# K- Means Clustering Exercise

(MATH 3210 Data Mining Foundations- Report)
Professor: Dr. John Aleshunas

**Executive Summary**

In this report, the R k-means algorithm will be implemented to discover the natural clusters in the “Auto MPG dataset”. Once the number of clusters in the dataset is determined (if any), analytical techniques will be provided that support the findings. An experiment will be designed that will discover the number of possible clusters in the dataset. The experiment (using R studios) will conduct an analysis of the possible clusters to verify that they are an optimal set of clusters. The cluster count will be validated with appropriate analytical techniques. Due to the nature of this exercise, the code might be run multiple times until satisfactory results are achieved.

Due to the nature of this exercise, the definite “correct” answer is achievable, yet may be ambiguous and somewhat difficult to derive. That is, the data could likely be “fuzzy”. The aim will be to run the k-means code multiple times until optimal results are achieved. For this to work, visual tools will be used such as diagrams and graphs. The optimal number of natural clusters will evidently be concluded by via discretion.

According to the description of the dataset in the “UCI Machine Learning Repository”, the dataset has 398 instances and 9 attributes including the class attribute.
Here is the complete description of the dataset:

```
1. Title: Auto-Mpg Data

2. Sources:
   (a) Origin:  This dataset was taken from the StatLib library which is
                maintained at Carnegie Mellon University. The dataset was
                used in the 1983 American Statistical Association Exposition.
   (c) Date: July 7, 1993

3. Past Usage:
    -  See 2b (above)
    -  Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning.
       In Proceedings on the Tenth International Conference of Machine
       Learning, 236-243, University of Massachusetts, Amherst. Morgan
       Kaufmann.

4. Relevant Information:

   This dataset is a slightly modified version of the dataset provided in
   the StatLib library.  In line with the use by Ross Quinlan (1993) in
   predicting the attribute "mpg", 8 of the original instances were removed
   because they had unknown values for the "mpg" attribute.  The original
   dataset is available in the file "auto-mpg.data-original".

   "The data concerns city-cycle fuel consumption in miles per gallon,
    to be predicted in terms of 3 multivalued discrete and 5 continuous
    attributes." (Quinlan, 1993)

5. Number of Instances: 398

6. Number of Attributes: 9 including the class attribute
```

```
7. Attribute Information:

    1. mpg:           continuous
    2. cylinders:     multi-valued discrete
    3. displacement:  continuous
    4. horsepower:    continuous
    5. weight:        continuous
    6. acceleration:  continuous
    7. model year:    multi-valued discrete
    8. origin:        multi-valued discrete
    9. car name:      string (unique for each instance)

8. Missing Attribute Values:  horsepower has 6 missing values
```

Due to the issues with the "rgl" package and the 3d plotting functions, a two-dimensional approach may personally be favorable to those who are newer to using R studios. Still, one possibility could be erroneous errors in the code while navigating R studios.

The biggest assumption made in this exercise is the number of clusters based on the graphical plot of the data in the dataset. This number may easily be discerned differently to a different beholder, and is therefore completely intuitive. However, by using the k-means algorithm provided in R, one can differentiate the groups via color in the graphical analysis. This may boost the distinguishability of the clusters further.

It was concluded that there are no clearly discernable natural clusters in the dataset. This was decided upon observation of the data using the plot() function provided in R. There are too many variables that all have an effect on the fuel economy of a vehicle for this dataset.

**Problem Description**

This report uses the R k-means algorithm to discover the natural clusters in the "Auto MPG dataset". The aim is to come up with an optimal number of clusters in the dataset. The method being used, k-means clustering, is a method of 'vector quantization', originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition 'n' observations into 'k' clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells (MacQueen, 2016).

In mathematics, a Voronoi diagram is a partitioning of a plane into regions based on the distance between points in a specific subset of the plane. That set of points (called seeds, sites, or generators) should be specified beforehand, and for each seed there is a corresponding region consisting of all the points that are closer to that seed than to any other points. These regions are called Voronoi cells (Voronoi, 2016).

In regards to k-means, the most common algorithm uses an 'iterative refinement technique'. This basically means that it may be necessary to repeat the process until acceptable outcomes are derived. Due to its 'ubiquity' it is often called the k-means algorithm; it is also referred to as "Lloyd's algorithm", particularly in the computer science community. The algorithm proceeds by alternating between two steps:

Assignment step- this step assigns each observation to the cluster whose mean yields the least within-cluster sum of squares (WCSS). Basically, it assigns the observation to the cluster that it is nearest to.

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \forall j, 1 \leq j \leq k \right\},$$

Update step- this step calculates the new means to be the 'centroids' of the observations in the new clusters. Basically, this step creates the center point for a new cluster.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares (WCSS) objective (MacQueen, 2016).

The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a finite number of such partitions, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm (MacQueen, 2016). In applied mathematics and computer science, a local optimum of an optimization problem is a solution that is optimal (either maximal or minimal) within a neighboring set of candidate solutions. This is in contrast to a global optimum, which is the optimal solution among all possible solutions, not just those in a particular neighborhood of values (Stewart, 2016).

The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by “least sum of squares”, which is exactly equivalent to assigning by the smallest Euclidean distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging (MacQueen, 2016). In mathematics, the Euclidean distance or Euclidean metric is the straight-line distance between two points in “Euclidean space”. With this distance, Euclidean space becomes a metric space (Deza, 2016).

The Euclidean distance formula may look like the distance formula. However, it is given by the Pythagorean theorem.

The **Euclidean distance** between points **p** and **q** is the length of the line segment connecting them.

In Cartesian coordinates, if **p** = ($p_1$, $p_2$,..., $p_n$) and **q** = ($q_1$, $q_2$,..., $q_n$) are two points in Euclidean n-space, then the distance (d) from **p** to **q**, or from **q** to **p** is given by the Pythagorean formula:

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

(Deza, 2016).

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (or cluster) are more similar (in this case, data proximity) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning and computer graphics- just to name a few. There is no perfectly “correct” clustering algorithm per say. The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another. It should be noted that an algorithm that is designed for one kind of model would not fare well on a data set that contains a radically different kind of model. For example, k-means cannot find “non-convex clusters” (Tryon, 2016).

**Analysis Technique**

Due to the nature of this exercise, the definite “correct” answer is achievable, but may be difficult to derive. The data could likely be “fuzzy”. The aim will be to run the k-means code multiple times until

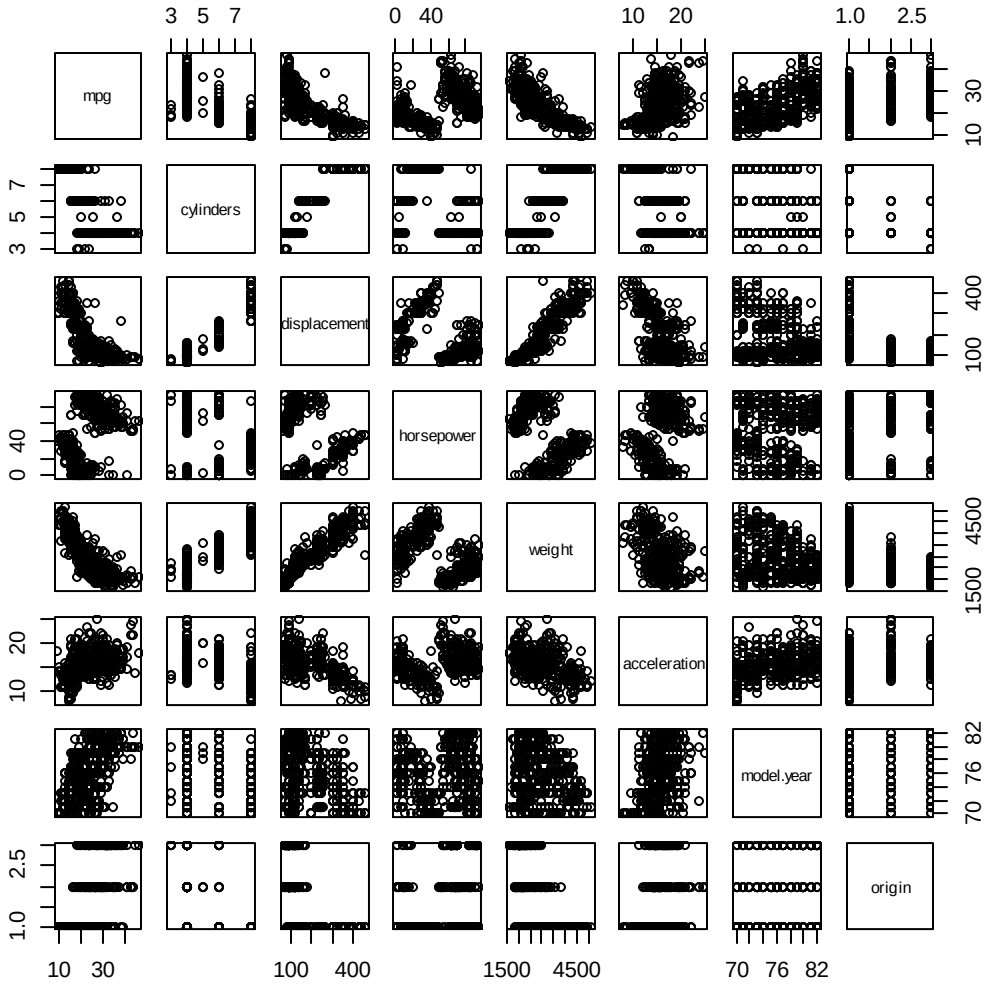
optimal results are achieved. For this to work, visual tools will be used such as diagrams and graphs. The optimal number of natural clusters will evidently be concluded by personal discretion.

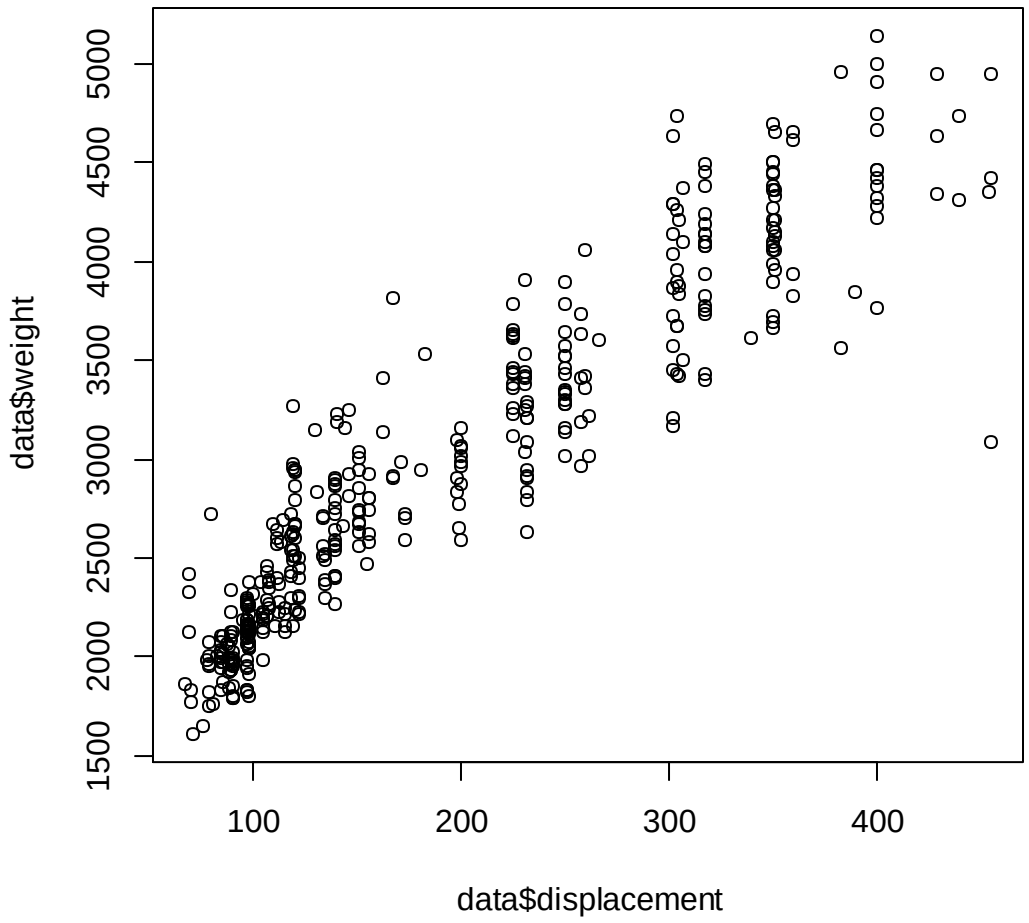
The first task completed was the instillation of the “rgl” package in R. This package may be necessary for the plotting and graphing in this exercise. Next the “Auto MPG dataset” will be loaded into R. For the sake of data clarity, the str(), summary(), and fix() functions will be utilized to get a look at the numbers. The View() function was also used after the car.name category was nullified. This was also done to get a clearer look at the dataset.

The entire data set was plotted, save the class category- which only contained the names of the vehicles. There was so much information on the graph- it was virtually impossible to sit there and derive an estimate number of natural clusters by reading the graph alone. To atone for this, the data was looked at in section categories with a x and y axis format.

The data was partitioned- using the displacement, which is the swept volume of all the pistons inside the cylinders of a reciprocating engine in a single movement from top dead centre (TDC) to bottom dead centre (BDC), and the weight of the vehicle (Chrysler, 2016). Even by partitioning the data using these two categories, no clear natural clusters were humanly derivable to the naked eye alone.

Here are both the plot of the dataset and the partitioned graph of the weight of the vehicle against the displacement of the engine:


mpg
cylinders
displacement
horsepower
weight
acceleration
model.year
origin
3 5 7
0 40
10 20
1.0 2.5
10 30
100 400
1500 4500
70 76 82




As can be seen here, there are two many variables that separately affect the fuel economy of a vehicle. Therefore, it may be concluded that there are no clear natural clusters in this particular dataset.

**Assumptions**

The biggest assumption made in this exercise is the number of clusters based on the graphical plot of the data in the dataset. This number may easily be something different to a different eye, and is therefore completely intuitive. However, by using the k-means algorithm provided in R, one can differentiate the groups via color. This can help distinguish the clusters further.

**Results**

According to the study of the data via the graphing of the separate categories and the plotting of the dataset, it may be concluded that there are no clearly derivable natural clusters in the “autoMPG” dataset.

**Issues**

Due to the nature of the dataset results may be slightly skewed. This is due to the fact that there are six missing values in the horse power category, and the kmeans clustering algorithm works best when there are no missing values in a dataset.

Due to the nature of this exercise, the definite “correct” answer is achievable, yet may be somewhat ambiguous. The data could likely be “fuzzy”.

Initially, there were some issues with the “rgl” package. The package would seem to load properly but failed to be accessed using the require() and library() functions. Additionally, when the plot functions were used, an error message would be displayed stating that there is an invalid x.

Due to the issues with the “rgl” package and the 3d plotting functions, a two-dimensional approach may personally be favorable to those who are newer to using R studios. Still, one possibility could be erroneous errors in the code while navigating R studios.

It may have been more beneficial to have all the clusters separated by color on one diagram instead of the separate diagrams showing the separate clusters. However, this could not be achieved due to a lack of knowledge of R coding. It is recommended that those attempting analytical experimentation using R have a firm grasp of at least the fundamentals of R code. One helpful resource is the “R Cookbook”. Other helpful tutorials may also be found online.

**Appendices**

Additional evidence to the claim that there are no clearly derivable natural clusters in this dataset may be the fact that the kmeans algorithm failed in this experiment. This may be because the algorithm could not find groupings of similar points to build clusters with.

**References**


Deza, E. (2016, October 20). *Euclidean distance.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Euclidean_distance

MacQueen, J. B. (2016, October 20). *k-means clustering.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/K-means_clustering

Stewart, J. (2016, October 20). *Local optimum.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Local_optimum

Tryon, R. C. (2016, October 20). *Cluster analysis.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Cluster_analysis

Voronoi, G. (2016, October 20). *Voronoi diagram.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Voronoi_diagram

Chrysler. (2016, December 7). *Engine Displacement.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Engine_displacement