NAME:VEDASHREEBHALERAO

ROLL NO:2213191

BATCH:B

SUBJECT:BDA LAB ASSIGNMENT NO 5

Big Data Analytics Experiment no. 05

Aim: To implement the K-means clustering algorithm using R programming and analyze the results graphically. The analysis includes selecting an appropriate value for justifying the choice, and interpreting the output.

Theory:

K-means Clustering Algorithm:

K-means is a widely used unsupervised learning algorithm that partitions a dataset into k clusters. The goal is to minimize the variance within each cluster. The algorithm follows these steps:

1. **Initialization:** Choose k initial centroids randomly from the dataset.
2. **Assignment:** Assign each data point to the nearest centroid, forming k clusters.
3. **Update:** Calculate the new centroids by taking the mean of all data points in each cluster.
4. **Repeat:** Repeat the assignment and update steps until the centroids no longer change or a predefined number of iterations is reached.

Selecting the Value of k:

Choosing the right value for k is crucial for the performance of the K-means algorithm. Common methods to determine the optimal k include:

- **Elbow Method:** Plot the sum of squared distances from each point to its assigned centroid (within-cluster sum of squares, WCSS) for different values of k. The optimal k is typically at the "elbow" point, where the rate of decrease sharply slows.
- **Silhouette Analysis:** Measures how similar each point is to its own cluster compared to other clusters. The optimal k maximizes the average silhouette score.
- **Gap Statistic:** Compares the WCSS with that expected under a null reference distribution of the data.

Experiment Details

Implementation in R:

1. **Load the necessary libraries:**

```
library(ggplot2)
library(cluster)
```

2. **Load and preprocess the dataset:**

```
data <- read.csv('path_to_dataset.csv')
```

```
data <- na.omit(data)  # Handle missing values
```

3. **Determine the optimal value of k using the Elbow Method:**

```
set.seed(123)
wcss <- vector()
for (i in 1:10) {
  kmeans_model <- kmeans(data, centers = i)
  wcss[i] <- sum(kmeans_model$tot.withinss)
}
plot(1:10, wcss, type = 'b', main = 'Elbow Method', xlab = 'Number of clusters
(k)', ylab = 'WCSS')
```

4. **Apply the K-means algorithm with the selected k**

```
optimal_k <- 3  # Assume 3 is the optimal k from the Elbow Method
kmeans_result <- kmeans(data, centers = optimal_k)
data$cluster <- as.factor(kmeans_result$cluster)
```

5. **Visualize the results:**

```
ggplot(data, aes(x = Feature1, y = Feature2, color = cluster)) +
  geom_point() +
  labs(title = 'K-means Clustering', x = 'Feature 1', y = 'Feature 2')
```

Conclusion

In this experiment, we successfully implemented the K-means clustering algorithm in R. The Elbow Method was used to determine the optimal value of k, which was found to be 3. The clustering results were visualized, showing distinct clusters. The choice of k is crucial as it directly affects the cluster formation and overall analysis. By using the Elbow Method, we ensured a balance between simplicity and accuracy, avoiding both underfitting and overfitting. Further analysis, such as silhouette scores or the gap statistic, could be used to validate the chosen k.

CODE:

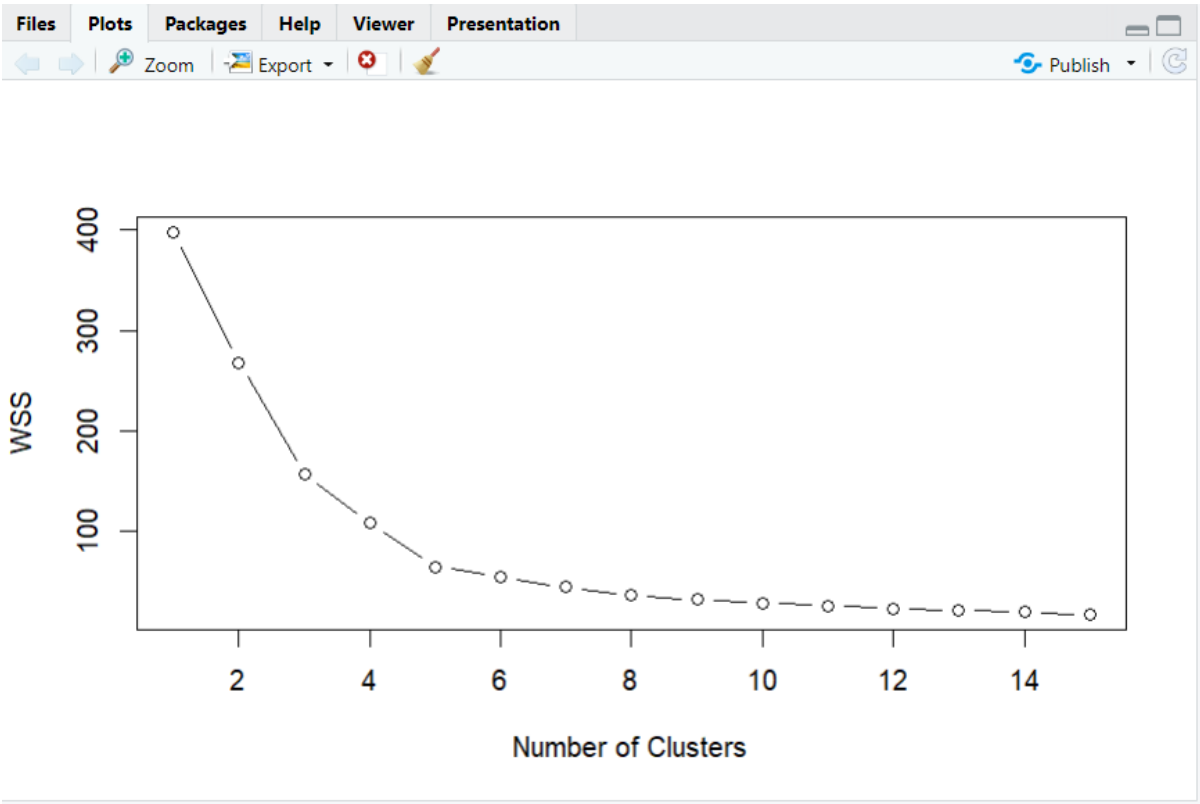
```
install.packages("ggplot2")
install.packages("factoextra")
install.packages("dplyr")
install.packages("colorspace")


library(ggplot2)
library(factoextra)
```

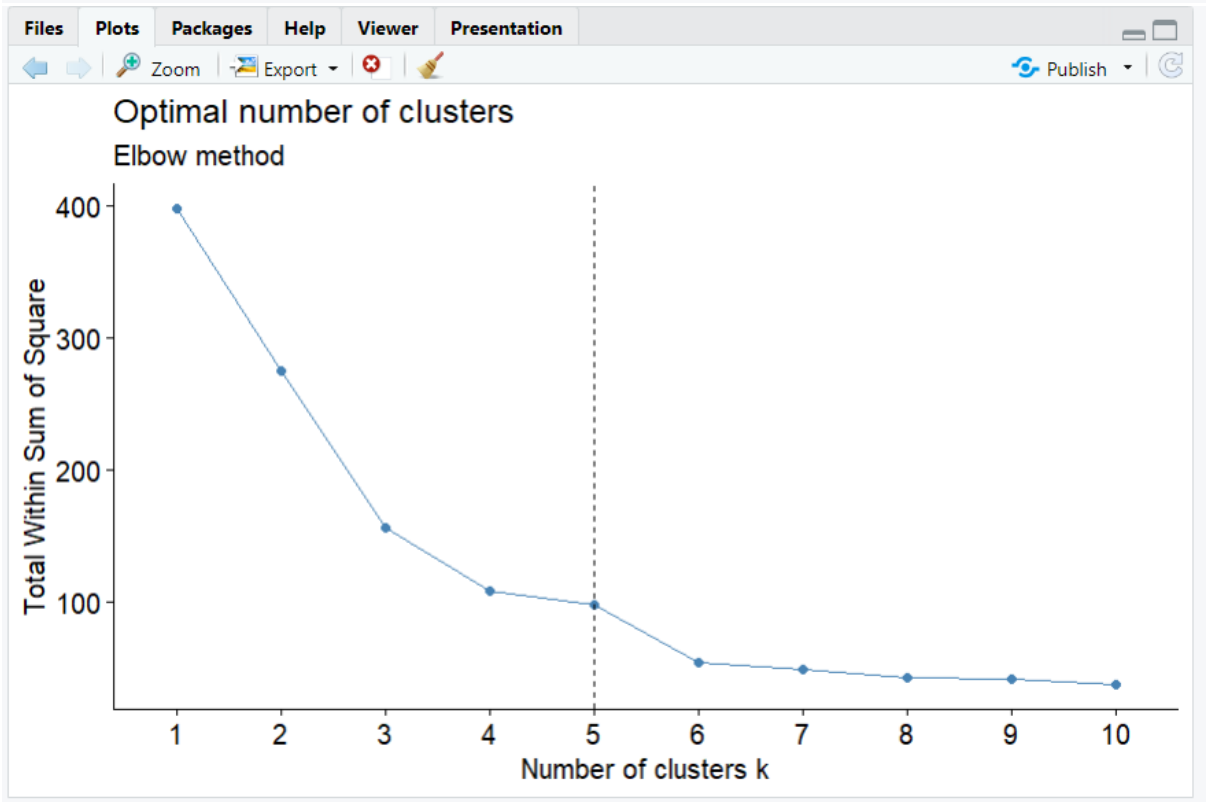
```
library(dplyr)

mall_customers = read.csv("C:\\Users\\user\\OneDrive\\Documents\\Mall_Customers.csv")
head(mall_customers)

names(mall_customers)

print(colnames(mall_customers))
data <- mall_customers[c("Annual.Income..k..", "Spending.Score..1.100.")]

head(data)

data_scaled <- scale(data)
print("scaled data")
head(data_scaled)

wss <- numeric(15)
for (k in 1:15) wss[k] <- sum(kmeans(data_scaled, centers=k,
                                     nstart=25)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="WSS")
fviz_nbclust(data_scaled, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2) +
  labs(subtitle = "Elbow method")



# Silhouette method
```

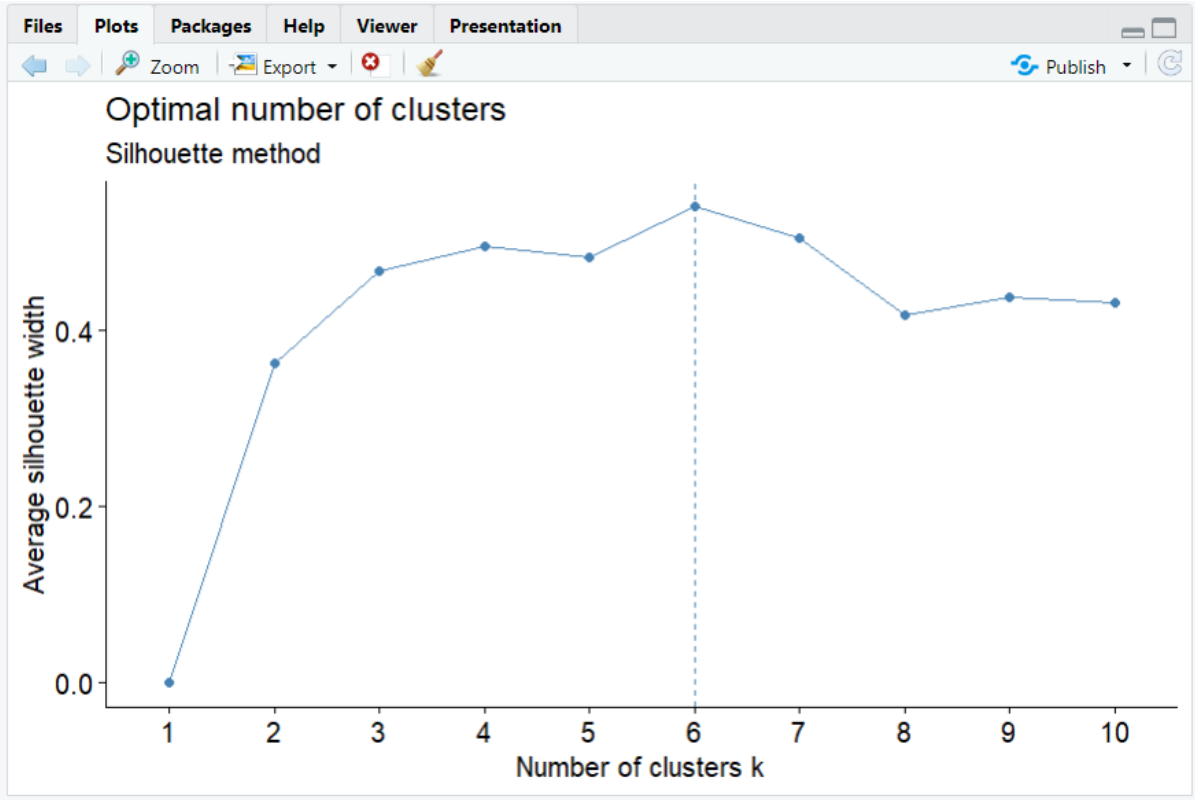
```
fviz_nbclust(data_scaled, kmeans, method = "silhouette") +
  labs(subtitle = "Silhouette method")

# Apply K-Means clustering with 3 clusters
set.seed(123)   # Set seed for reproducibility in case python-random_state
kmeans_result <- kmeans(data_scaled, centers = 5, nstart = 25)

# Add the cluster assignments to the original dataset
mall_customers$Cluster <- as.factor(kmeans_result$cluster)
tail(mall_customers)

# Scatter plot of the clusters
ggplot(mall_customers, aes(x = AnnualIncome, y = SpendingScore, color = Cluster)) +
  geom_point(size = 3) +
  scale_color_manual(values = c("red", "blue", "green","yellow","black")) +
  labs(title = "K-Means Clustering of Mall Customers",
       x = "Annual Income (k$)",
       y = "Spending Score (1-100)") +
  theme_minimal()

# Scatter plot with cluster centers
fviz_cluster(kmeans_result, data = data_scaled,
             geom = "point",
             ellipse.type = "norm",
             ggtheme = theme_minimal(),
             palette = c("red", "blue", "green","yellow","black"))
```

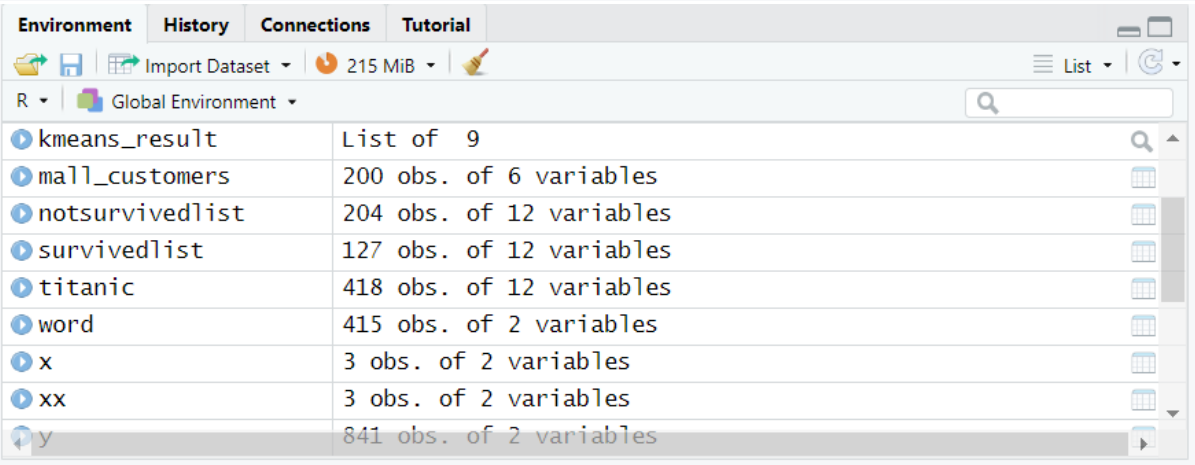
kmeans_result

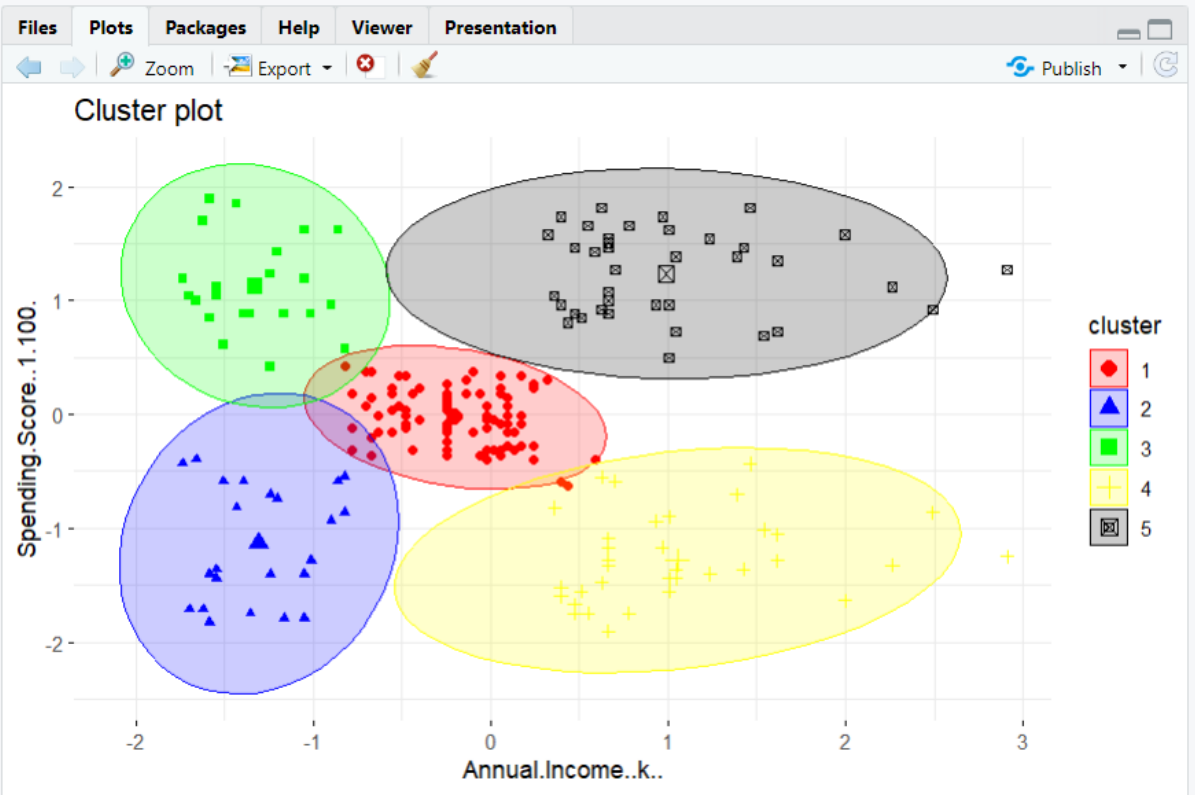
OUTPUT:






Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Optimal number of clusters
Silhouette method
Average silhouette width
0.0
0.2
0.4
1
2
3
4
5
6
7
8
9
10
Number of clusters k


Environment
History
Connections
Tutorial
Import Dataset
215 MiB
List
R
Global Environment
kmeans_result List of 9
mall_customers 200 obs. of 6 variables
notsurvivedlist 204 obs. of 12 variables
survivedlist 127 obs. of 12 variables
titanic 418 obs. of 12 variables
word 415 obs. of 2 variables
x 3 obs. of 2 variables
xx 3 obs. of 2 variables
y 841 obs. of 2 variables


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Cluster plot
Spending.Score..1.100.
2
1
0
-1
-2
-2
-1
0
1
2
3
Annual.Income..k..
cluster
1
2
3
4
5

```
+              ggtheme = theme_minimal(),
+              palette = c("red", "blue", "green","yellow","black"))
> kmeans_result
K-means clustering with 5 clusters of sizes 81, 23, 22, 35, 39

Cluster means:
  Annual.Income..k.. Spending.Score..1.100.
1         -0.2004097            -0.02638995
2         -1.3042458            -1.13411939
3         -1.3262173             1.12934389
4          1.0523622            -1.28122394
5          0.9891010             1.23640011

Clustering vector:
  [1] 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 1 2
 [46] 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [91] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 4 5 1 5 4 5 4 5 1 5 4
[136] 5 4 5 4 5 4 5 1 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5
[181] 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5

Within cluster sum of squares by cluster:
[1] 14.485632  7.577407  5.217630 18.304646 19.655252
 (between_SS / total_SS =  83.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> |
```