Experiment no- 9

Objective- Implement K-mean Clustering Algorithm.

K-Means Clustering Algorithm

K-Means is a **popular unsupervised machine learning algorithm** used for **clustering** data into **K distinct groups** based on the similarity of data points. It works by dividing a dataset into K clusters, where each data point belongs to the cluster with the **nearest mean**. The algorithm is particularly useful in **data analytics** for tasks like customer segmentation, anomaly detection, and image compression.

Steps in K-Means Clustering:

1. **Initialization**:

- Choose **K** (the number of clusters) and randomly initialize **K centroids** (center points of clusters).

2. **Assignment Step**:

- Assign each data point to the nearest centroid based on the **Euclidean distance** (or other distance metrics).

3. **Update Step**:

- After assigning points, **update** the centroids by calculating the **mean** of all points in each cluster.

4. **Repeat**:

- Repeat the assignment and update steps until convergence, meaning the centroids no longer change significantly, or a maximum number of iterations is reached.

Key Features:

- **Scalability**: K-Means can handle large datasets efficiently.
- **Simplicity**: It's easy to understand and implement.
- **Distance Metric**: Uses distance (usually Euclidean) to measure similarity between points and centroids.

Challenges:

- **Choosing K**: The optimal number of clusters (K) is often not obvious and may require methods like the **elbow method** to determine.
- **Sensitivity to Initialization**: Poor initialization of centroids can lead to suboptimal clustering, though techniques like **K-Means++** can help.
- **Assumption of Spherical Clusters**: K-Means assumes clusters are spherical and of similar size, which can be a limitation for complex data distributions.

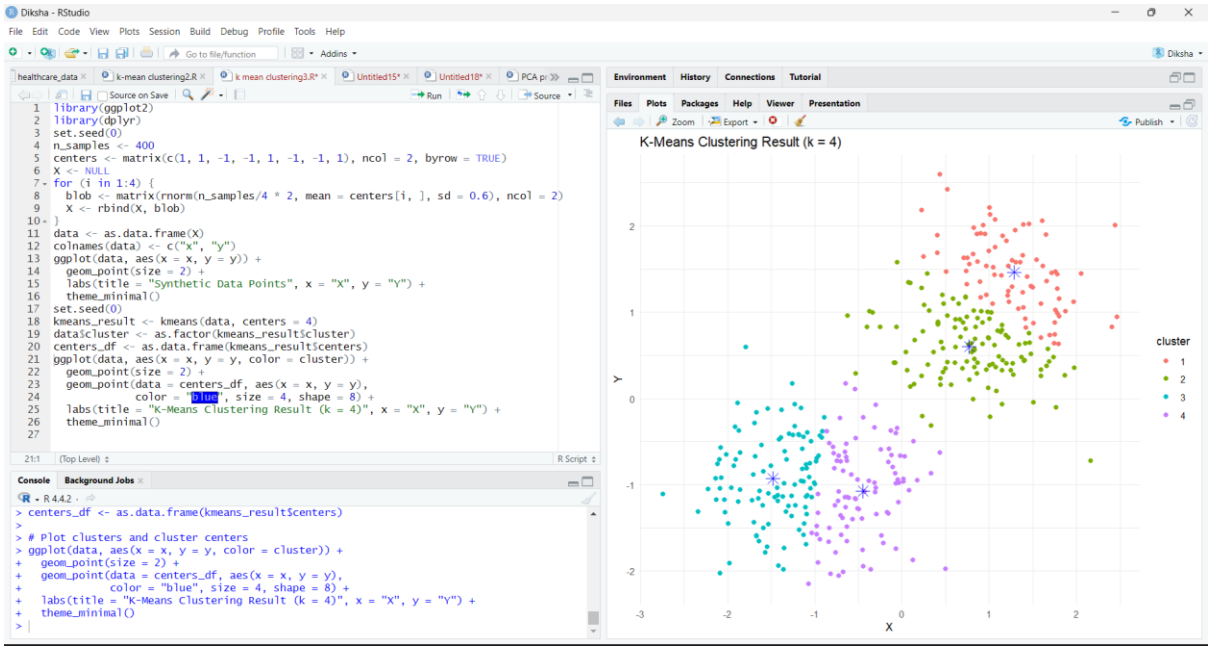
Diksha Kumari
Roll no- 2304221529005

Applications in Data Analytics:

- **Customer Segmentation**: Grouping customers based on their purchasing behavior or demographics.
- **Market Basket Analysis**: Identifying similar products often bought together.
- **Image Compression**: Reducing the size of an image by representing it with a smaller number of representative colors.
- **Anomaly Detection**: Identifying outliers or rare events by examining clusters.

Example of Experiment:

Example 1. In this experiment, we apply the **K-Means clustering** algorithm to a synthetic dataset of 400 data points, grouped into **4 distinct clusters**. The dataset is generated using random data points (referred to as "blobs") around four predefined centers. These blobs represent naturally occurring clusters that we aim to identify using the K-Means algorithm.

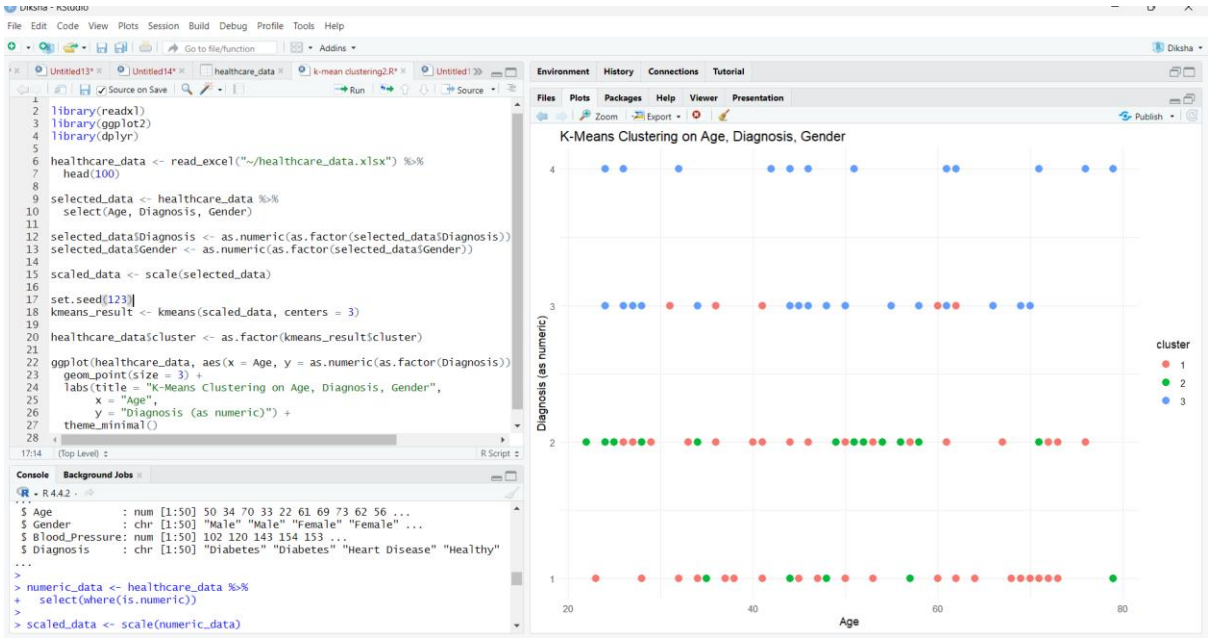


Example 2- The goal of this experiment is to apply the **K-Means clustering algorithm** to a healthcare dataset to identify patterns or groupings based on key variables: **Age**, **Diagnosis**, and **Gender**. The dataset is from a healthcare context, where these features may help in categorizing patients into different groups based on their health characteristics.

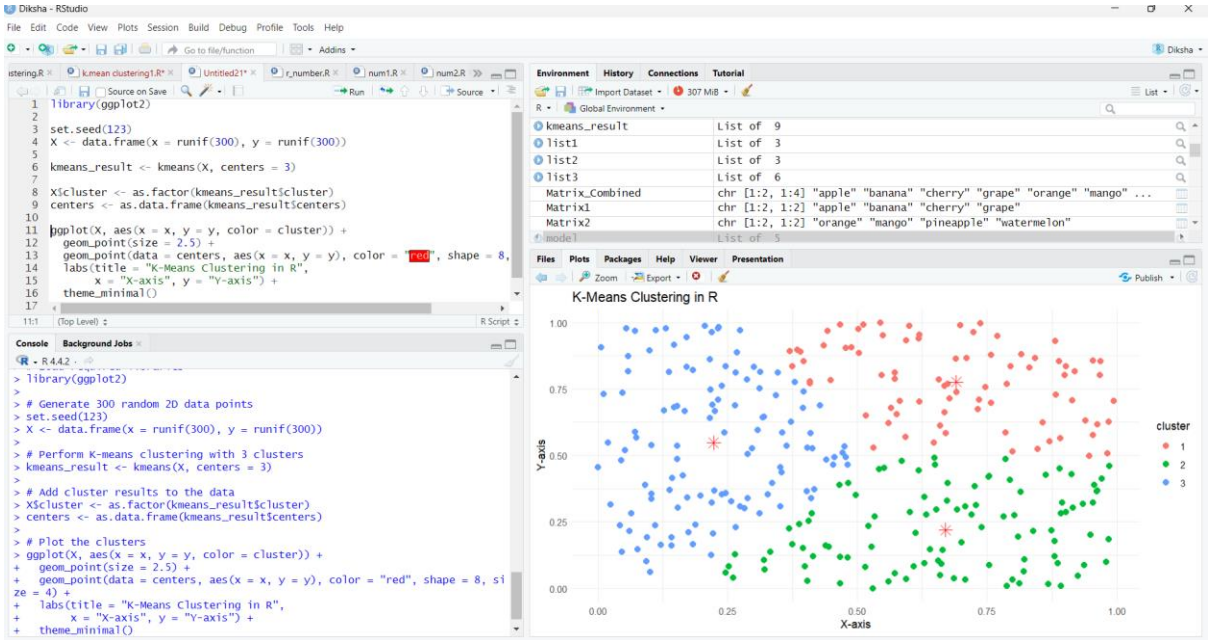
The **Diagnosis** and **Gender** columns are converted into numeric format using as.factor() and as.numeric(), enabling the use of these variables in K-Means clustering.

- The K-Means algorithm is applied with **3 clusters** (a predefined choice based on domain knowledge or experimentation).
- The K-Means algorithm assigns each data point to one of the 3 clusters based on the similarity of the features (Age, Diagnosis, and Gender).

Diksha Kumari
Roll no- 2304221529005



Example 3-In this experiment, we performed **K-Means clustering** on a set of randomly generated 2D points to understand how the algorithm groups data based on similarity. We used R's runif() function to generate 300 random points with x and y coordinates between 0 and 1. These points represent an unlabelled dataset, ideal for unsupervised learning techniques like clustering.



Diksha Kumari
Roll no- 2304221529005