# Note:

# \* Solutions to all the sections of the Week# assignment is consolidated

# Question 4.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.

# Why this use case?

I have always a great appreciation for Railroad sector, not just for historically influencing the growth and extending our reach to unexplored territories, but also in our current day to day life by moving tons of freight and meeting our necessities in a fuel-efficient way.

# Background:

This use case is associated in clustering the inbound and outbound Freight cars in a Rail yard.

A   railroad yard is a complex series of [railroad tracks](https://en.wikipedia.org/wiki/Track_(rail_transport)) for storing, sorting, or loading/unloading [railroad cars](https://en.wikipedia.org/wiki/Railroad_car) and [locomotives](https://en.wikipedia.org/wiki/Locomotives).

There are several categories of cars, say Reefers (refrigerated box cars to carry Perishable foods), tank cars (chemicals, molasses, edible tallow, water, and diesel fuel), well cars (to carry containers) and more…

# Business value:

As there is direct association of the cars category to the kind of commodity it can carry, it becomes crucial from business perspective to get visibility of trending commodities when we can cluster the railcars into its respective category. For sake of simplicity, we are considering only the railcars and associate them to the respective clusters based on one or more predictors

# Planned Approach:

This specific use case can be done with a supervised classification or an unsupervised clustering model.

I am using unsupervised K-means to support this use case due to the below reasons

1. In rail industry, it is common to update & reuse an existing car to support an entire different set of commodity. Say, when box cars was yielding less revenue, it was updated to be used as reefer with added temperature control to support perishables goods transport. In such scenarios, we will **not** have historical results to use a supervised model.
2. K-means performance advantage is a perfect fit when dealing with the high data volume as we see in the yard(s).

# Predictors

|  |  |  |  |
| --- | --- | --- | --- |
| **Predictors** | **Data type** | **Sample values** | **Comments** |
| Enclosed Flag | Boolean | Y/N | Flat cars used to carry lumber, logs are open design ; while tank cars that carry liquid are closed |
| Cars Dimension |  |  | Car dimensions are crucial factors in identifying the kind of the cars |
| Height | Number(nominal) | values in lbs. |  |
| Weight(unloaded) | Number(nominal) | values in lbs. |  |
| Capacity | Number(nominal) | values in tons |  |
| Weight(loaded) | Number(nominal) | values in lbs. | The range of weight when the cars is loaded with commodity |
| Location | Characters | e.g : Chicago, IL Atlanta, GA | specific locations are assigned to certain commodity transportation and this can be key to clustering the cars |
| Construct | Characters | e.g :Metal | Auto racks that carry cars are made of metal |
| Temperature Controlled? | Boolean | Y/N | Reefers are temperature controlled while open cars are not. |
| Shape | Characters | e.g : Coils, Flat, Cylinder, well etc. |  |

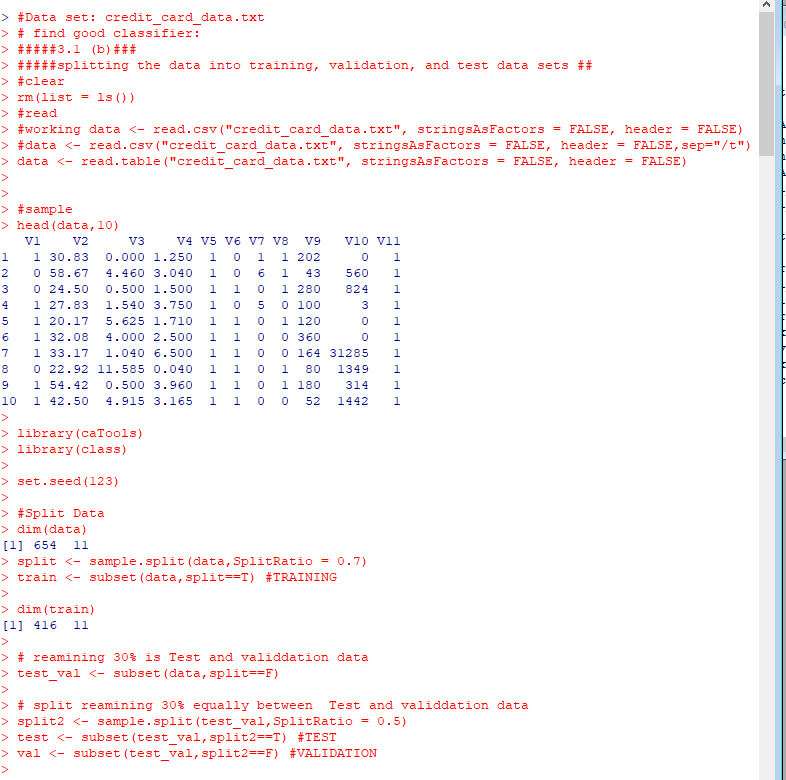
# Question 3.1

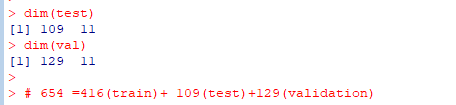
Using the same data set (credit\_card\_data.txt or credit\_card\_data-headers.txt) as in Question 2.2, use the ksvm or kknn function to find a good classifier:

1. using cross-validation (do this for the k-nearest-neighbors model; SVM is optional); and
2. splitting the data into training, validation, and test data sets (pick either KNN or SVM; the other is optional).

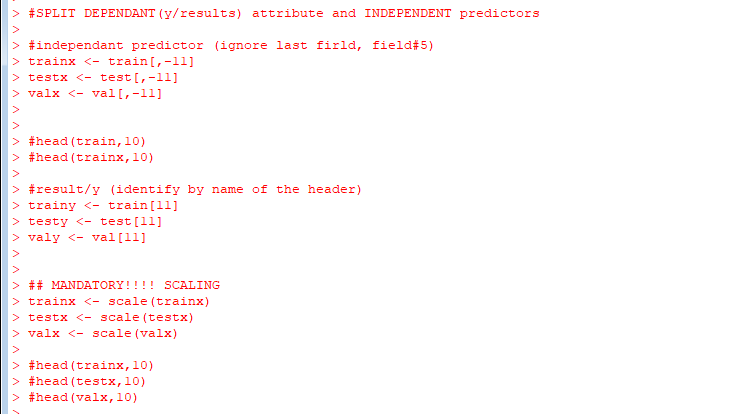
# Question 3.1.b Splitting data into training, validation and test data sets

Below screenshot covers: Ingest dataset; import library; Split data set (70% training; 15% each for test and validation)

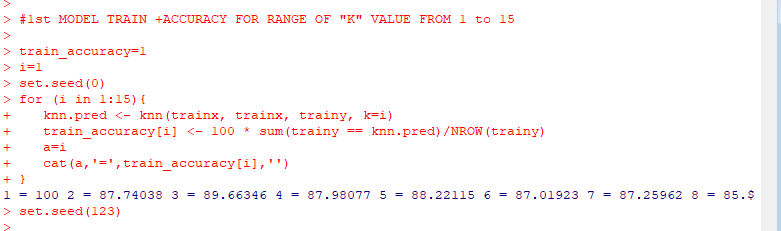




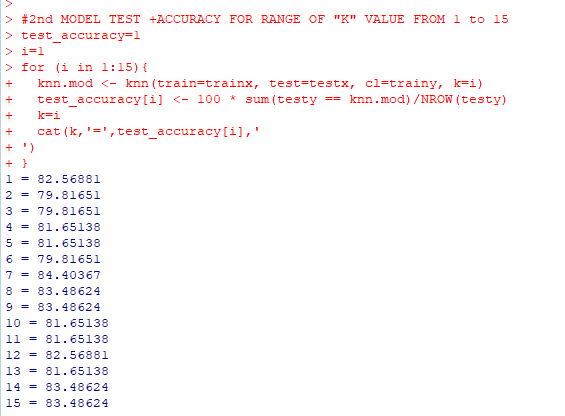
## Split Independent(x) and dependent(y) variables



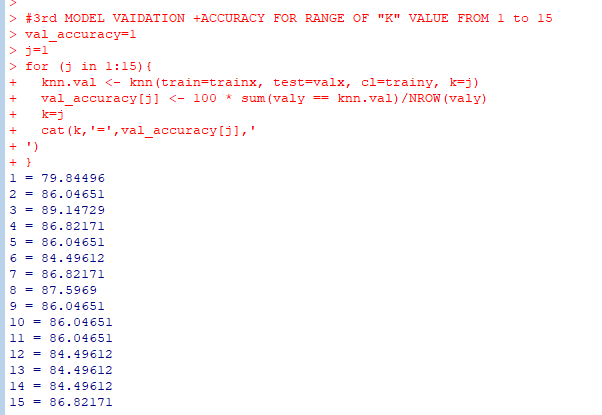
Create training model and Accuracy



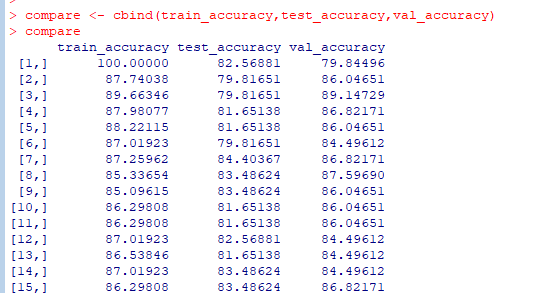
Model#2: Test model and accuracy



## Model#3 validation model and its accuracy



Comparing all the 3 model’ accuracy



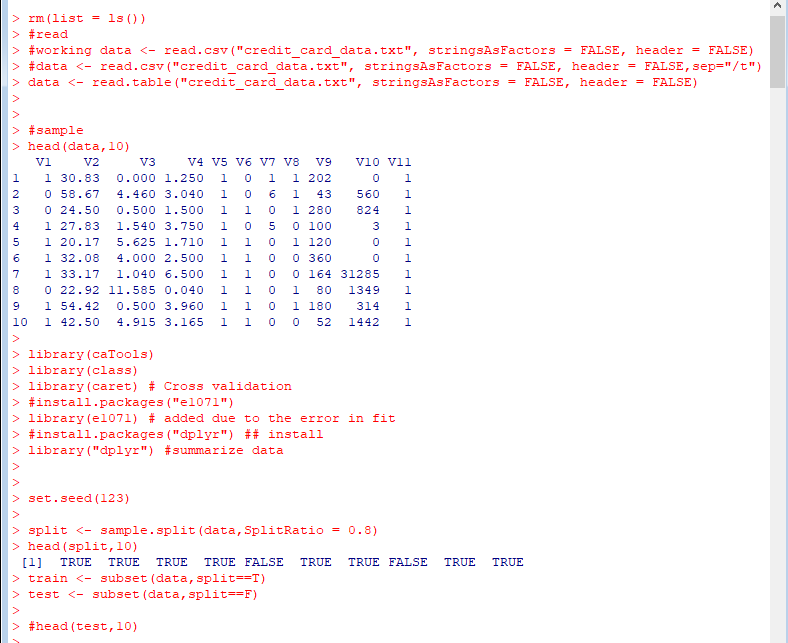
**Results: K=3 is the best fitted accurate model (k=1 is not considered due to overfitting)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **K** | **train\_accuracy** | **test\_accuracy** | **val\_accuracy** | **Average Accuracy** |  |
| 6 | 87.01923 | 79.81651 | 84.49612 | 83.77729 |  |
| 13 | 86.53846 | 81.65138 | 84.49612 | 84.22865 |  |
| 2 | 87.74038 | 79.81651 | 86.04651 | 84.53447 |  |
| 10 | 86.29808 | 81.65138 | 86.04651 | 84.66532 |  |
| 11 | 86.29808 | 81.65138 | 86.04651 | 84.66532 |  |
| 12 | 87.01923 | 82.56881 | 84.49612 | 84.69472 |  |
| 9 | 85.09615 | 83.48624 | 86.04651 | 84.8763 |  |
| 14 | 87.01923 | 83.48624 | 84.49612 | 85.00053 |  |
| 5 | 88.22115 | 81.65138 | 86.04651 | 85.30635 |  |
| 8 | 85.33654 | 83.48624 | 87.5969 | 85.47323 |  |
| 4 | 87.98077 | 81.65138 | 86.82171 | 85.48462 |  |
| 15 | 86.29808 | 83.48624 | 86.82171 | 85.53534 |  |
| 7 | 87.25962 | 84.40367 | 86.82171 | 86.16167 |  |
| 3 | 89.66346 | 79.81651 | 89.14729 | 86.20909 |  |
| 1 | 100 | 82.56881 | 79.84496 | 87.47126 | **Over fitted model** |

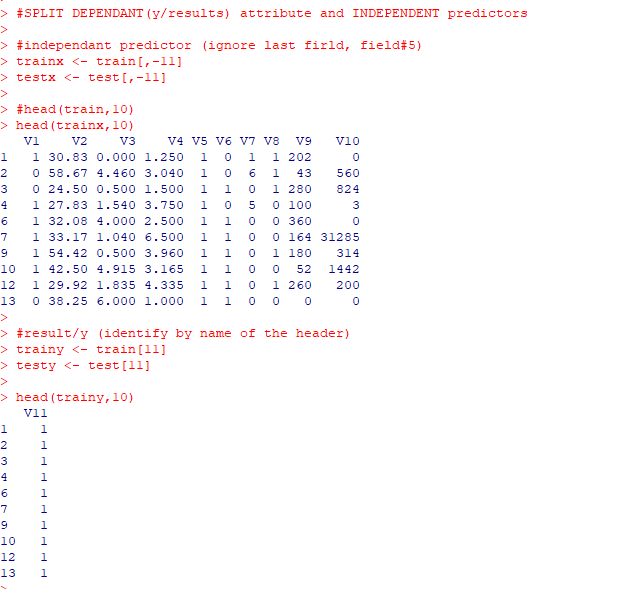
# Question 3.1.a use cross-validation (do this for the k-nearest-neighbors model; SVM is optional)

Screenshots includes the below steps:

1. Clear screen
2. Ingest Data
3. Import libraries
4. Split data into 80% training for k-fold and remaining 20% for test data



Steps of below screenshot

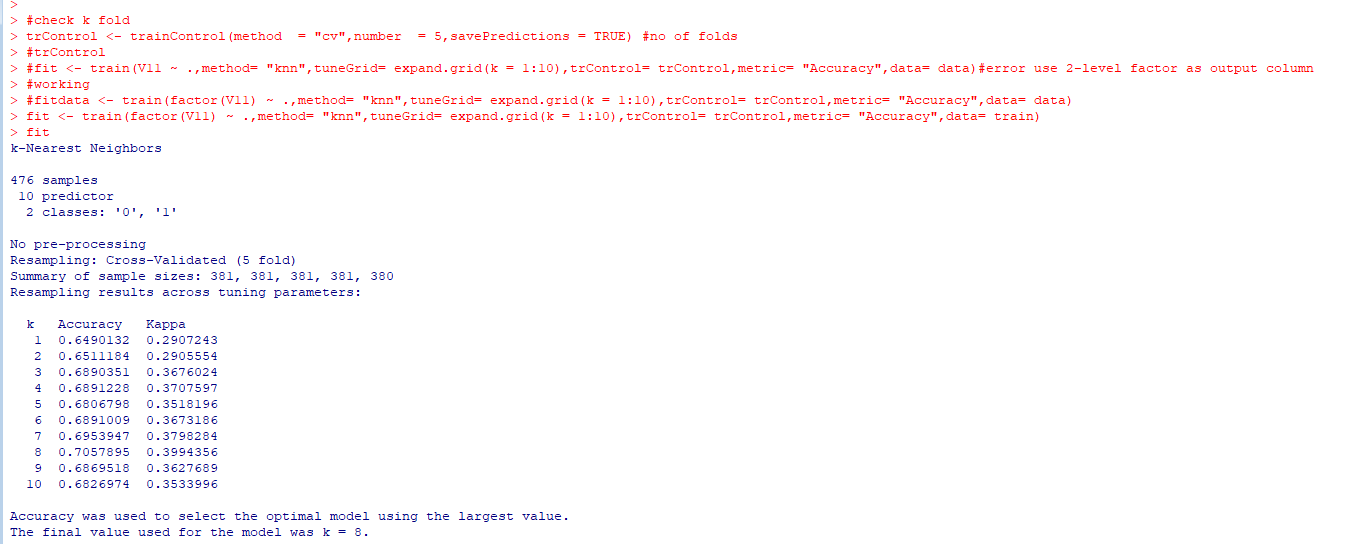
1. Separate variables for predictors (x) and results (y)
2. 

Below screenshot includes

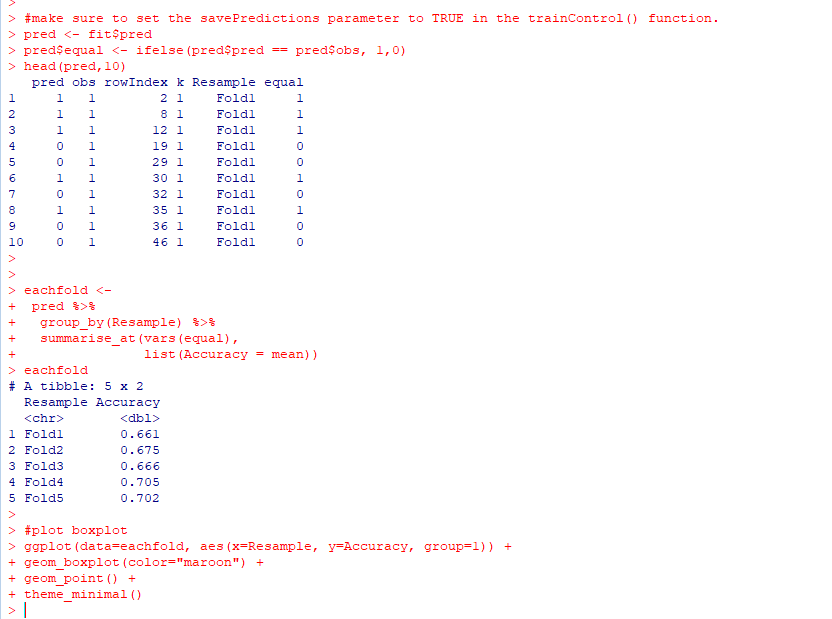


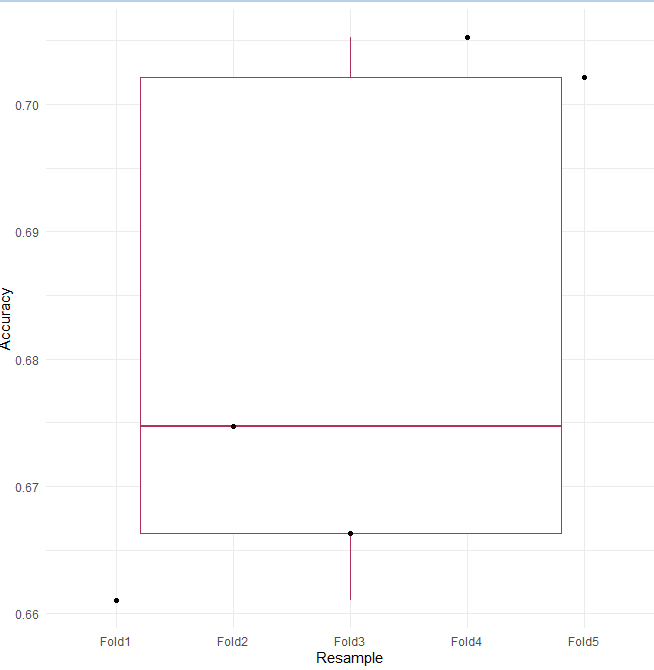
Establish K fold on training set.

After the model is fit, the highest accuracy was observed in k value = 8



Obtained the “pred” objects of the model and summarized to see the accuracy for each fold.





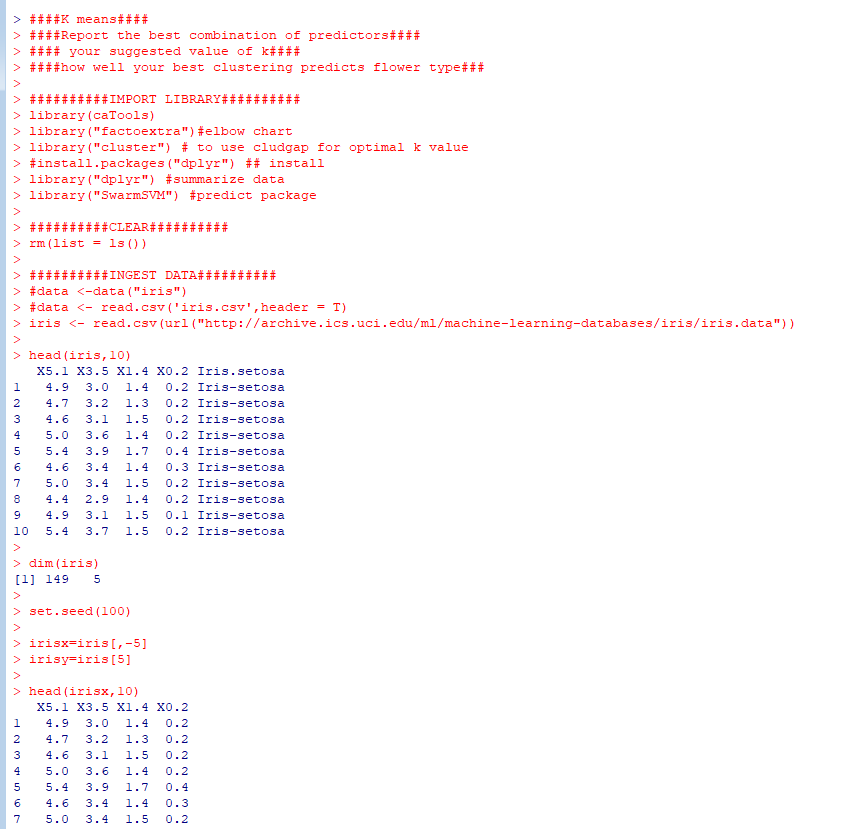
# Question 4.2

The *iris* data set iris.txt contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/Iris> ). *The response values are only given to see how well a specific method performed and should not be used to build the model.*

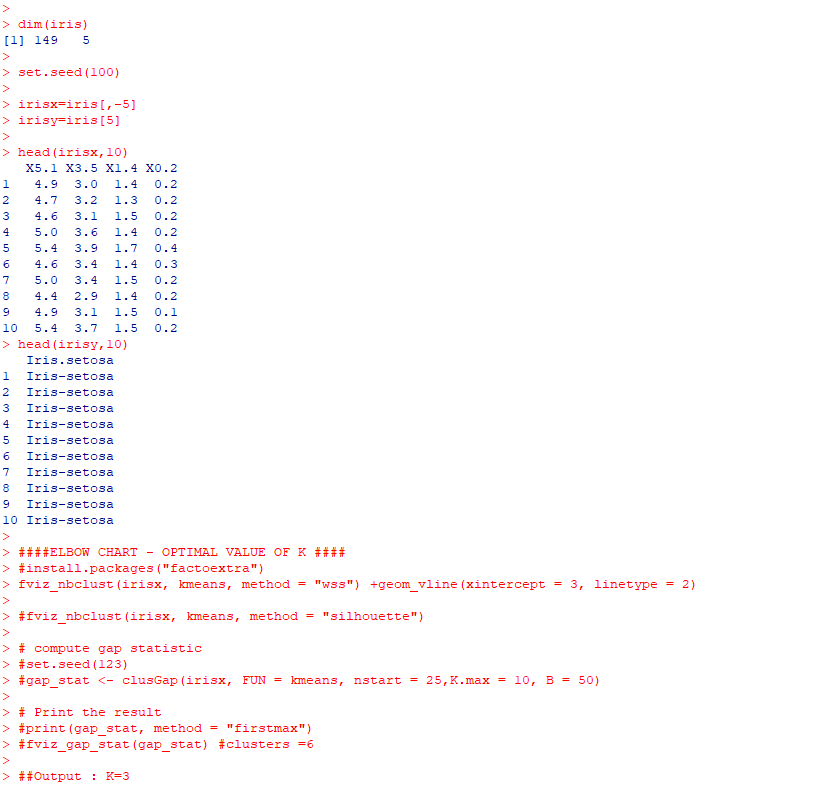
Use the R function kmeans to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.

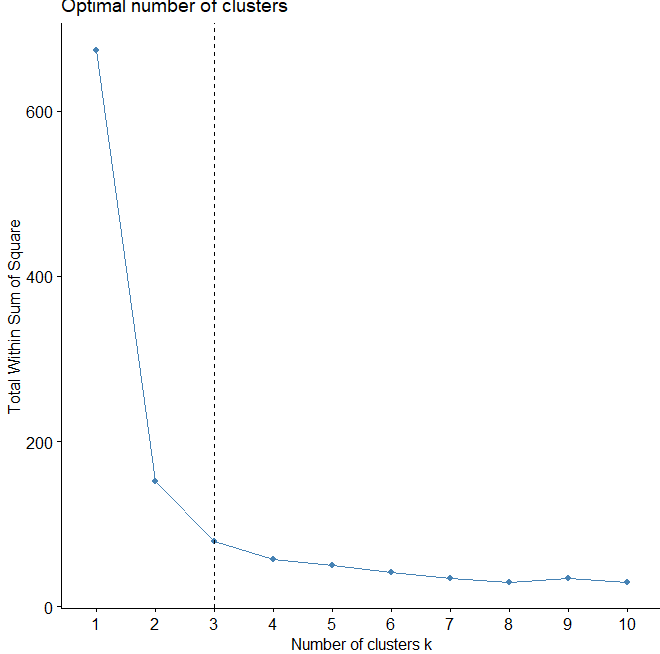
# Suggested K-value

Ingest iris data from url and import library



Use Elbow chart to predict best K value ; k=3 is the suggested output





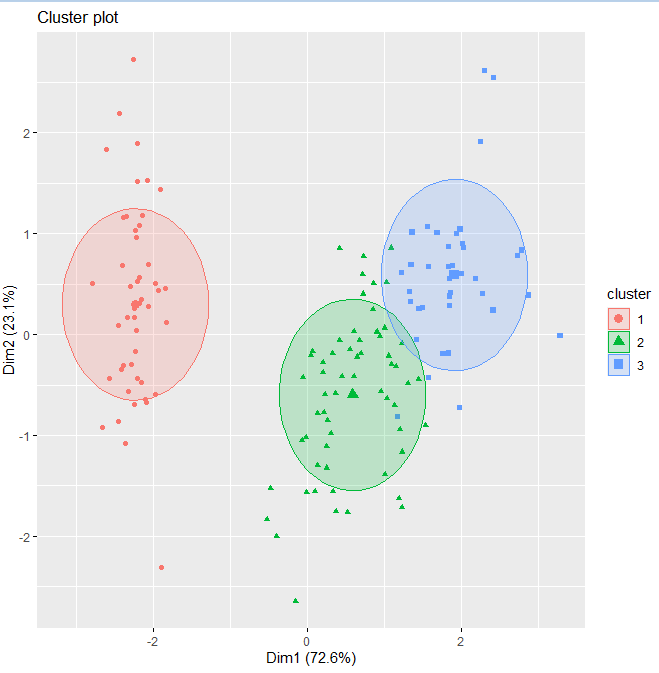
# Best combination of predictors

Train the model and understand the names of the k means model.

I was having issues with ggplot . but cluster came out fine.

Challenges with ggplots: The behavior was sporadic and hence I have not included the same in the results window

Created a table view and a dataframe on the kmeans model

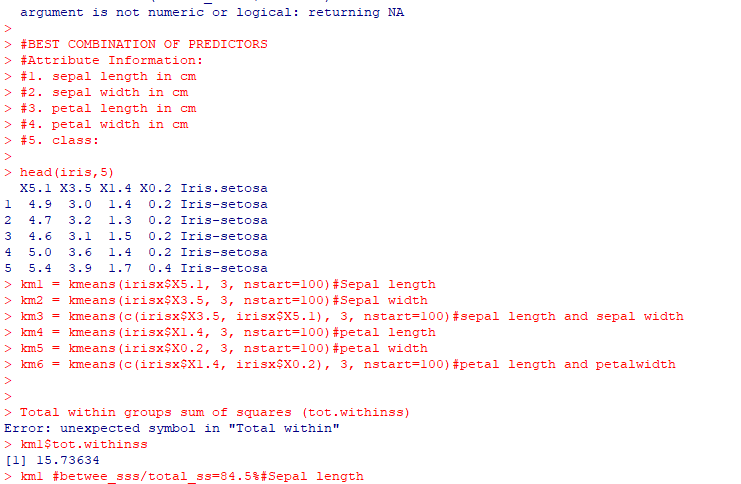




To identify best combination of predictors, I used the individual predictors and combination of petal and sepal length and width.

**I created multiple model for each of this combination and understood the accuracy of the model with the predictors**

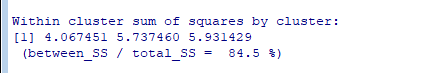
The



Below are the results of each of the predictors and combination.

Between\_ss/total\_ss provides accuracy of the model

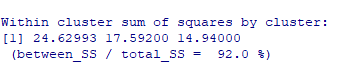
sepal length



Sepal width



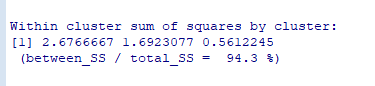
Sepal Width and length



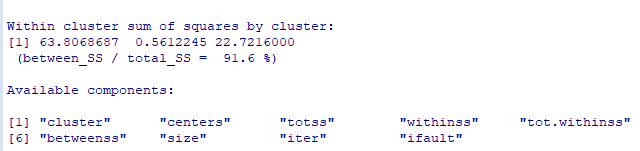
Petal length



**Petal width**



**Petal width and length**

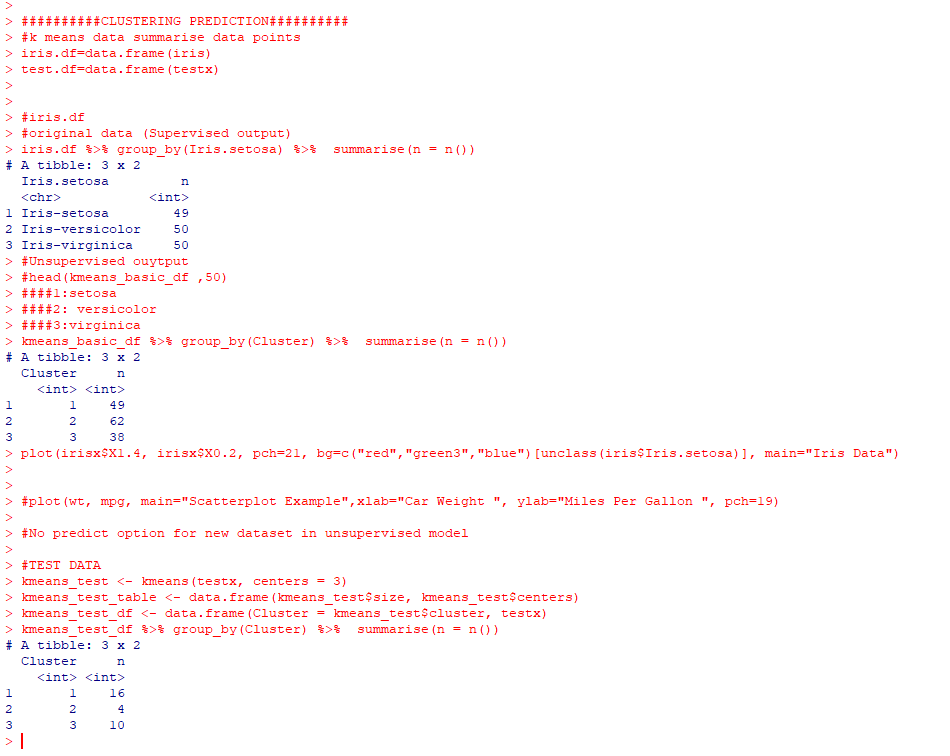


**##WINNER IS Petal Length with 94.7%**

**4.2 How well your best clustering predicts flower type.**

# I searched to obtain accuracy of the model. However unsupervised model donot have the accuracy predict component.

So, I followed an approach to use the input set which already holds the expected output and compare the cluster produced with k-means against the actual output itself.



# Summary of the comparison:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Actual | Unsupervised | variance | Variance% |
| Setosa | 49 | 49 | 0 | 0% |
| Versicolor | 50 | 62 | 12 | 24% |
| Virginica | 50 | 38 | -12 | -24% |