EXPERIMENT 9

Taniya Ahmed

21BDS0059

1. **Loading the libraries used, importing the dataset and renaming the column names.**

CODE:

```
library(ggplot2)

df = read.csv("D:\\Downloads\\Mall_Customers.csv")

colnames(df)

colnames(df) = c("CustomerID", "Gender", "Age", "AnnualIncome", "SpendingScore")

head(df)

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> library(ggplot2)
>
> df = read.csv("D:\\Downloads\\Mall_Customers.csv")
> colnames(df)
[1] "CustomerID"             "Gender"                 "Age"                    "Annual.Income..k.."
[5] "Spending.Score..1.100."
> colnames(df) = c("CustomerID", "Gender", "Age", "AnnualIncome", "SpendingScore")
> head(df)
  CustomerID Gender Age AnnualIncome SpendingScore
1          1   Male  19           15            39
2          2   Male  21           15            81
3          3 Female  20           16             6
4          4 Female  23           16            77
5          5 Female  31           17            40
6          6 Female  22           17            76
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
```

2. **Selecting features based on which the clustering will be done.**

CODE:

```
cluster_features = df[, c("AnnualIncome", "SpendingScore")]

head(cluster_features)

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> cluster_features = df[, c("AnnualIncome", "SpendingScore")]
> head(cluster_features)
  AnnualIncome SpendingScore
1           15            39
2           15            81
3           16             6
4           16            77
5           17            40
6           17            76
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

3. **Calculating shortest distance between points using Euclidean distance.**

CODE:

```
euclid_distance = function(a, b){
 return(sqrt(sum((a - b) ^ 2)))
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> euclid_distance = function(a, b){
+   return(sqrt(sum((a - b) ^ 2)))
+ }
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

4. **Assigning clusters according to shortest distance between the points and the centroids.**

CODE:

```
assign_cluster = function(data, centroids){
 clusters = vector("numeric", nrow(data))

 for(i in 1 : nrow(data)){
  distances = apply(centroids, 1, euclid_distance, b= data[i, ])
  clusters[i] = which.min(distances)
 }
```

```
 return(clusters)
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> assign_cluster = function(data, centroids){
+   clusters = vector("numeric", nrow(data))
+
+   for(i in 1 : nrow(data)){
+     distances = apply(centroids, 1, euclid_distance, b= data[i, ])
+     clusters[i] = which.min(distances)
+   }
+
+   return(clusters)
+ }
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

5. **Calculating the new centroids as every iteration requires a centroid calculation.**

CODE:

```
new_centroid_calculation = function(data, clusters, k){
 centroids = matrix(NA, nrow = k, ncol = ncol(data))

 for(i in 1 : k){
  centroids[i, ] = colMeans(data[clusters == i, , drop = FALSE])
 }
 return(centroids)
}
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> new_centroid_calculation = function(data, clusters, k){
+   centroids = matrix(NA, nrow = k, ncol = ncol(data))
+
+   for(i in 1 : k){
+     centroids[i, ] = colMeans(data[clusters == i, , drop = FALSE])
+   }
+
+   return(centroids)
+ }
> new_centroid_calculation = function(data, clusters, k){
+   centroids = matrix(NA, nrow = k, ncol = ncol(data))
+
+   for(i in 1 : k){
+     centroids[i, ] = colMeans(data[clusters == i, , drop = FALSE])
+   }
+
+   return(centroids)
+ }
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```

6. **Designing the custom kmeans algorithm by assigning clusters, , calculating centroids and iterating multiple times until convergence or 150 iterations whichever happens first.**

CODE:

```
kmeans_custom = function(data, k, max_iter = 150){
 set.seed(125)
 ini_centroid = data[sample(1 : nrow(data), k),]
 centroids = ini_centroid
 clusters = assign_cluster(data, centroids)

 for(i in 1 : max_iter){
  centroids = new_centroid_calculation(data, clusters, k)
  new_clusters = assign_cluster(data, centroids)

  if(all(clusters == new_clusters)){
   break
  }

  clusters = new_clusters
 }
```

```
 return(list(clusters = clusters, centroids = centroids))
}

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

```
> kmeans_custom = function(data, k, max_iter = 150){
+   set.seed(125)
+   ini_centroid = data[sample(1 : nrow(data), k),]
+   centroids = ini_centroid
+   clusters = assign_cluster(data, centroids)
+
+   for(i in 1 : max_iter){
+     centroids = new_centroid_calculation(data, clusters, k)
+     new_clusters = assign_cluster(data, centroids)
+
+     if(all(clusters == new_clusters)){
+       break
+     }
+
+     clusters = new_clusters
+   }
+
+   return(list(clusters = clusters, centroids = centroids))
+ }
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
```

7. **Specifying the number of clusters, checking the clusters and centroids.**

CODE:

```
k = 4

kmeans_final = kmeans_custom(cluster_features, k)


print(kmeans_final$clusters)

print(kmeans_final$centroids)

print("Taniya Ahmed 21BDS0059")
```

OUTPUT:

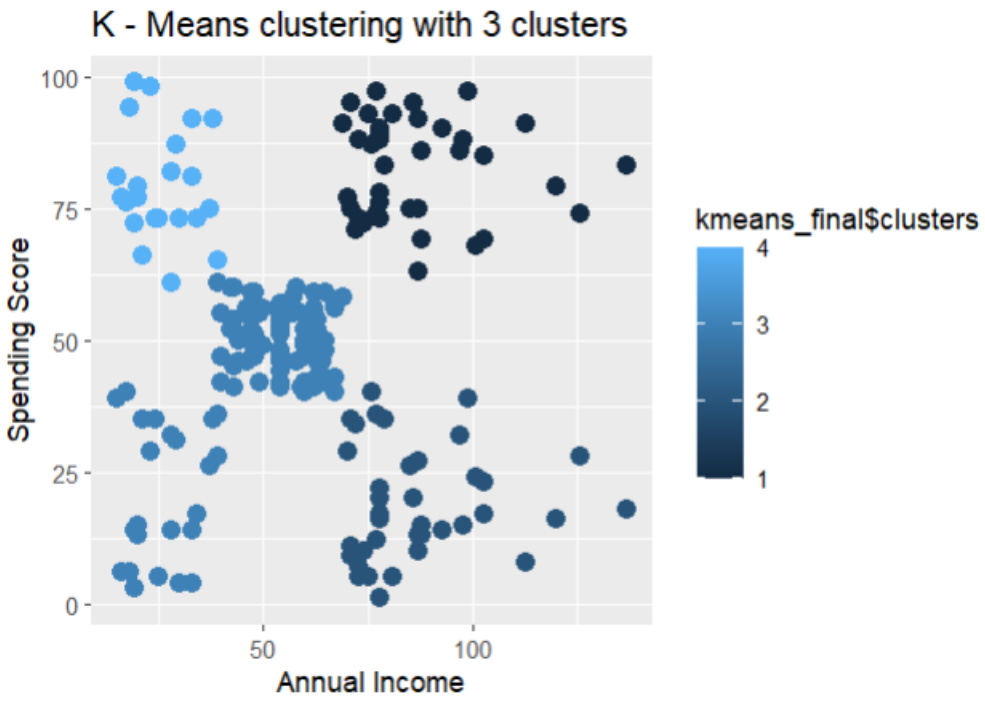
```
> k = 4
> kmeans_final = kmeans_custom(cluster_features, k)
> print(kmeans_final$clusters)
  [1] 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 3 3 4 3 3 3 3 3 3 3 3 3
 [57] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[113] 3 3 3 3 3 3 3 3 3 3 3 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
[169] 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
> print(kmeans_final$centroids)
         [,1]     [,2]
[1,] 86.53846 82.12821
[2,] 87.00000 18.63158
[3,] 48.16832 43.39604
[4,] 25.72727 79.36364
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
```

8. **Making a scatter plot to evaluate the clusters.**

CODE:

```
ggplot(df, aes(x = AnnualIncome, y = SpendingScore, color = kmeans_final$clusters)) +
 geom_point(size = 3) +
 labs(title = "K - Means clustering with 3 clusters", x = "Annual Income", y = "Spending Score")
print("Taniya Ahmed 21BDS0059")
```

OUTPUT:




```
> ggplot(df, aes(x = AnnualIncome, y = SpendingScore, color = kmeans_final$clusters)) +
+   geom_point(size = 3) +
+   labs(title = "K - Means clustering with 3 clusters", x = "Annual Income", y = "Spending Score")
> print("Taniya Ahmed 21BDS0059")
[1] "Taniya Ahmed 21BDS0059"
>
```