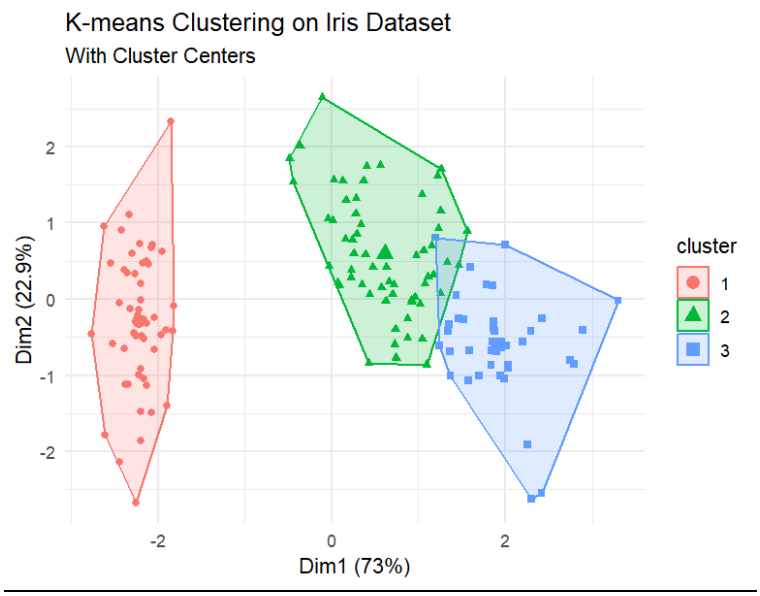
EDA Experiment – 9

Digital Assignment / Mid term assignment



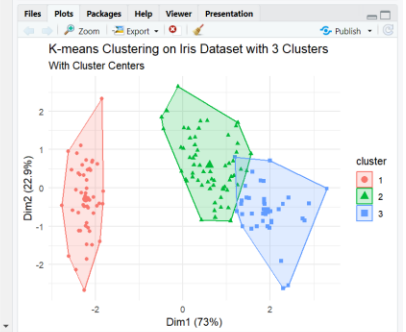
Name : Jvn Ganesh

Roll No : 21BDS0085

Screenshots

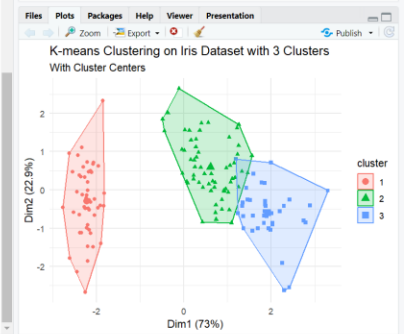
For with taking k value as input

```
> # Load necessary libraries
> library(ggplot2)  # For data visualization
> library(dplyr)    # For data manipulation

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

> library(cluster)  # For clustering evaluation metrics like silhouette
> library(factoextra)  # For visualizing clusters
> library(GGally)
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
>
> print("21BDS0085 JvnGanesh")
[1] "21BDS0085 JvnGanesh"
>
> # Function to perform detailed K-means analysis
> perform_detailed_kmeans_analysis <- function() {
+
+   # Step 1: Load the iris dataset (without the Species column)
+   data <- iris %>% select(-Species)  # Removing the Species column for clustering
+   cat("\nPerforming analysis on the built-in iris dataset\n")
+
+   # Step 2: Basic dataset exploration
+   cat("\nDataset Summary:\n")
+   print(summary(data))
+
+   # Step 3: Prompt the user for the number of clusters (k)
+   k <- as.numeric(readline(prompt = "Enter the number of clusters (k): "))
+
+   # Step 4: Perform K-means clustering
+   set.seed(123)  # Setting seed for reproducibility
+   kmeans_result <- kmeans(data, centers = k)
+
+   # Step 5: Analyze and display cluster characteristics
+   cat("\nK-means Clustering Results:\n")
+   cat("Cluster Centers:\n")
+   print(kmeans_result$centers)  # Print the centers of clusters
+
+   cat("\nCluster Sizes:\n")
+   print(kmeans_result$size)  # Print the number of points in each cluster
+
+   # Step 6: Evaluate clusters using silhouette analysis
+   cat("\nEvaluating the quality of clusters using silhouette analysis...\n")
+   silhouette_avg <- silhouette(kmeans_result$cluster, dist(data))
+   print(summary(silhouette_avg))
```

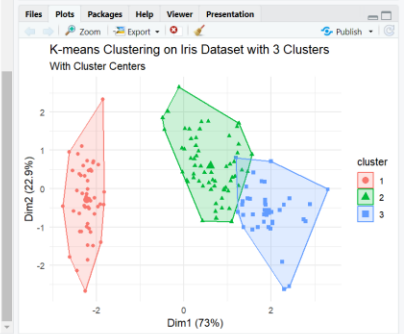



```
+   cat("\nK-means Clustering Results:\n")
+   cat("Cluster Centers:\n")
+   print(kmeans_result$centers)  # Print the centers of clusters
+
+   cat("\nCluster Sizes:\n")
+   print(kmeans_result$size)  # Print the number of points in each cluster
+
+   # Step 6: Evaluate clusters using silhouette analysis
+   cat("\nEvaluating the quality of clusters using silhouette analysis...\n")
+   silhouette_avg <- silhouette(kmeans_result$cluster, dist(data))
+   print(summary(silhouette_avg))
+
+   # Step 7: Visualization - PCA plot of clusters
+   cat("\nVisualizing clusters using PCA...\n")
+   data$Cluster <- as.factor(kmeans_result$cluster)
+
+   pca <- prcomp(data[, -ncol(data)], scale = TRUE)  # Perform PCA for visualization
+   data_pca <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2], Cluster = data$Cluster)
+
+   ggplot(data_pca, aes(x = PC1, y = PC2, color = Cluster)) +
+     geom_point(size = 3) +
+     labs(title = paste("K-means Clustering with", k, "Clusters on Iris Dataset (PCA)"),
+          x = "Principal Component 1",
+          y = "Principal Component 2") +
+     theme_minimal()
+
+   # Step 8: Visualization - Pair plot for feature combinations
+   cat("\nVisualizing pair plots for feature combinations...\n")
+   ggpairs(data, aes(color = Cluster))
+
+   # Step 9: Elbow method for optimal cluster detection
+   cat("\nVisualizing the Elbow Method to check the optimal number of clusters...\n")
+   fviz_nbclust(data[, -ncol(data)], kmeans, method = "wss") +
+     geom_vline(xintercept = k, linetype = 2) +
+     labs(subtitle = "Elbow method for optimal number of clusters")
+
+   # Step 10: Cluster visualization with cluster centers
+   cat("\nVisualizing clusters along with cluster centers...\n")
+   fviz_cluster(kmeans_result, data = data[, -ncol(data)], geom = "point") +
+     labs(title = paste("K-means Clustering on Iris Dataset with", k, "Clusters"),
+          subtitle = "With Cluster Centers") +
+     theme_minimal()
+ }
>
> # Run the function to perform a detailed K-means clustering analysis
> perform_detailed_kmeans_analysis()

Performing analysis on the built-in iris dataset

Dataset Summary:
  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
```

```
+     geom_vline(xintercept = k, linetype = 2) +
+     labs(subtitle = "Elbow method for optimal number of clusters")
+
+   # Step 10: Cluster visualization with cluster centers
+   cat("\nVisualizing clusters along with cluster centers...\n")
+   fviz_cluster(kmeans_result, data = data[, -ncol(data)], geom = "point") +
+     labs(title = paste("K-means Clustering on Iris Dataset with", k, "Clusters"),
+          subtitle = "With Cluster Centers") +
+     theme_minimal()
+ }
>
> # Run the function to perform a detailed K-means clustering analysis
> perform_detailed_kmeans_analysis()

Performing analysis on the built-in iris dataset

Dataset Summary:
  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
Enter the number of clusters (k): 3

K-means Clustering Results:
Cluster Centers:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Cluster Sizes:
[1] 50 62 38

Evaluating the quality of clusters using silhouette analysis...
Silhouette of 150 units in 3 clusters from silhouette.default(x = kmeans_result$cluster, dist = dist(data)) :
 Cluster sizes and average silhouette widths:
       50        62        38
0.7981405 0.4173199 0.4511051
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02636 0.39115 0.56231 0.55282 0.77552 0.85391

Visualizing clusters using PCA...

Visualizing pair plots for feature combinations...

Visualizing the Elbow Method to check the optimal number of clusters...

Visualizing clusters along with cluster centers...
>
```




```
+     geom_vline(xintercept = k, linetype = 2) +
+     labs(subtitle = "Elbow method for optimal number of clusters")
+
+   # Step 10: Cluster visualization with cluster centers
+   cat("\nVisualizing clusters along with cluster centers...\n")
+   fviz_cluster(kmeans_result, data = data[, -ncol(data)], geom = "point") +
+     labs(title = paste("K-means Clustering on Iris Dataset with", k, "Clusters"),
+          subtitle = "With Cluster Centers") +
+     theme_minimal()
+ }
>
> # Run the function to perform a detailed K-means clustering analysis
> perform_detailed_kmeans_analysis()

Performing analysis on the built-in iris dataset

Dataset Summary:
  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
Enter the number of clusters (k): 3

K-means Clustering Results:
Cluster Centers:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Cluster Sizes:
[1] 50 62 38

Evaluating the quality of clusters using silhouette analysis...
Silhouette of 150 units in 3 clusters from silhouette.default(x = kmeans_result$cluster, dist = dist(data)) :
 Cluster sizes and average silhouette widths:
       50        62        38
0.7981405 0.4173199 0.4511051
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02636 0.39115 0.56231 0.55282 0.77552 0.85391

Visualizing clusters using PCA...

Visualizing pair plots for feature combinations...

Visualizing the Elbow Method to check the optimal number of clusters...

Visualizing clusters along with cluster centers...
>
```

For without taking k value input

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source

Console Terminal Background Jobs

R 4.4.1 · ~/

```
> install.packages("GGally")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools
before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/lenovo/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
also installing the dependencies 'patchwork', 'ggstats'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/patchwork_1.3.0.zip'
Content type 'application/zip' length 3349341 bytes (3.2 MB)
downloaded 3.2 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/ggstats_0.7.0.zip'
Content type 'application/zip' length 1177239 bytes (1.1 MB)
downloaded 1.1 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/GGally_2.2.1.zip'
Content type 'application/zip' length 1950717 bytes (1.9 MB)
downloaded 1.9 MB

package 'patchwork' successfully unpacked and MD5 sums checked
package 'ggstats' successfully unpacked and MD5 sums checked
package 'GGally' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\lenovo\AppData\Local\Temp\RtmpCwYYDS\downloaded_packages
> # Load necessary libraries
> library(ggplot2)  # For data visualization
> library(dplyr)    # For data manipulation
> library(cluster)  # For clustering evaluation metrics like silhouette
> library(factoextra)  # For visualizing clusters
> library(GGally)
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
>
> print("21BDS0085 JvnGanesh")
[1] "21BDS0085 JvnGanesh"
> # Function to perform detailed K-means analysis
> perform_detailed_kmeans_analysis <- function() {
+
+   # Step 1: Load the iris dataset (without the Species column)
+   data <- iris %>% select(-Species)  # Removing the Species column for clustering
+   cat("\nPerforming analysis on the built-in iris dataset\n")
+
+   # Step 2: Basic dataset exploration
+   cat("\nDataset Summary:\n")
+   print(summary(data))
+
+   # Step 3: Automatically set the number of clusters (k)
+   k <- 3  # As per domain knowledge (iris has 3 species)
+
+   # Step 4: Perform K-means clustering
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source

Console Terminal Background Jobs

R 4.4.1 · ~/

```
> perform_detailed_kmeans_analysis <- function() {
+
+   # Step 1: Load the iris dataset (without the Species column)
+   data <- iris %>% select(-Species)  # Removing the Species column for clustering
+   cat("\nPerforming analysis on the built-in iris dataset\n")
+
+   # Step 2: Basic dataset exploration
+   cat("\nDataset Summary:\n")
+   print(summary(data))
+
+   # Step 3: Automatically set the number of clusters (k)
+   k <- 3  # As per domain knowledge (iris has 3 species)
+
+   # Step 4: Perform K-means clustering
+   set.seed(123)  # Setting seed for reproducibility
+   kmeans_result <- kmeans(data, centers = k)
+
+   # Step 5: Analyze and display cluster characteristics
+   cat("\nK-means Clustering Results:\n")
+   cat("Cluster Centers:\n")
+   print(kmeans_result$centers)  # Print the centers of clusters
+
+   cat("\nCluster Sizes:\n")
+   print(kmeans_result$size)  # Print the number of points in each cluster
+
+   # Step 6: Evaluate clusters using silhouette analysis
+   cat("\nEvaluating the quality of clusters using silhouette analysis...\n")
+   silhouette_avg <- silhouette(kmeans_result$cluster, dist(data))
+   print(summary(silhouette_avg))
+
+   # Step 7: Visualization - PCA plot of clusters
+   cat("\nVisualizing clusters using PCA...\n")
+   data$Cluster <- as.factor(kmeans_result$cluster)
+
+   pca <- prcomp(data[, -ncol(data)], scale = TRUE)  # Perform PCA for visualization
+   data_pca <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2], Cluster = data$Cluster)
+
+   ggplot(data_pca, aes(x = PC1, y = PC2, color = Cluster)) +
+     geom_point(size = 3) +
+     labs(title = "K-means Clustering with 3 Clusters on Iris Dataset (PCA)",
+          x = "Principal Component 1",
+          y = "Principal Component 2") +
+     theme_minimal()
+
+   # Step 8: Visualization - Pair plot for feature combinations
+   cat("\nVisualizing pair plots for feature combinations...\n")
+   ggpairs(data, aes(color = Cluster))
+
+   # Step 9: Elbow method for optimal cluster detection
+   cat("\nVisualizing the Elbow Method to check the optimal number of clusters...\n")
+   fviz_nbclust(data[, -ncol(data)], kmeans, method = "wss") +
+     geom_vline(xintercept = 3, linetype = 2) +
+     labs(subtitle = "Elbow method for optimal number of clusters")
+
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

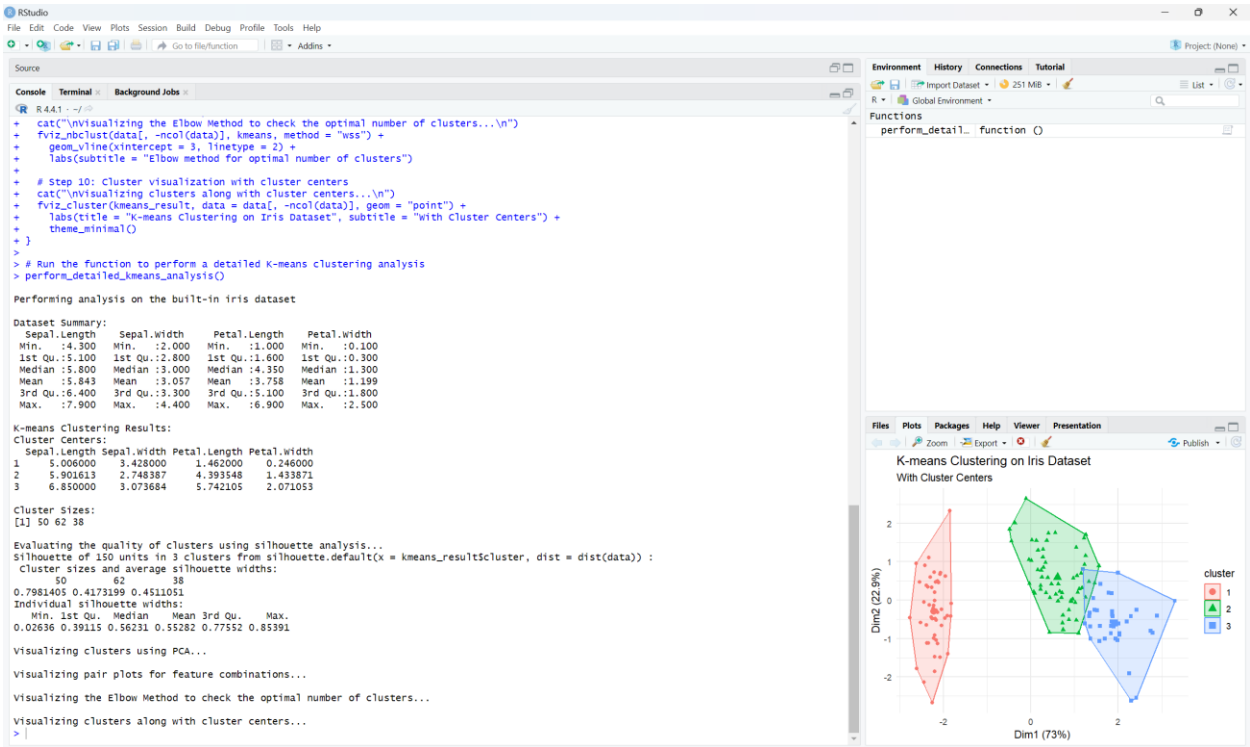
Go to file/function Addins

Source

Console Terminal Background Jobs

R 4.4.1 · ~/

```
+   # Step 8: Visualization - Pair plot for feature combinations
+   cat("\nVisualizing pair plots for feature combinations...\n")
+   ggpairs(data, aes(color = Cluster))
+
+   # Step 9: Elbow method for optimal cluster detection
+   cat("\nVisualizing the Elbow Method to check the optimal number of clusters...\n")
+   fviz_nbclust(data[, -ncol(data)], kmeans, method = "wss") +
+     geom_vline(xintercept = 3, linetype = 2) +
+     labs(subtitle = "Elbow method for optimal number of clusters")
+
+   # Step 10: Cluster visualization with cluster centers
+   cat("\nVisualizing clusters along with cluster centers...\n")
+   fviz_cluster(kmeans_result, data = data[, -ncol(data)], geom = "point") +
+     labs(title = "K-means Clustering on Iris Dataset", subtitle = "With Cluster Centers") +
+     theme_minimal()
+ }
>
> # Run the function to perform a detailed K-means clustering analysis
> perform_detailed_kmeans_analysis()

Performing analysis on the built-in iris dataset

Dataset Summary:
  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

K-means Clustering Results:
Cluster Centers:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Cluster Sizes:
[1] 50 62 38

Evaluating the quality of clusters using silhouette analysis...
Silhouette of 150 units in 3 clusters from silhouette.default(x = kmeans_result$cluster, dist = dist(data)) :
 Cluster sizes and average silhouette widths:
       50        62        38
0.7981405 0.4173199 0.4511051
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02636 0.39115 0.56231 0.55282 0.77552 0.85391

Visualizing clusters using PCA...

Visualizing pair plots for feature combinations...
```



Code with k value input:

```
# Load necessary libraries
library(ggplot2)  # For data visualization
library(dplyr)    # For data manipulation
library(cluster)  # For clustering evaluation metrics like silhouette
library(factoextra)  # For visualizing clusters
library(GGally)
```

```
print("21BDS0085 JvnGanesh")

# Function to perform detailed K-means analysis
perform_detailed_kmeans_analysis <- function() {

  # Step 1: Load the iris dataset (without the Species column)
  data <- iris %>% select(-Species)  # Removing the Species column for clustering
  cat("\nPerforming analysis on the built-in iris dataset\n")

  # Step 2: Basic dataset exploration
  cat("\nDataset Summary:\n")
```

```
print(summary(data))

# Step 3: Prompt the user for the number of clusters (k)
k <- as.numeric(readline(prompt = "Enter the number of clusters (k): "))

# Step 4: Perform K-means clustering
set.seed(123)  # Setting seed for reproducibility
kmeans_result <- kmeans(data, centers = k)

# Step 5: Analyze and display cluster characteristics
cat("\nK-means Clustering Results:\n")
cat("Cluster Centers:\n")
```

```
  print(kmeans_result$centers)  # Print the
centers of clusters


  cat("\nCluster Sizes:\n")
  print(kmeans_result$size)  # Print the
number of points in each cluster


  # Step 6: Evaluate clusters using silhouette
analysis
  cat("\nEvaluating the quality of clusters
using silhouette analysis...\n")
  silhouette_avg <-
silhouette(kmeans_result$cluster, dist(data))
  print(summary(silhouette_avg))


  # Step 7: Visualization - PCA plot of clusters
```

```
cat("\nVisualizing clusters using PCA...\n")
data$Cluster <- as.factor(kmeans_result$cluster)

pca <- prcomp(data[, -ncol(data)], scale = TRUE)  # Perform PCA for visualization
data_pca <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2], Cluster = data$Cluster)

ggplot(data_pca, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(size = 3) +
  labs(title = paste("K-means Clustering with", k, "Clusters on Iris Dataset (PCA)"),
       x = "Principal Component 1",
       y = "Principal Component 2") +
```

```
  theme_minimal()

# Step 8: Visualization - Pair plot for feature combinations
cat("\nVisualizing pair plots for feature combinations...\n")
ggpairs(data, aes(color = Cluster))

# Step 9: Elbow method for optimal cluster detection
cat("\nVisualizing the Elbow Method to check the optimal number of clusters...\n")
fviz_nbclust(data[, -ncol(data)], kmeans, method = "wss") +
  geom_vline(xintercept = k, linetype = 2) +
```

```
  labs(subtitle = "Elbow method for optimal
number of clusters")


 # Step 10: Cluster visualization with cluster
centers
 cat("\nVisualizing clusters along with cluster
centers...\n")
 fviz_cluster(kmeans_result, data = data[, -
ncol(data)], geom = "point") +
  labs(title = paste("K-means Clustering on
Iris Dataset with", k, "Clusters"),
       subtitle = "With Cluster Centers") +
  theme_minimal()
}
```

```
# Run the function to perform a detailed K-means clustering analysis
perform_detailed_kmeans_analysis()
```

Code without k value input:

```
install.packages("GGally")
# Load necessary libraries
library(ggplot2)  # For data visualization
library(dplyr)    # For data manipulation
library(cluster)  # For clustering evaluation metrics like silhouette
library(factoextra)  # For visualizing clusters
library(GGally)

print("21BDS0085 JvnGanesh")
```

```
# Function to perform detailed K-means analysis
perform_detailed_kmeans_analysis <- function() {

  # Step 1: Load the iris dataset (without the Species column)
  data <- iris %>% select(-Species)  # Removing the Species column for clustering
  cat("\nPerforming analysis on the built-in iris dataset\n")

  # Step 2: Basic dataset exploration
  cat("\nDataset Summary:\n")
  print(summary(data))
```

```
# Step 3: Automatically set the number of clusters (k)
k <- 3  # As per domain knowledge (iris has 3 species)

# Step 4: Perform K-means clustering
set.seed(123)  # Setting seed for reproducibility
kmeans_result <- kmeans(data, centers = k)

# Step 5: Analyze and display cluster characteristics
cat("\nK-means Clustering Results:\n")
cat("Cluster Centers:\n")
print(kmeans_result$centers)  # Print the centers of clusters
```

```
cat("\nCluster Sizes:\n")
print(kmeans_result$size)  # Print the number of points in each cluster

# Step 6: Evaluate clusters using silhouette analysis
cat("\nEvaluating the quality of clusters using silhouette analysis...\n")
silhouette_avg <- silhouette(kmeans_result$cluster, dist(data))
print(summary(silhouette_avg))

# Step 7: Visualization - PCA plot of clusters
cat("\nVisualizing clusters using PCA...\n")
```

```
data$Cluster <-
as.factor(kmeans_result$cluster)


pca <- prcomp(data[, -ncol(data)], scale =
TRUE)  # Perform PCA for visualization
data_pca <- data.frame(PC1 = pca$x[, 1],
PC2 = pca$x[, 2], Cluster = data$Cluster)


ggplot(data_pca, aes(x = PC1, y = PC2, color
= Cluster)) +
  geom_point(size = 3) +
  labs(title = "K-means Clustering with 3
Clusters on Iris Dataset (PCA)",
      x = "Principal Component 1",
      y = "Principal Component 2") +
  theme_minimal()
```

```
# Step 8: Visualization - Pair plot for feature combinations
cat("\nVisualizing pair plots for feature combinations...\n")
ggpairs(data, aes(color = Cluster))

# Step 9: Elbow method for optimal cluster detection
cat("\nVisualizing the Elbow Method to check the optimal number of clusters...\n")
fviz_nbclust(data[, -ncol(data)], kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2) +
  labs(subtitle = "Elbow method for optimal number of clusters")
```

```
  # Step 10: Cluster visualization with cluster centers
  cat("\nVisualizing clusters along with cluster centers...\n")
  fviz_cluster(kmeans_result, data = data[, -ncol(data)], geom = "point") +
    labs(title = "K-means Clustering on Iris Dataset", subtitle = "With Cluster Centers") +
    theme_minimal()
}

# Run the function to perform a detailed K-means clustering analysis
perform_detailed_kmeans_analysis()
```

Outputs :

install.packages("GGally")

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/

Installing package into ‘C:/Users/lenovo/AppData/Local/R/win-library/4.4’

(as ‘lib’ is unspecified)

also installing the dependencies ‘patchwork’, ‘ggstats’

```
trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.4/patchwork_1.3.0.zip'
Content type 'application/zip' length 3349341 bytes (3.2 MB)
downloaded 3.2 MB

trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.4/ggstats_0.7.0.zip'
Content type 'application/zip' length 1177239 bytes (1.1 MB)
downloaded 1.1 MB

trying URL
'https://cran.rstudio.com/bin/windows/contrib/4.4/GGally_2.2.1.zip'
```

```
Content type 'application/zip' length 1950717 bytes (1.9 MB)
downloaded 1.9 MB


package ‘patchwork’ successfully unpacked and MD5 sums checked
package ‘ggstats’ successfully unpacked and MD5 sums checked
package ‘GGally’ successfully unpacked and MD5 sums checked


The downloaded binary packages are in
        C:\Users\lenovo\AppData\Local\Temp\RtmpCwYYDS\downloaded_packages
> # Load necessary libraries
> library(ggplot2)  # For data visualization
```

```
> library(dplyr)    # For data manipulation
> library(cluster)  # For clustering evaluation metrics like silhouette
> library(factoextra)  # For visualizing clusters
> library(GGally)
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
>
> print("21BDS0085 JvnGanesh")
[1] "21BDS0085 JvnGanesh"
> # Function to perform detailed K-means analysis
> perform_detailed_kmeans_analysis <- function() {
```

```
+
+   # Step 1: Load the iris dataset (without the
Species column)
+   data <- iris %>% select(-Species)  #
Removing the Species column for clustering
+   cat("\nPerforming analysis on the built-in
iris dataset\n")
+
+   # Step 2: Basic dataset exploration
+   cat("\nDataset Summary:\n")
+   print(summary(data))
+
+   # Step 3: Automatically set the number of
clusters (k)
+   k <- 3  # As per domain knowledge (iris has
3 species)
```

```
+
+   # Step 4: Perform K-means clustering
+   set.seed(123)  # Setting seed for
reproducibility
+   kmeans_result <- kmeans(data, centers =
k)
+
+   # Step 5: Analyze and display cluster
characteristics
+   cat("\nK-means Clustering Results:\n")
+   cat("Cluster Centers:\n")
+   print(kmeans_result$centers)  # Print the
centers of clusters
+
+   cat("\nCluster Sizes:\n")
```

```
+   print(kmeans_result$size)  # Print the
number of points in each cluster
+
+   # Step 6: Evaluate clusters using
silhouette analysis
+   cat("\nEvaluating the quality of clusters
using silhouette analysis...\n")
+   silhouette_avg <-
silhouette(kmeans_result$cluster, dist(data))
+   print(summary(silhouette_avg))
+
+   # Step 7: Visualization - PCA plot of
clusters
+   cat("\nVisualizing clusters using PCA...\n")
+   data$Cluster <-
as.factor(kmeans_result$cluster)
```

```
+
+   pca <- prcomp(data[, -ncol(data)], scale =
TRUE)  # Perform PCA for visualization
+   data_pca <- data.frame(PC1 = pca$x[, 1],
PC2 = pca$x[, 2], Cluster = data$Cluster)
+
+   ggplot(data_pca, aes(x = PC1, y = PC2,
color = Cluster)) +
+     geom_point(size = 3) +
+     labs(title = "K-means Clustering with 3
Clusters on Iris Dataset (PCA)",
+          x = "Principal Component 1",
+          y = "Principal Component 2") +
+     theme_minimal()
+
```

```
+   # Step 8: Visualization - Pair plot for
feature combinations
+   cat("\nVisualizing pair plots for feature
combinations...\n")
+   ggpairs(data, aes(color = Cluster))
+
+   # Step 9: Elbow method for optimal cluster
detection
+   cat("\nVisualizing the Elbow Method to
check the optimal number of clusters...\n")
+   fviz_nbclust(data[, -ncol(data)], kmeans,
method = "wss") +
+     geom_vline(xintercept = 3, linetype = 2) +
+     labs(subtitle = "Elbow method for optimal
number of clusters")
+
```

```
+   # Step 10: Cluster visualization with
cluster centers
+   cat("\nVisualizing clusters along with
cluster centers...\n")
+   fviz_cluster(kmeans_result, data = data[, -
ncol(data)], geom = "point") +
+     labs(title = "K-means Clustering on Iris
Dataset", subtitle = "With Cluster Centers") +
+     theme_minimal()
+ }
>
> # Run the function to perform a detailed K-
means clustering analysis
> perform_detailed_kmeans_analysis()
```

Performing analysis on the built-in iris dataset

Dataset Summary:

```
 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


K-means Clustering Results:

Cluster Centers:

  Sepal.Length Sepal.Width Petal.Length
Petal.Width

1     5.006000    3.428000     1.462000
0.246000

2     5.901613    2.748387     4.393548
1.433871

3     6.850000    3.073684     5.742105
2.071053


Cluster Sizes:

[1] 50 62 38
```

Evaluating the quality of clusters using silhouette analysis...

Silhouette of 150 units in 3 clusters from silhouette.default(x = kmeans_result$cluster, dist = dist(data)) :

Cluster sizes and average silhouette widths:

```
      50       62       38
0.7981405 0.4173199 0.4511051
```

Individual silhouette widths:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02636 0.39115 0.56231 0.55282 0.77552 0.85391
```

Visualizing clusters using PCA...

Visualizing pair plots for feature combinations...

Visualizing the Elbow Method to check the optimal number of clusters...

Visualizing clusters along with cluster centers...