3.073684      5.742105      2.071053
2      5.901613      2.748387      4.393548      1.433871
3      5.006000      3.428000      1.462000      0.246000
> View(iris)
> |

```