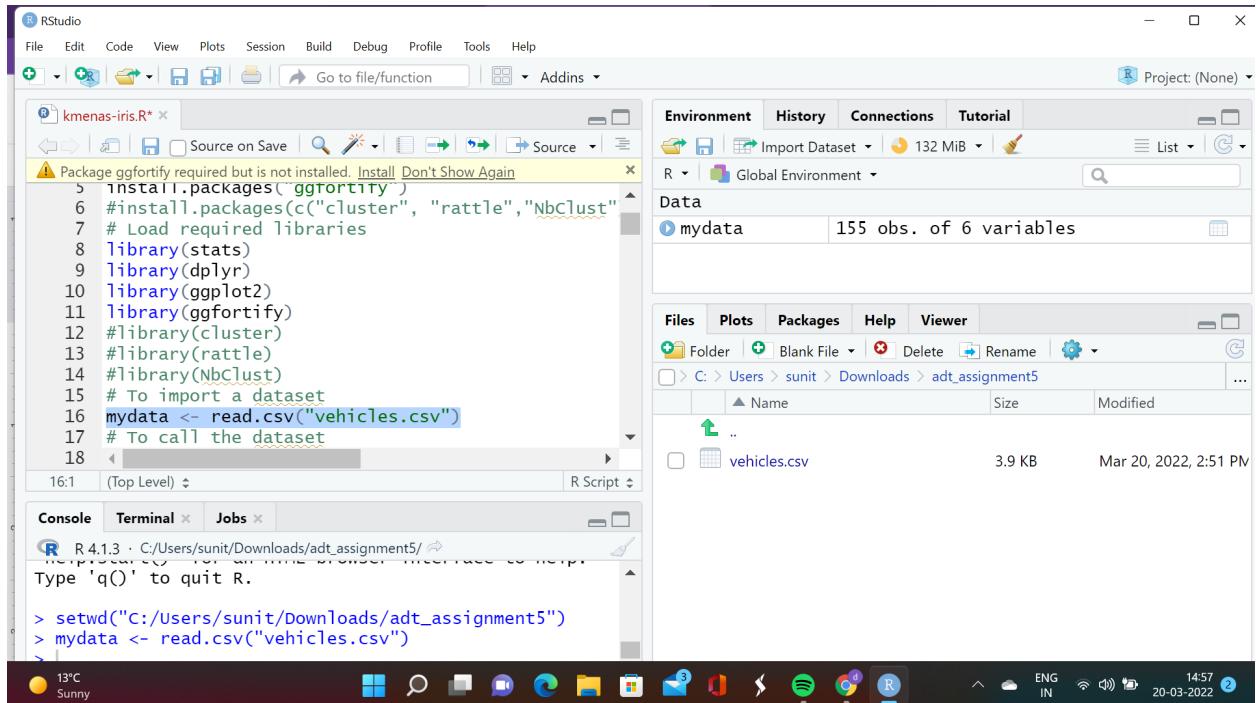


ADVANCED DATABASE TOPICS

ASSIGNMENT - 4

SUBMITTED BY: ABINAYA ELANCHEZHIAN
STUDENT ID: 110061220

-
1. Install, load the required packages, and import the data set (10 marks).



The screenshot shows the RStudio interface. The code editor pane contains the following R script:

```
install.packages("ggfortify")
#install.packages(c("cluster", "rattle", "NbClust")
# Load required libraries
library(stats)
library(dplyr)
library(ggplot2)
library(ggfortify)
#library(cluster)
#library(rattle)
#library(NbClust)
# To import a dataset
mydata <- read.csv("vehicles.csv")
# To call the dataset
```

The environment pane shows a data frame named `mydata` with 155 observations and 6 variables. The console pane shows the command `mydata <- read.csv("vehicles.csv")` being run.

Command: `mydata <- read.csv("vehicles.csv")`

Explanation: Firstly, the working directory is set. The ‘vehicles’ dataset is imported in the R studio. The environment displays that there are 155 observations of 6 variables.

The screenshot shows the RStudio interface. In the top-left pane, the script file 'kmenas-iris.R' is open, containing the following code:

```

14 # library(NCUTS)
15 # To import a dataset
16 mydata <- read.csv("vehicles.csv")
17 # To call the dataset
18 mydata <- vehicles
19 head(mydata)
20 head(mydata,15)
21 tail(mydata)
22 colnames(mydata)
23
24

```

In the bottom-left pane, the R console displays the output of the 'head()' function:

```

R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
> head(mydata)
  engine horse weight length fuelcap type
1     1.8    140   2639   172.4    13.2    0
2     3.2    225   3517   192.9    17.2    0
3     3.5    210   3850   196.6    18.0    0
4     1.8    150   2998   178.0    16.4    0
5     2.8    200   3561   192.0    18.5    0
6     4.2    310   3902   198.2    23.7    0

```

Command: head(mydata)

Explanation: The head() is used to display the first 6 rows of the ‘vehicles’ dataset. Hence, the dataset is imported successfully.

2. Upload data to R and view the structure of data, dimensionality, and attribute names (10 marks).

The screenshot shows the RStudio interface. In the top-left pane, the script file 'kmenas-iris.R' is open, containing the following code:

```

21 tail(mydata)
22 colnames(mydata)
23
24 #dataset pre-processing
25 dim(mydata)
26 summary(mydata)
27 str(mydata)
28

```

In the bottom-left pane, the R console displays the output of the 'str()' function:

```

R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
> str(mydata)
'data.frame': 155 obs. of 6 variables:
 $ engine : num  1.8 3.2 3.5 1.8 2.8 4.2 2.5 2.8 2.8 3.1 ...
 $ horse  : int  140 225 210 150 200 310 170 193 193 175 ...
 $ ...
 $ weight : int  2639 3517 3850 2998 3561 3902 3179 3197 3472 3368 ...
 $ length : num  172 193 197 178 192 ...
 $ fuelcap: num  13.2 17.2 18 16.4 18.5 23.7 16.6 16.6 18.5 17.5 ...
 $ type   : int  0 0 0 0 0 0 0 0 0 0 ...

```

Command: str(mydata)

Explanation: The command compactly displays the internal structure of each of the attributes in the ‘vehicles’ dataset.

RStudio Environment Tab Screenshot:

```

21: tail(mydata)
22: colnames(mydata)
23:
24: #dataset pre-processing
25: dim(mydata)
26: summary(mydata)
27: str(mydata)
28: # To change the class type of the class variable
29: mydata$outcome <- as.factor(mydata$outcome)
30: names(mydata)
31: View(mydata)
32: class(mydata)
33: # Find columns with missing values
34: colSums(is.na(mydata))
35: 
```

Console Output:

```

R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
8.5 17.5 ...
$ type   : int  0 0 0 0 0 0 0 0 0 0 ...
> dim(mydata)
[1] 155  6

```

Command: dim(mydata)

Explanation: Displays the number of observations(rows) and variables(columns) available in the dataset. In the ‘vehicles’ dataset, there are 155 rows and 6 columns.

RStudio Environment Tab Screenshot:

```

21: tail(mydata)
22: colnames(mydata)
23:
24: #dataset pre-processing
25: dim(mydata)
26: summary(mydata)
27: str(mydata)
28: # To change the class type of the class variable
29: mydata$outcome <- as.factor(mydata$outcome)
30: names(mydata)
31: View(mydata)
32: class(mydata)
33: # Find columns with missing values
34: colSums(is.na(mydata))
35: 
```

Console Output:

```

R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
[1] 155  6
> colnames(mydata)
[1] "engine"    "horse"     "weight"    "length"   "fuelcap"
[6] "type"

```

Command: colnames(mydata)

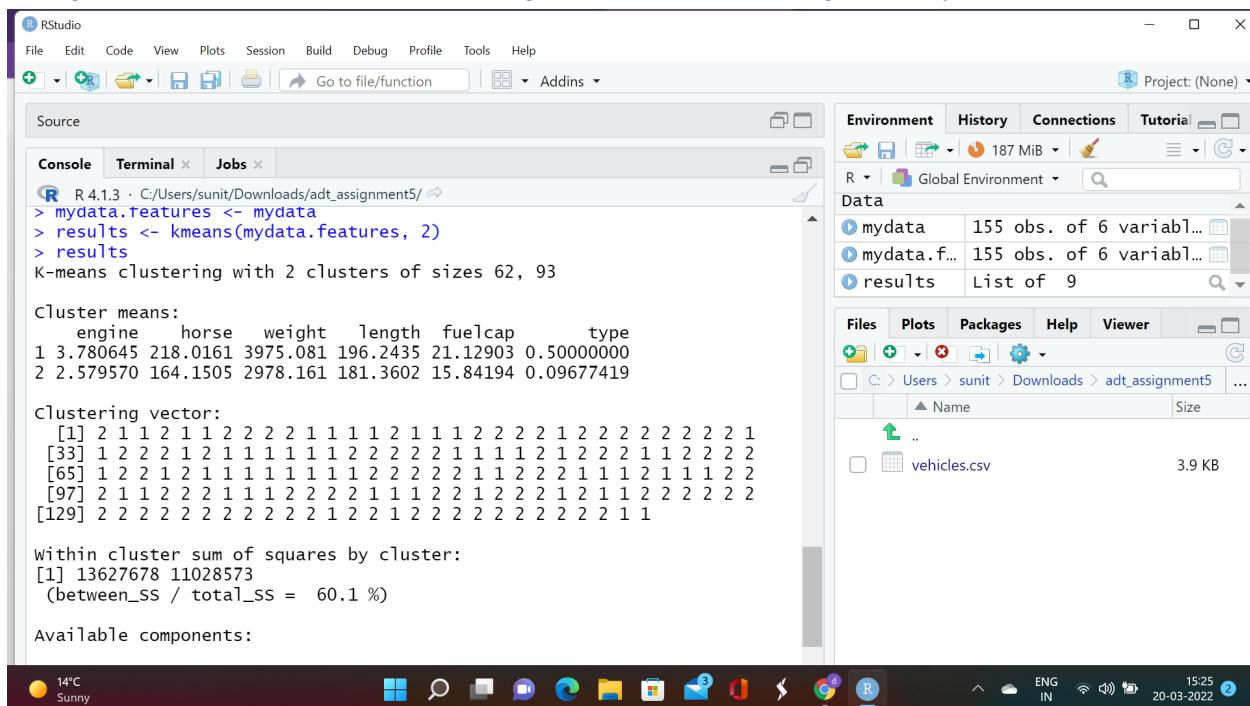
Explanation: The columns available in the ‘vehicles’ dataset are displayed. They are engine, horse, weight, length, fuelcap and type.

3. Explain what attributes you are using for clustering and why? (20 marks).

In simpler terms, K-means clustering minimizes the distance within a cluster and maximizes the distance between clusters. So the ideal combination of attributes will have two(considering k=2) distinct clusters that are apart.

Further, not every attribute would affect the cluster equally. Some attributes may be totally unrelated and may not add any difference to the clusters formed. As this dataset is completely new, the only way to know for sure the best combination of attributes to choose is through trying out various combinations. Once various clusters with various combinations are made, the ‘Dunn index’ can be used to evaluate and determine the best clusters.

To begin with, the attributes ‘horse’ and ‘length’ are used for clustering and analysis.



The screenshot shows the RStudio interface with the R console tab selected. The console output displays the following R code and its execution results:

```
R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
> mydata.features <- mydata
> results <- kmeans(mydata.features, 2)
> results
K-means clustering with 2 clusters of sizes 62, 93

Cluster means:
  engine    horse   weight   length  fuelcap      type
1 3.780645 218.0161 3975.081 196.2435 21.12903 0.50000000
2 2.579570 164.1505 2978.161 181.3602 15.84194 0.09677419

Clustering vector:
 [1] 2 1 1 2 1 1 2 2 2 1 1 1 1 2 1 1 1 2 2 2 2 1 2 2 2 2 2 2 2 1
[33] 1 2 2 2 1 2 1 1 1 1 1 2 2 2 2 1 1 1 2 1 2 2 2 1 1 2 2 2 2 2 2
[65] 1 2 2 2 1 2 1 1 1 1 1 1 2 2 2 2 2 1 1 2 2 2 1 1 1 2 1 1 1 2 2 2
[97] 2 1 1 2 2 2 1 1 1 2 2 2 2 1 1 1 2 2 1 2 2 2 1 2 1 1 2 2 2 2 2 2
[129] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1

within cluster sum of squares by cluster:
[1] 13627678 11028573
(between_SS / total_SS =  60.1 %)

Available components:
```

Command: mydata.features <- mydata

```
results <- kmeans(mydata.features, 2)
results
```

Explanation: A copy of mydata is stored in mydata.features and kmeans clustering is performed on mydata.features and this stored in results. In kmeans(mydata,2), the 2 denotes the number of clusters to be formed. The results displays information about the size of the clusters, clustering vector, cluster means, etc.

RStudio Environment pane showing available components:

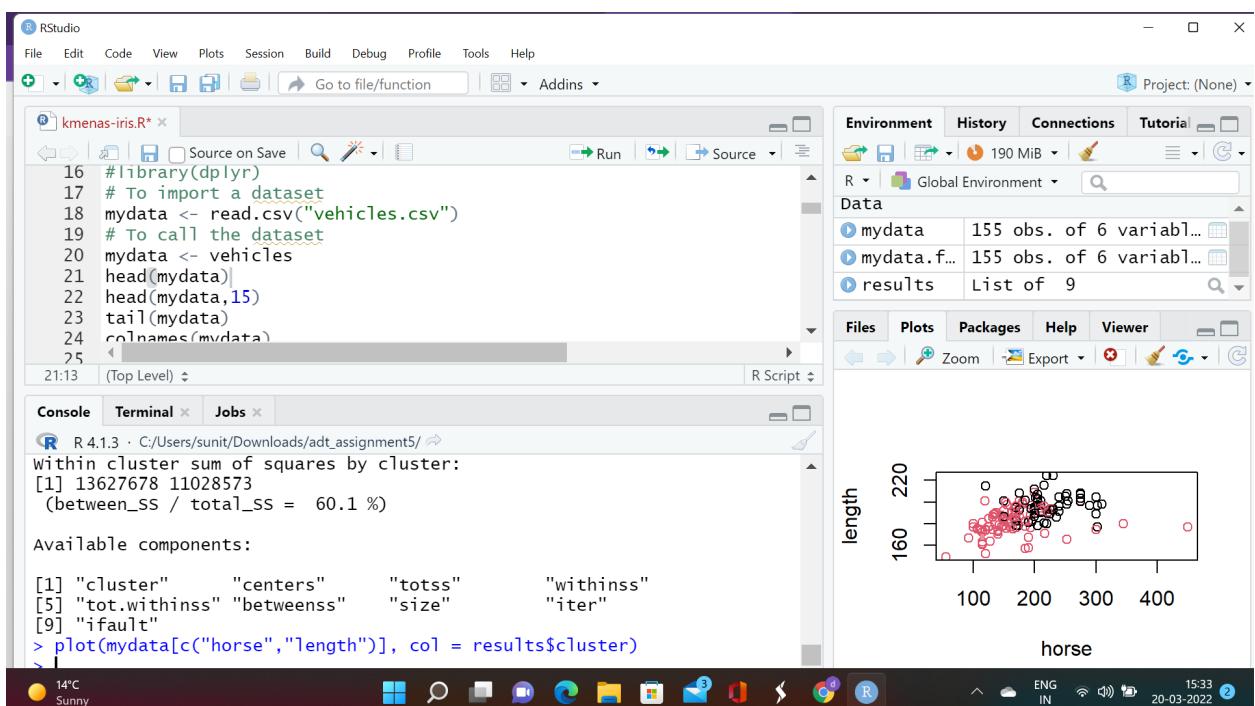
```

[1] "cluster"      "centers"       "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"

```

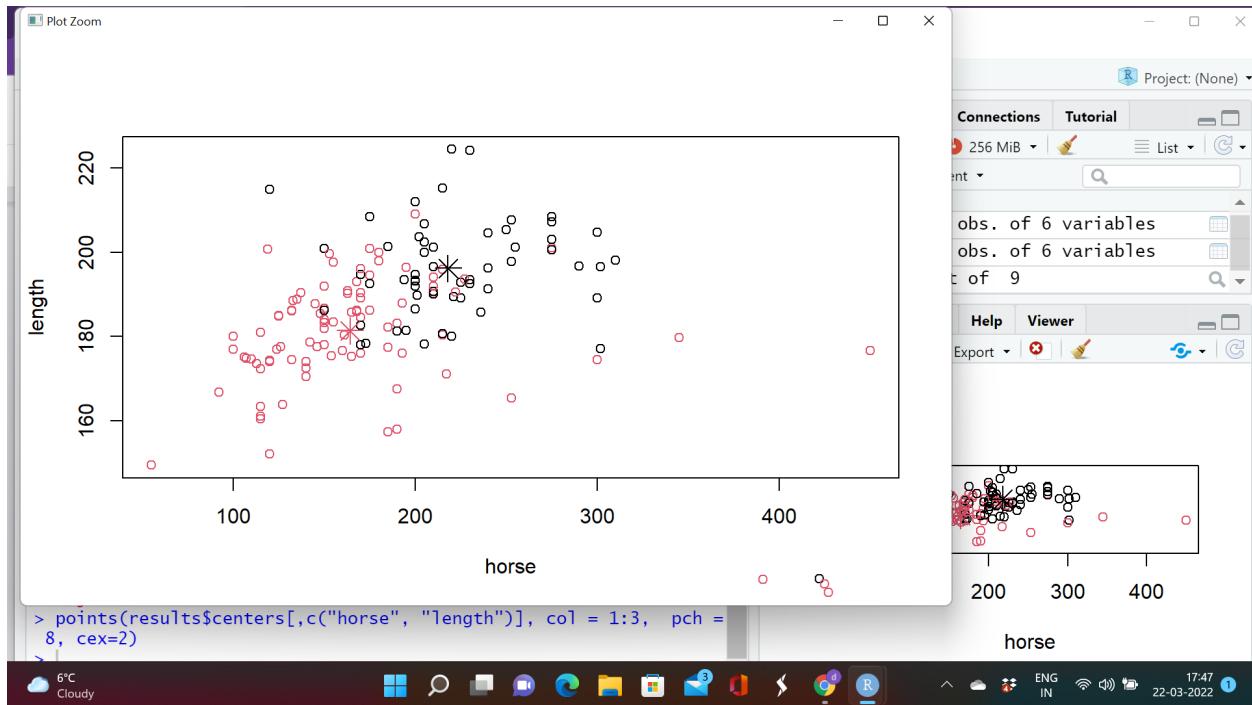
RStudio Files pane showing the vehicles.csv file.

Explanation: The various components available in the cluster are displayed. The clusters can be differentiated into different colors by using any of these components.



Command: `plot(mydata[c("horse", "length")], col = results$cluster)`

Explanation: The cluster is plotted by choosing the horse and length attributes. The clusters are differentiated in different colors using the 'col' command.



Command: `points(results$centers[,c("horse", "length")], col = 1:3, pch = 8, cex=2)`

Explanation: The centroids of both the clusters are plotted.

4. Apply clustering on the dataset for several K values, e.g., 3, 4 and 5 (10 marks).

Command: `results <- kmeans(mydata.features, k)` where k can be 3,4,5, etc.

results

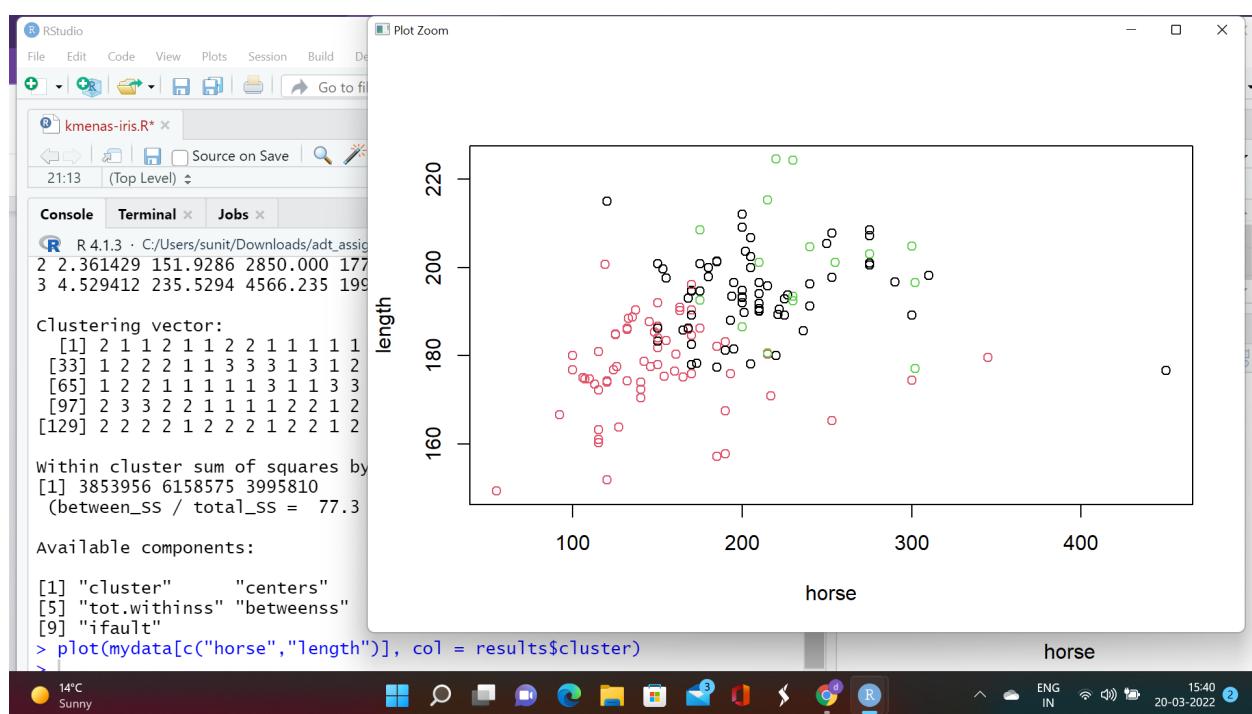
`plot(mydata[c("horse", "length")], col = results$cluster)`

Explanation: As similar to question 3, k-means clustering is performed on mydata.features for different k values where k denotes the number of clusters to be formed. The results are displayed which displays various information like cluster sizes, cluster means, cluster vectors and available components. The `plot()` is used to plot the clusters and the clusters are differentiated in various colors.

(i) K = 3:

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Code Editor:** A file named "kmeans-iris.R" is open. The code runs `kmeans` on the "iris" dataset with 3 clusters, resulting in cluster means and a clustering vector.
- Console:** Displays the output of the R code, including the cluster means for three clusters and the clustering vector for each sample.
- Data View:** Shows the "mydata" dataset with 155 observations and 6 variables, and the "results" object as a list of 9 items.
- File Explorer:** Shows a project directory structure under "adt_assignment5" containing a "vehicles.csv" file (3.9 KB).
- Bottom Status Bar:** Shows the date (20-03-2022), time (15:36), battery level (14%), and system status (ENG IN).



(ii) $K = 4$:

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

kmenas-iris.R* 21:13 (Top Level)

Console Terminal Jobs

```
R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
> results <- kmeans(mydata.features, 4)
> results
K-means clustering with 4 clusters of sizes 67, 39, 44, 5

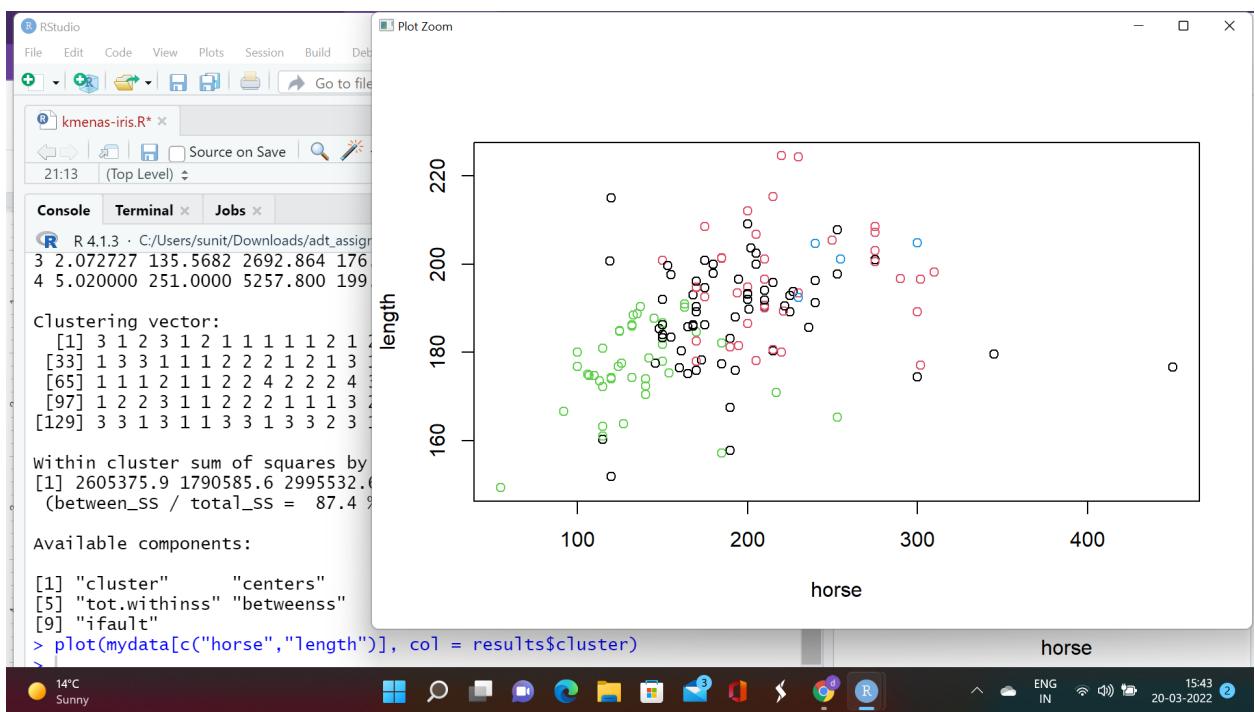
Cluster means:
  engine horse weight length fuelcap type
1 3.058209 193.7164 3321.104 188.1642 17.51493 0.16417910
2 3.925641 220.1026 4003.462 196.6718 21.56667 0.56410256
3 2.072727 135.5682 2692.864 176.3818 14.36136 0.04545455
4 5.020000 251.0000 5257.800 199.1200 27.36000 1.00000000

Clustering vector:
 [1] 3 1 2 3 1 2 1 1 1 2 1 2 1 2 2 4 3 1 1 1 1 3 3 1 3 1 1 3 1
[33] 1 3 3 1 1 1 2 2 2 1 2 1 3 1 3 1 2 2 4 1 2 3 3 1 2 2 3 3 1 1
[65] 1 1 1 2 1 1 2 2 4 2 2 4 3 3 3 1 1 2 1 3 3 1 2 2 2 1 2 2 2 1 3
[97] 1 2 2 3 1 1 2 2 2 1 1 1 3 2 2 3 3 1 3 3 1 1 1 2 3 1 1 1 3 3
[129] 3 3 1 3 1 1 3 3 1 3 3 2 3 1 4 3 3 1 1 3 3 3 1 1 1 1 1 1 1 1

within cluster sum of squares by cluster:
[1] 2605375.9 1790585.6 2995532.6 363817.5
(between_SS / total_SS =  87.4 %)

14°C Sunny 15:42 20-03-2022
```

Environment History Connections Tutorial
Data
mydata 155 obs. of 6 variables
mydata.f... 155 obs. of 6 variables
results List of 9
Files Plots Packages Help Viewer
C:\> Users > sunit > Downloads > adt_assignment5 ...
Name Size
..
vehicles.csv 3.9 KB



(iii) K = 5:

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins

kmenas-iris.R* 21:13 (Top Level)

Console Terminal Jobs

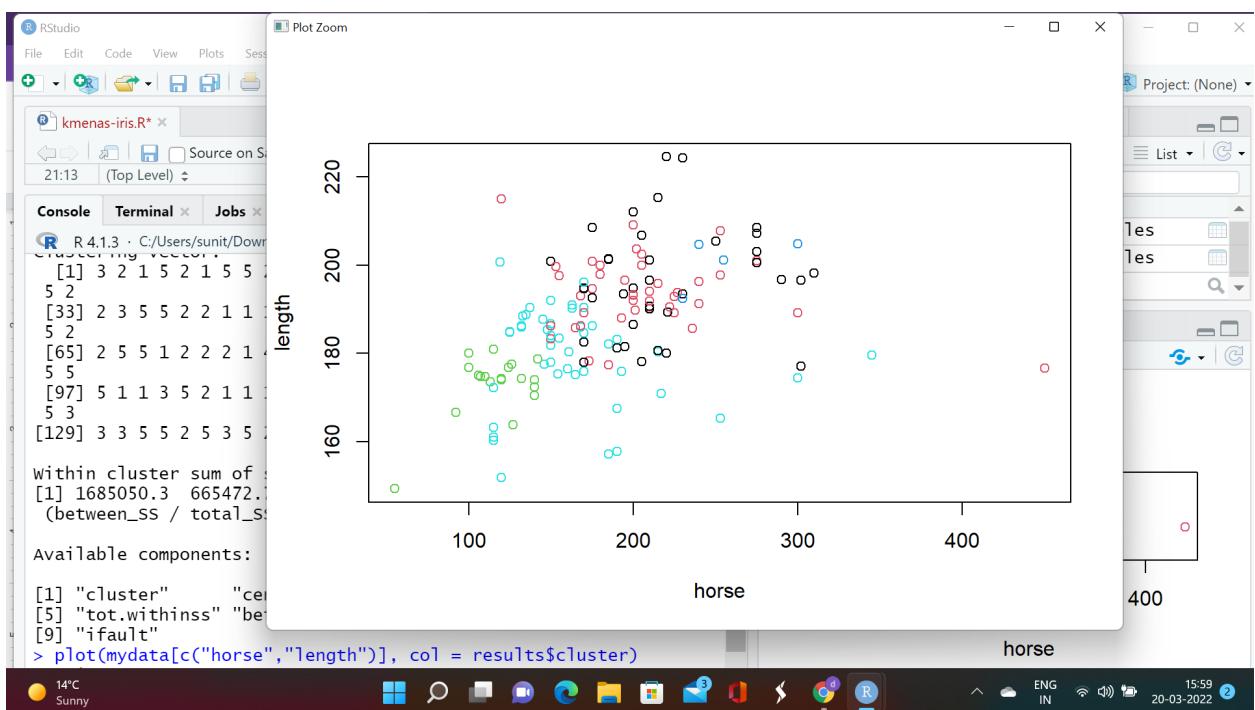
```
R 4.1.3 · C:/Users/sunit/Downloads/adt_assignment5/
> results <- kmeans(mydata.features, 5)
> results
K-means clustering with 5 clusters of sizes 38, 42, 20, 5, 50

Cluster means:
  engine horse weight length fuelcap type
1 3.923684 218.0000 4011.632 196.8684 21.61316 0.5789474
2 3.209524 204.9762 3456.976 193.4214 18.05238 0.1190476
3 1.860000 117.0500 2452.900 173.1500 13.05000 0.1000000
4 5.020000 251.0000 5257.800 199.1200 27.36000 1.0000000
5 2.562000 165.8800 3008.840 179.4060 16.12000 0.1200000

Clustering vector:
 [1] 3 2 1 5 2 1 5 5 2 2 2 1 2 1 2 1 1 4 3 5 2 2 2 5 3 3 2 5 2 2 5 2
[33] 2 3 5 5 2 2 1 1 2 1 2 3 5 5 2 3 1 1 1 4 5 1 3 5 5 1 1 3 3 5 2
[65] 2 5 5 1 2 2 2 1 4 1 1 1 4 3 5 5 2 5 1 2 5 5 2 1 1 2 1 1 1 5 5
[97] 5 1 1 3 5 2 1 1 5 5 2 5 1 1 1 3 5 2 5 5 5 2 2 2 1 5 5 5 2 5 3
[129] 3 3 5 5 2 5 3 5 2 3 3 1 3 2 4 5 5 5 5 5 5 2 2 2

within cluster sum of squares by cluster:
[1] 1685050.3 665472.7 680039.4 363817.5 1028496.7
```

14°C Sunny 15:58 20-03-2022

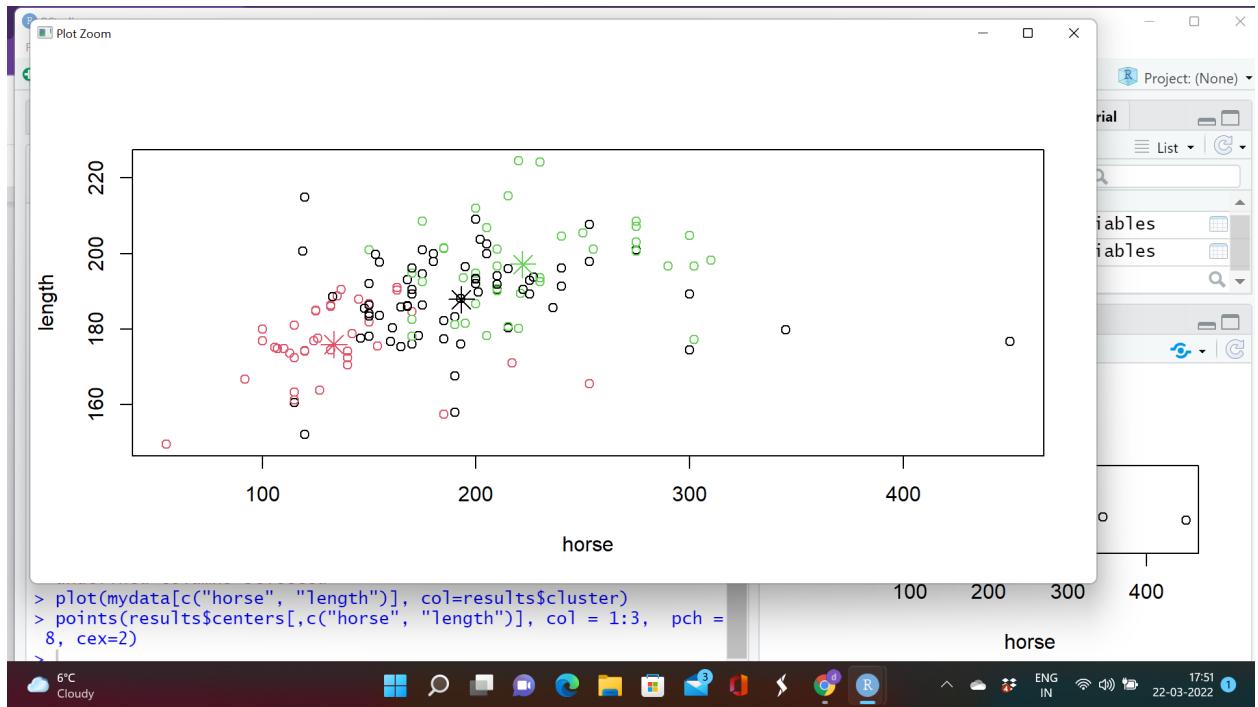


5. Plot the model and show the centroids for each K selected in the previous question (10 marks).

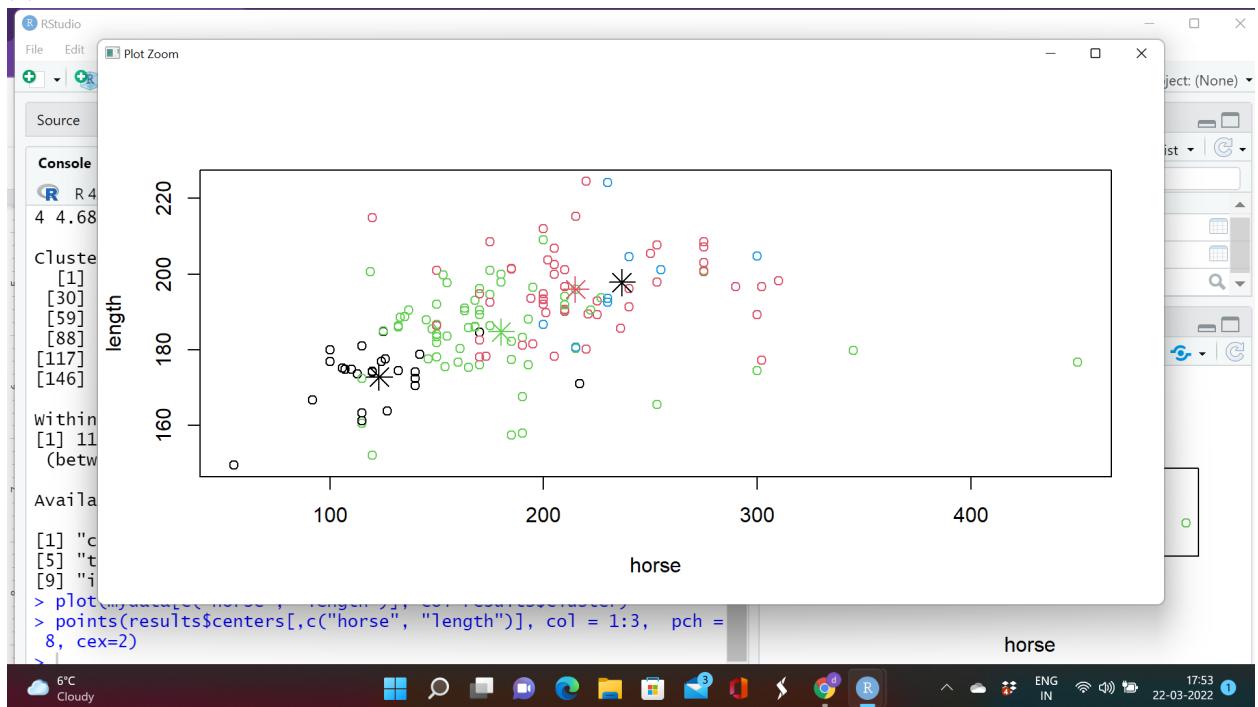
Command: points(results\$centers[,c("horse", "length")], col = 1:3, pch = 8, cex = 2)

Explanation: The centroids of all the clusters are plotted. They are denoted by * symbol.

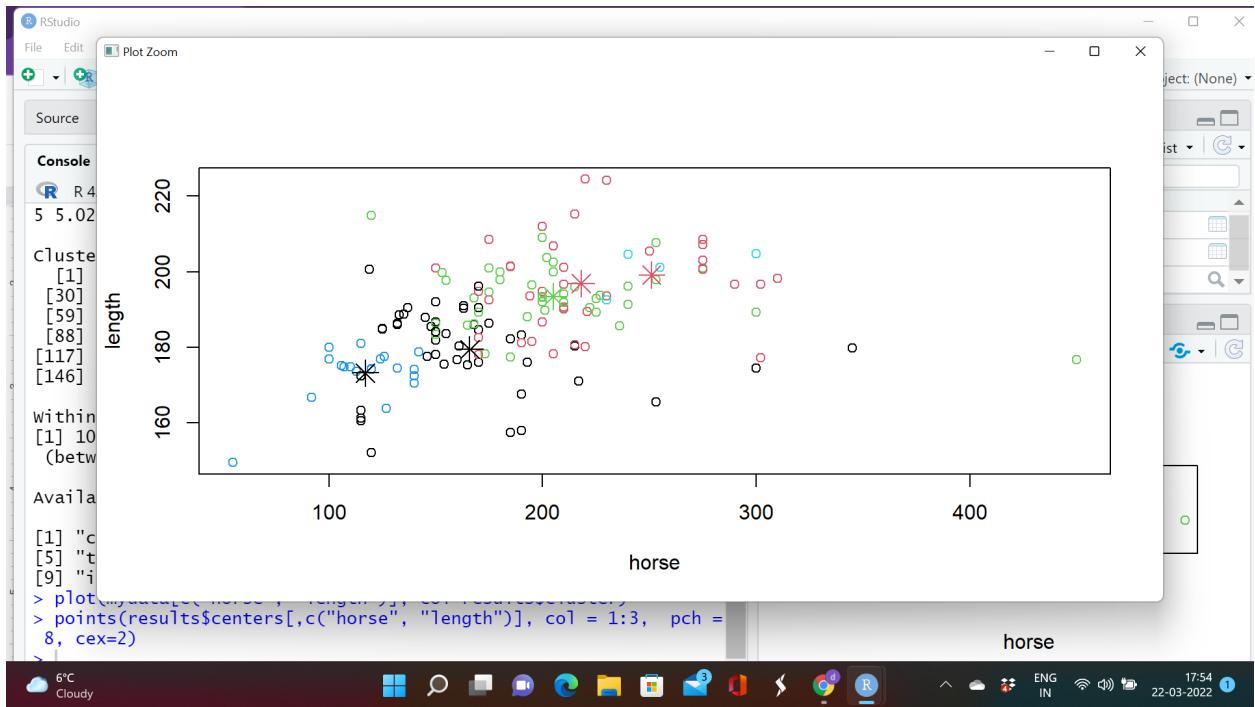
(i) K=3:



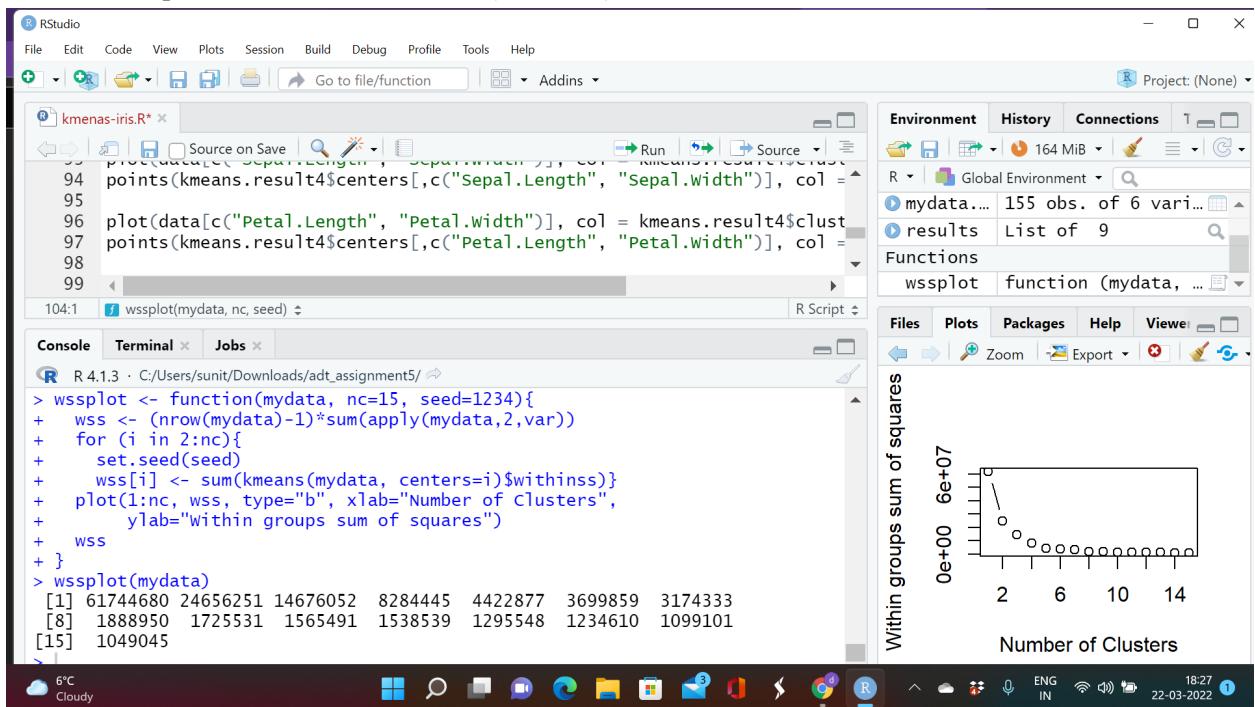
(ii) K=4:



(iii)K=5:



6. Show the optimum number of clusters (20 marks).



```

Command: wssplot <- function(mydata, nc=15, seed=1234){
  wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(mydata, centers=i)$withinss)}
}

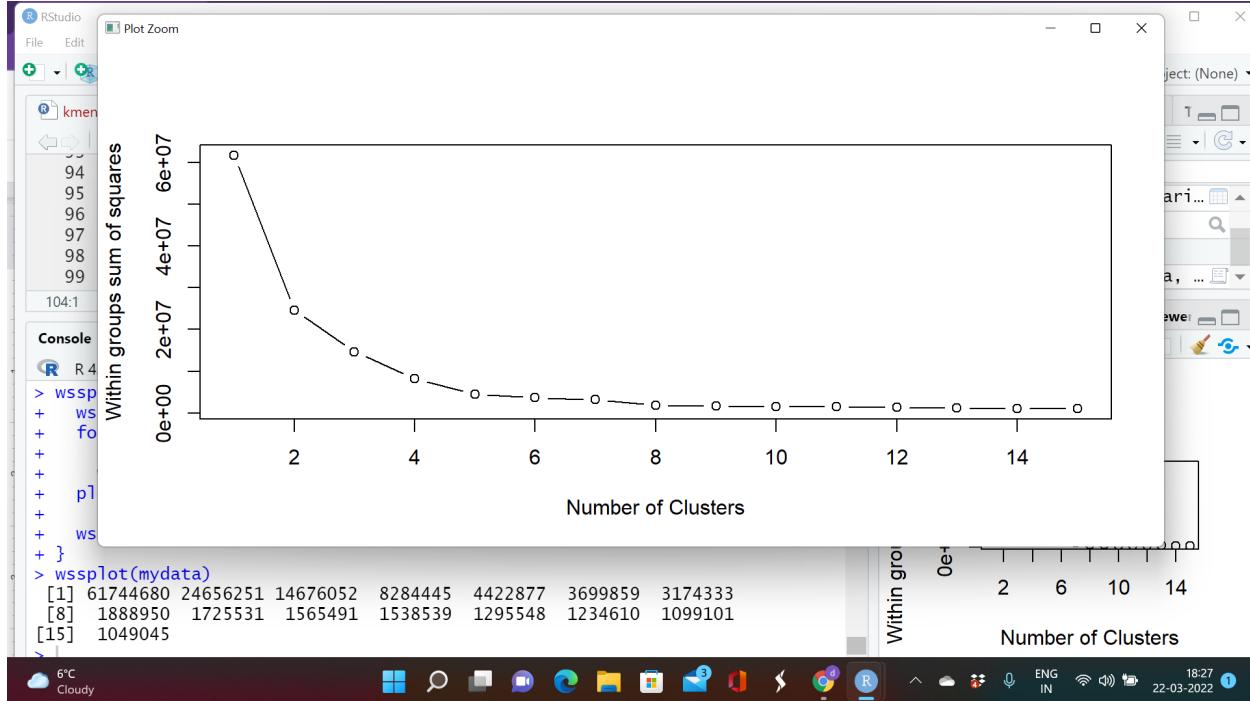
```

```

plot(1:nc, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
wss
}
wssplot(mydata)

```

Explanation: The command displays the optimum number of clusters for the ‘vehicles’ dataset. It is an elbow curve and the value at the elbow displays the optimum number of clusters that can be formed.



7. Select different attributes and explain which ones show good clusters (20 marks).

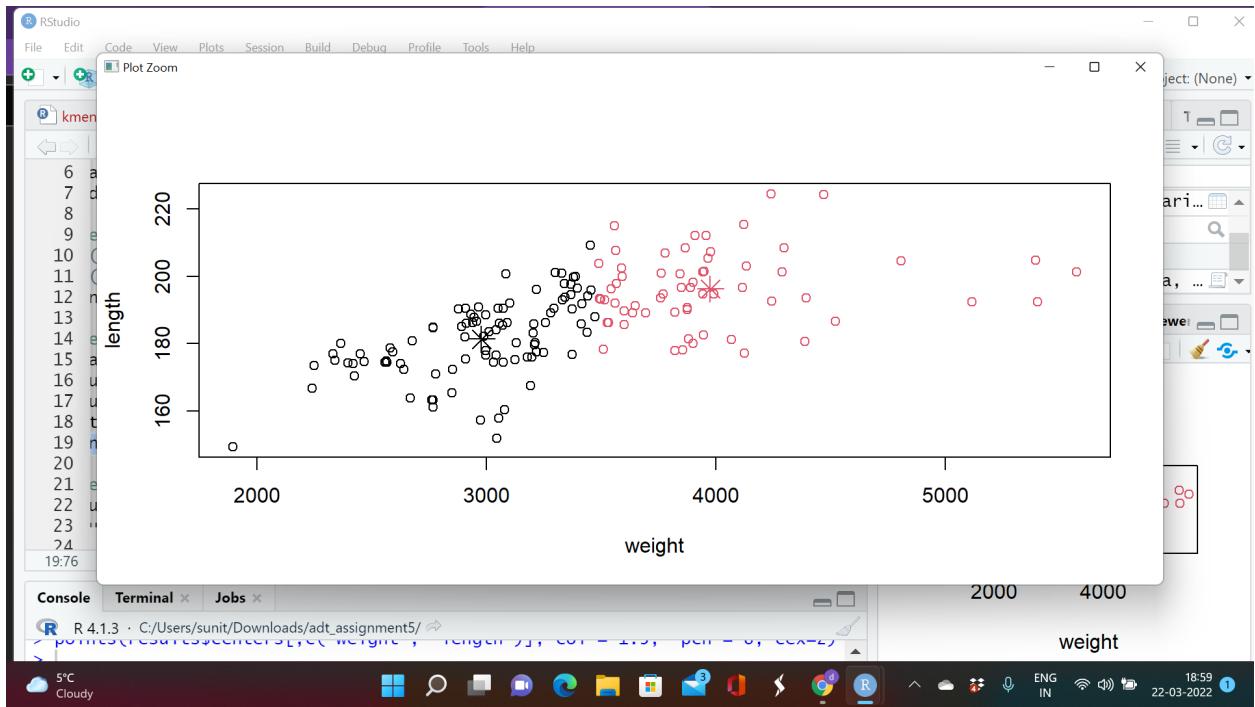


Command: `plot(mydata, col = results$cluster)`

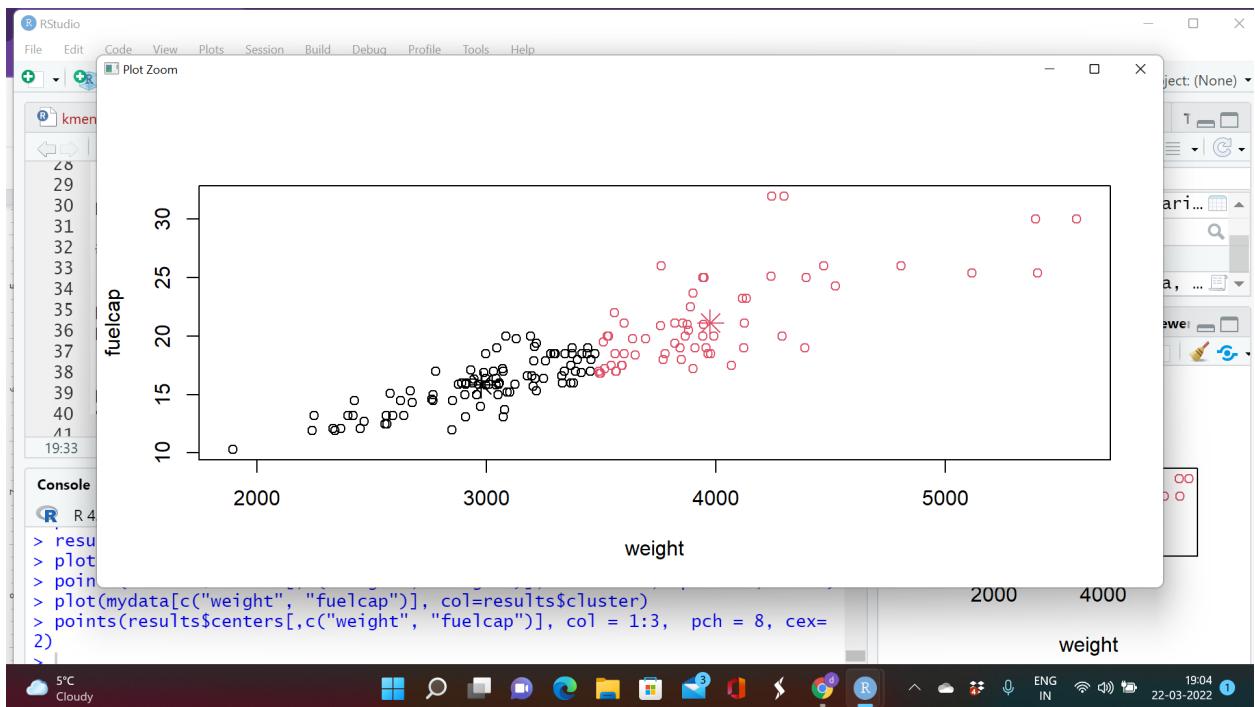
Explanation: The command plots clusters with various combinations of the available attributes.

In my opinion, good cluster combinations are shown by (weight, length) and (weight, fuelcap) combinations. The reasons are:

1. Distance inside the cluster should be minimal and distance between clusters should be maximum. Among the available cluster combinations, most combinations have overlapping clusters. So these clusters are not ideal.
2. From the other available cluster combinations, the clusters with well different centroids that are distinct and far from each other are chosen as ideal.



Explanation: K-means clustering for weight and length attributes from the ‘vehicles’ dataset.



Explanation: K-means clustering for weight and fuelcap attributes from the ‘vehicles’ dataset.