**Appling k-means Algorithm in Graduation Rate Dataset**

**About dataset:**

The dataset includes 1000 rows, with one row for each high school in the dataset. The graduation rates for each school were generated randomly and are not based on any actual data. This dataset could be useful for exploring trends in graduation rates at an terminate time, comparing graduation rates between different regions or states, or analyzing factors that may be associated with changes in graduation rates over time. However, it is important to keep in mind that the data is not based on actual data and should be used for exploratory or educational purposes only. [1]

**Columns**:

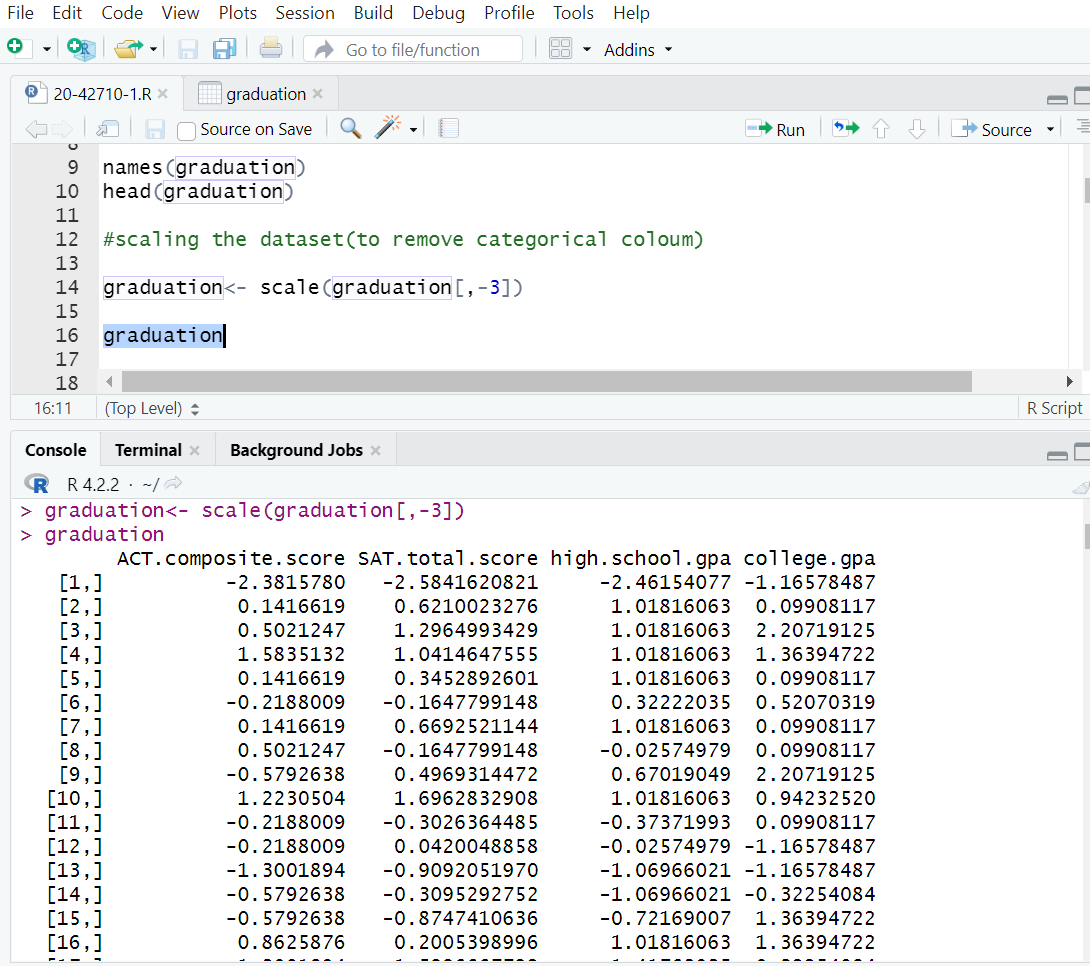
* **ACT composite score**
* **SAT total score**
* **Parental level of education**
* **Parental income**
* **high school GPA**
* **college GPA**
* **years to graduate**

**APPLING ALGORITHM:**

Before applying algorithm, I imported dataset and found out about its value by **names ()** & **head ()** functions. I followed those steps for k means

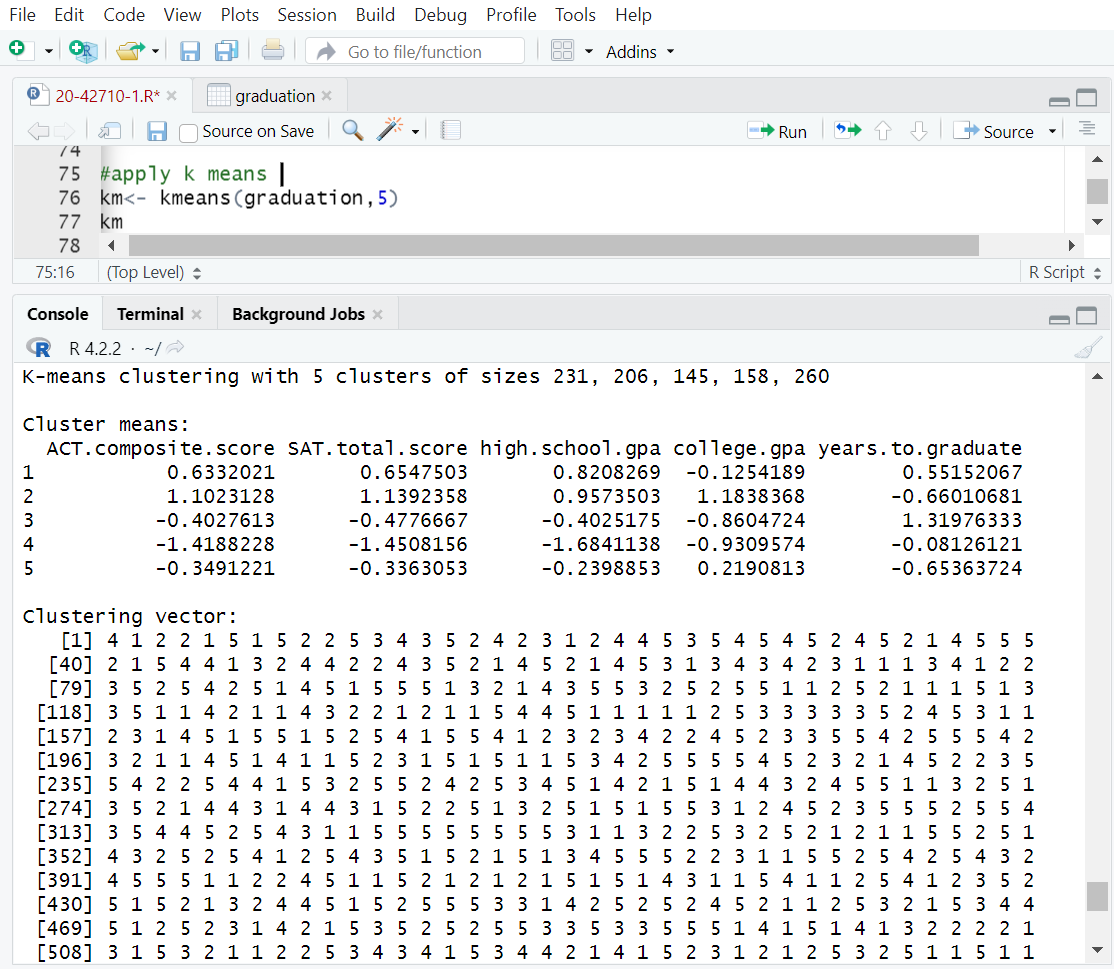
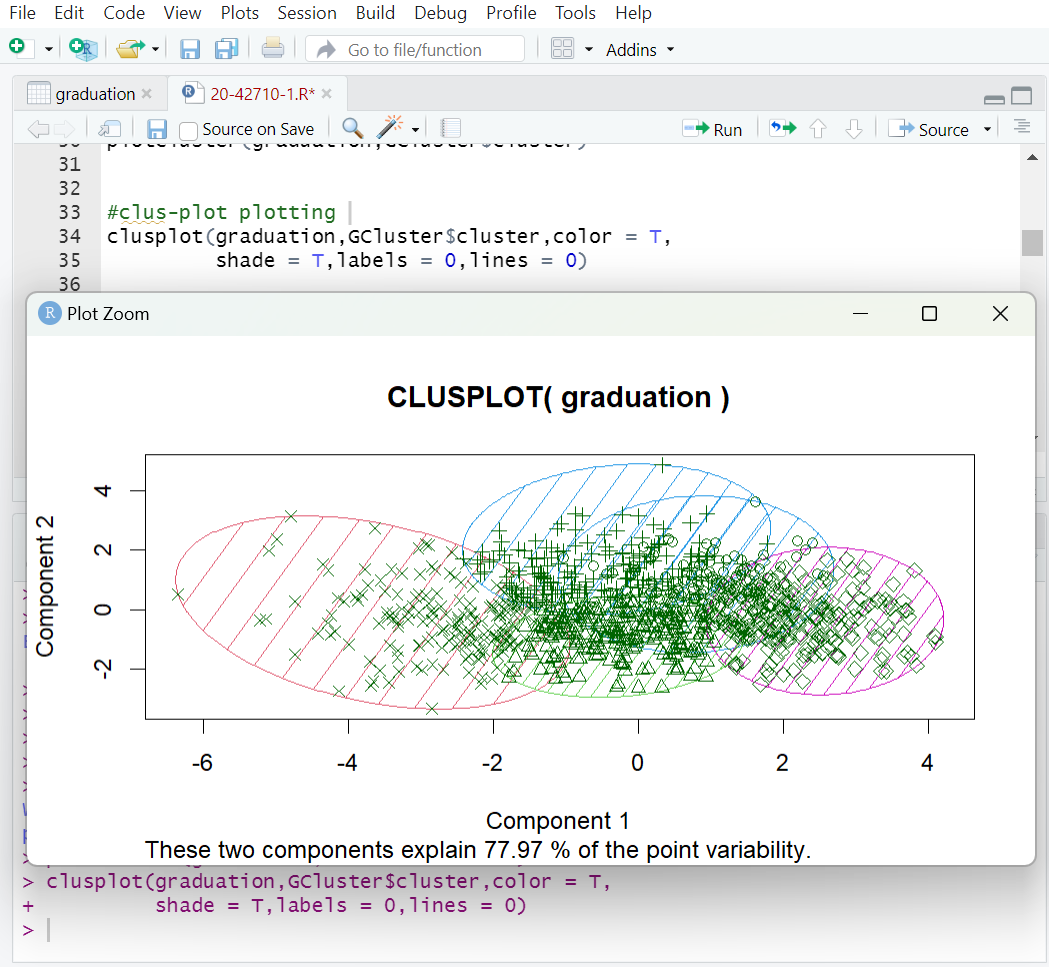
1. **Scaling**
2. **K means**
3. **Different Clustering method**
4. **Optimal clustering**
5. **Again, Different Clustering method with optimal value**
6. **Calculate cluster centers & Summary of the dataset**

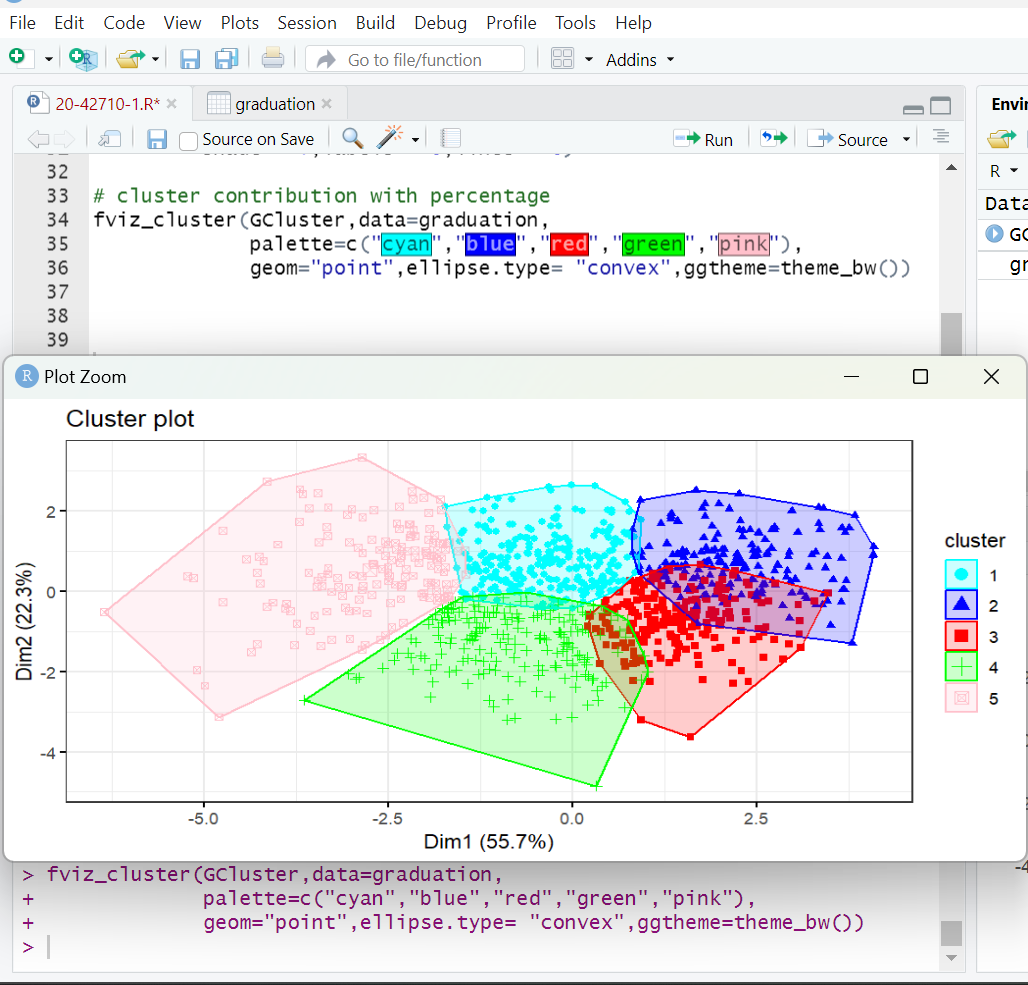
**Scaling dataset:**

In my dataset all the features are numeric but only 3rd feature is categorical so that I scale 3rd column of the graduation rate dataset to ensure that the gradient descent moves smoothly towards the minima and that the steps for gradient descent are updated at the same rate for all the features, we scale the data before feeding it to the model. For scaling the all column, I applied **scale ()** function.

**APPLING K MEANS:**

k-means is capable of being applied to data that has a smaller number of dimensions, is numeric, and is continuous. I applied **kmeans()** function to clustering which centers is 5 (randomly select) after applying k-means function, I used plotcluster() to plot my clusters dataset. I would like to have as a label the row name of each datapoint instead of the number of its cluster and preserve the color to separate clusters. I used this code to produce the plot, where graduation is a matrix and **Gcluster$cluster** is the cluster, each point is assigned to. Here our clustering point is 1 to 5. Some of the labels overlap other labels in picture 1. There was another way to plot, that is clusplot. I use **clusplot()** function for this where color & shade is true and labels and lines are 0. In this function each labels are represented in a circle and shows the overlapping in picture 2. Then I check clusters contribute with their percentage by using **fviz**\_**cluster()** function in picture 3. In the picture 3, dimension 1 percentage is maximum. which is 55.7%.

**picture 1&2**

**Graphical user interface, application

Description automatically generatedpicture 3**

**Picture 4**

**Clustering in different way**

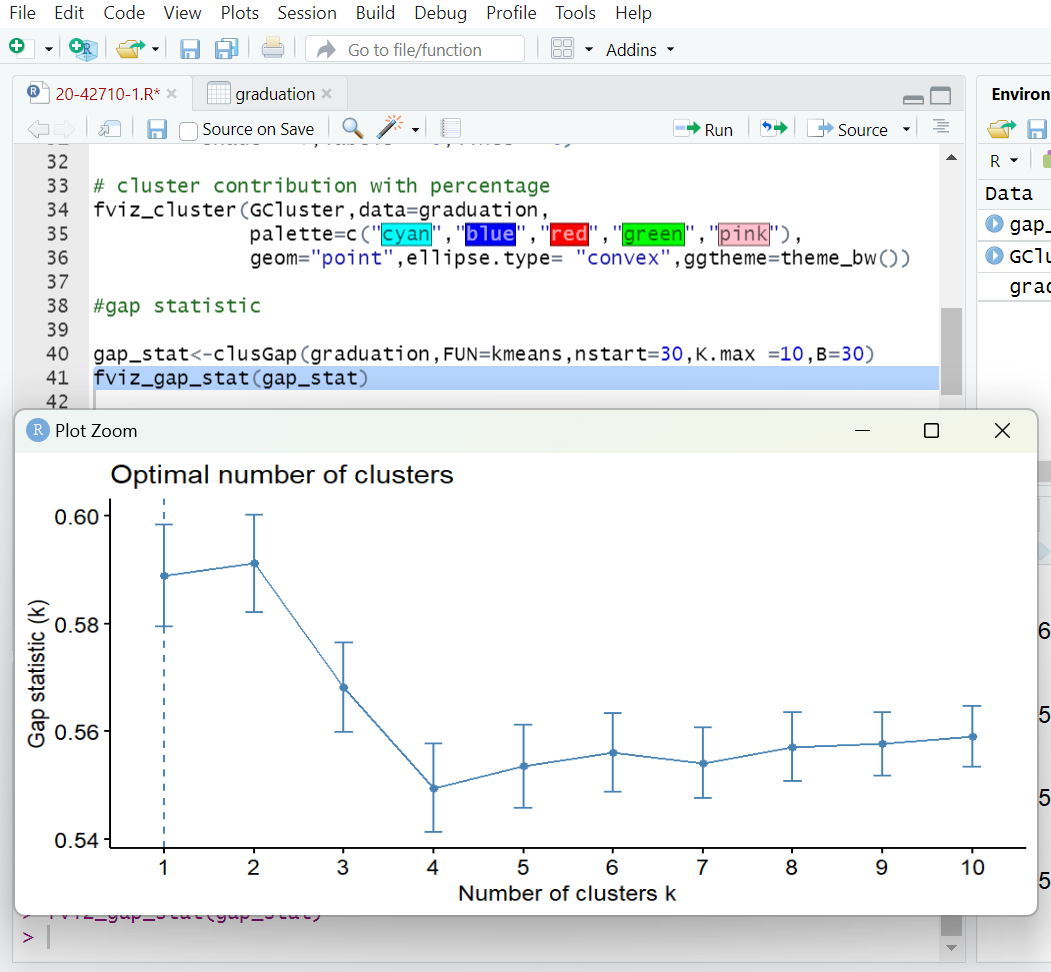
K-means is a data clustering approach for unsupervised machine learning that is capable of separating unlabeled data into a predetermined number of disjoint groups of equal variances. There are so many method to cluster a dataset. Like

1. **gap statistic method**
2. **silhouette method**
3. **within sum of square method (to get optimal clustering)**
4. **NbCluste Function (to get optimal clustering)**

I applied all those ways 2 times because first time with random value of k then with optimal value of k = 2. To applying all the functions, I had to install some library package and call them.

**Gap Statistic**

Gap statistic is a goodness of clustering measure, where for each hypothetical number of clusters k, it compares two functions: log of within-cluster sum of squares (wss) with its expectation under the null reference distribution of the data. I applied **clusGap()** function to iteration where k.max is 10. I randomly declare the starting portion is 30 when center is a number and B for bootstrapping is 30. Again, I cluster with center 3 in picture 2 showed that.

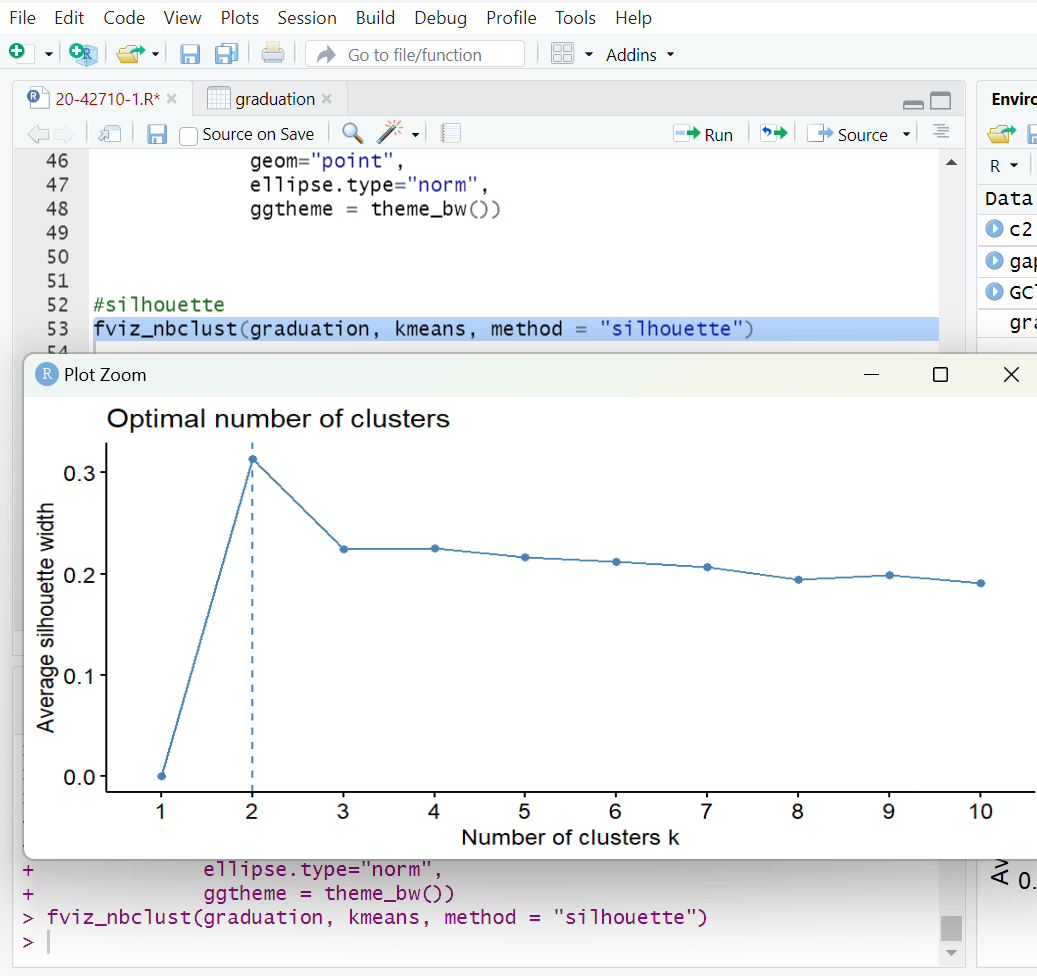
****

**Picture 1**

**Chart

Description automatically generated**

**Silhouette Method**

****The silhouette value expresses an object's cohesiveness, or similarity to its own cluster, in relation to other clusters, or separation. A high number on the silhouette implies that the object is well matched to its own cluster and poorly matched to nearby means of clusters. I used **fviz\_nbclust()** function and method is silhouette to get optimal clustering point. Here it shows that the optimal number of clustering is 2 Which is marks by the dummy line. In x axis it shows number of clustering and y axis show average silhouette width (in floating number). Which value is always - 1 to +1.

**wSS method**

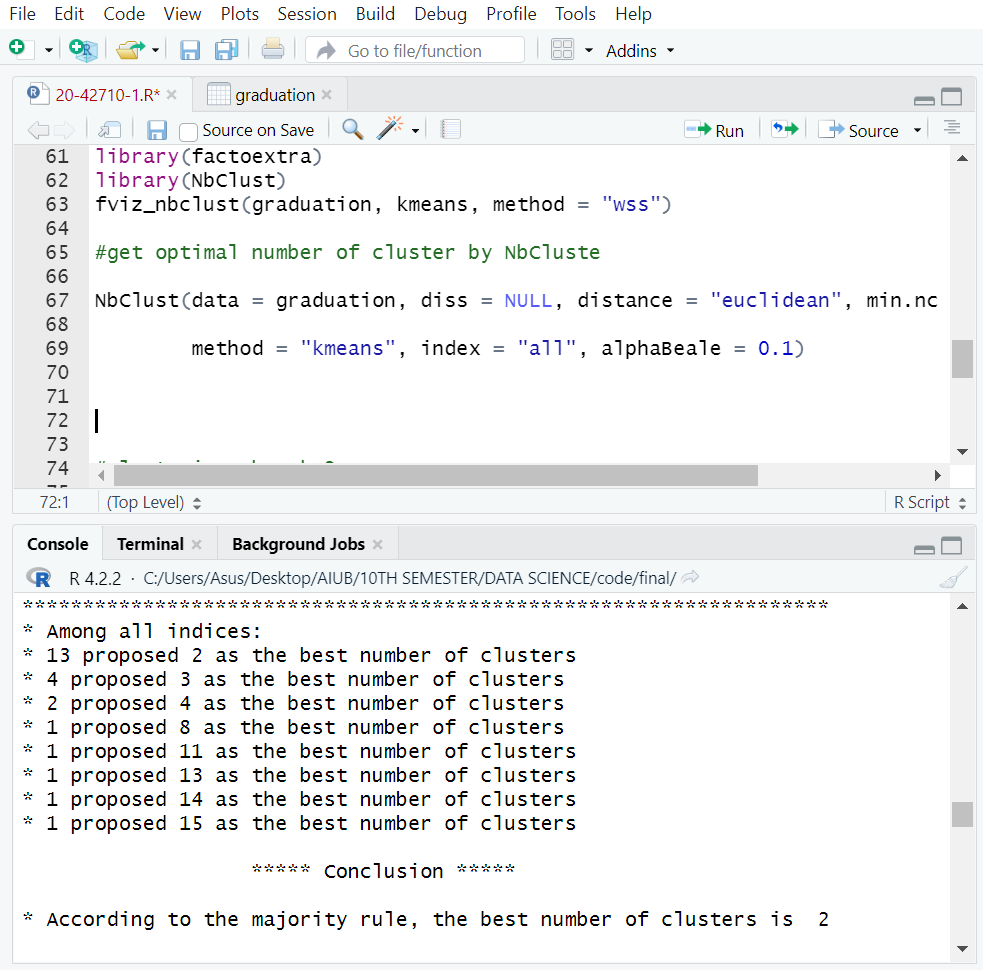
The WSS Plot, also known as "Within Sum of Squares," is another K-Means algorithm solution which helps in determining the value of K (number of clusters). The variation from each observation in the clusters to its centroid, added together to get a value, will be the values used to construct the WSS figure. Here it can be seen that the gap between 1 and 2 is larger than other number of clusters. So, it can be said that the optimal clustering is k = 1 or 2.

**Graphical user interface, chart

Description automatically generated**

**NbClust function**

NbClust provides 30 indices for calculating the number of clusters and suggests the optimum clustering strategy based on the results of adjusting the number of clusters, distance measurements, and clustering techniques. This function gives us the result according to majority rule. After applying **NbClust()** function I get the optimal clustering number is 2 and it also provides the indices that which data proposed the best number of clusters in picture 1. And it also provides graphical representation of Dindex values and second differences Dindex values in picture 2.



**Graphical user interface, application

Description automatically generated Picture 1&2**

**Different Clustering method with optimal value**

Graphical user interface, application

Description automatically generatedAfter getting optimal clustering value k=2. I applied all the plotting and clustering methods one by one. The pictures are given below.

Some noise was shown in the **clusplot ()** plotting method. here one ellipse overlaps other ellipse, the overlapping portion was the noise of the graduation rate dataset.

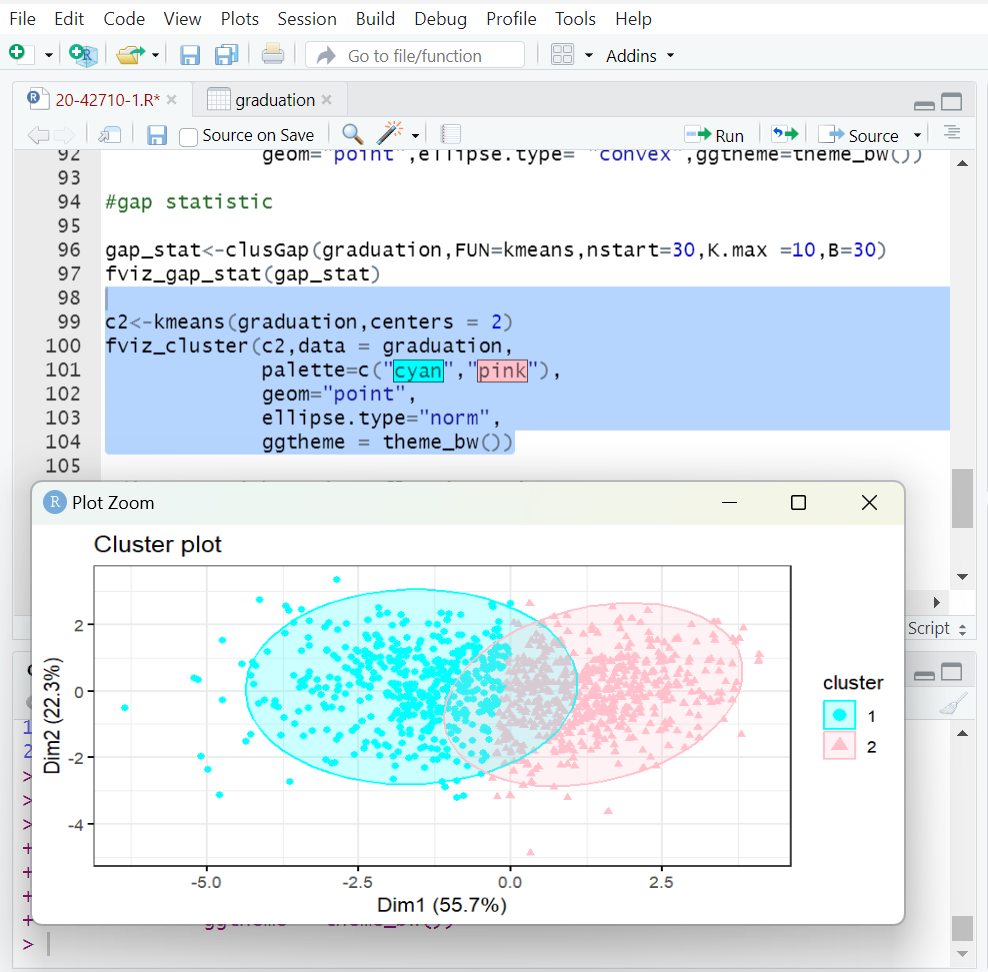
**Graphical user interface, application

Description automatically generated**

Graphical user interface

Description automatically generated

**Cluster plot with percentage**

** Gap statistic Method**

**Calculating Cluster Center & Summary of the dataset:**

We can calculate cluster center by **kmeans ()** function. After applying all the methods, I obtained the optimal clustering value k=2. So, I assume the center is 2. For center 2 our centers values are in between -1 to 1.

At last, I applied summary function to summaries my data set and check mean, median, min, max and quadrate values each of the attributes of the dataset.

**Graphical user interface, text, application

Description automatically generated**

**Center of Cluster**

**Graphical user interface, text, application

Description automatically generated**

**summary**

**Dataset link:**

1.[Graduation Rate | Kaggle](https://www.kaggle.com/datasets/rkiattisak/graduation-rate)