Mining Dynamic Traffic States Over Large Road Network from Big Data

Anju, Jose (n10434411)
IFN703/4 Assessment 3, due 11:59 pm Sunday 22 March 2020

Executive Summary

Traffic congestion is a major problem faced by the highly populated cities all over the world, where thousands of cars are passing through the streets every single day. The problem, traffic congestion will occur when the transport demand exceeds the infrastructure supply capacity. The main causes of traffic congestions are the increase in the number of cars on the road when people use their vehicles to get around, inadequate green time, obstacles in the road like road works, accidents, pets, speed bumps, and more. By widening roads, building a tunnel can reduce traffic congestion to a point. Because of the high cost and limited space for an extension makes these solutions impractical, also widening the whole road is practically impossible. So, by identifying the areas where there is very high traffic and then develop strategies to overcome the traffic congestion on the identified locations. Various studies were held to understand the dynamic traffic behavior based on historical data to develop strategies for Intelligent Transportation System (ITS) which provide services that are related to the different modes of transport and traffic management and help the end-users to get informed about the of the road network on a real-time basis. This study proposes a data mining technique to identify the traffic pattern over various locations in Brisbane based on historical data. The study uses the average travel time data captured by Bluetooth sensors from July 2019 to December 2019 for 96 intervals on every single day and then cluster the data based on days which shows some similarities, results in the assessment of travel time profile at a particular time interval based on the sensors, analysis of travel pattern and in the identification of the sensor which has taken a larger travel time. The sensors that show a larger travel time states that there is higher traffic congestion. Thus, the identification of travel time profile at various time interval across the entire road network helps the traffic authorities to analyze and optimize the configurations of the road network in reducing traffic congestions, so city development decisions can be made in a better way which ultimately helps the road users to plan their trip.

Introduction

As cities continue to grow, traffic congestion becomes an inevitable problem. Some causes of this are the increase in the number of privately-owned vehicles, weather, obstacles in the road, etc. There are numerous ways to reduce these causes like the use of public transport, widening roads, alternate travel paths, and more. Analysis of traffic data from past observations helps to identify the areas which are having heavy traffic and based on these analysis result authorities can bring up some solutions to deuce the traffic in the determined areas. Predicting the traffic model/pattern from big traffic data helps in the development of ITS, which aims to alleviate traffic congestion and enables road users to plan their trip depending on the information about the road network conditions [1]. Data mining is a tool that uses different techniques and algorithms [2-4] to devise the pattern or extract the

information from the dataset and to predict the future. In Traffic data analysis, data mining is used to identify the traffic pattern, helps in determining the locations with heavy traffic.

The study proposes a data mining technique called clustering to identify the traffic congestion areas in the Brisbane region. The study uses Bluetooth sensor data. Where the Bluetooth captures the average travel time in seconds between a pair of sensors for a time period in the form of time-series data. Group or cluster the data based on the days which are similar and thus it determines how travel time varies at different time intervals also, determine the locations that have heavy traffic. The main objective of the study is to analyze the locations which have heavy traffic and with the use of this information develop strategies to reduce traffic congestion in that locations also these traffic flow predictions help the traffic managers to analyze and optimize the road network in reducing the traffic congestion which ultimately helps the road users to plan their trip in advance.

The data used in this study is measured using Bluetooth sensors. Where every Bluetooth scanner has a communication range and this scanner will capture the unique Media Access Control address (MAC-ID) [5, 6] of the discoverable transiting Bluetooth device vehicle which passes through that zone. This identification helps in determining the average time taken to travel by considering the time difference between the two detections if the same MAC-ID is detected at two different locations and then it will compute the speed. Thus, the result obtained from the multiple Bluetooth devices is used to calculate the average travel time between the sensors. In our study, we use the average travel time calculated from the Bluetooth sensors is used to cluster the data for the analysis.

Data clustering is a method which group of objects which are like each other and group the different object, and each group of objects are called clusters. There are different clustering techniques [7-9] such as hierarchical, portioning, and density-based. In the proposed study, a partitioning method called K-means is used to group similar object belonging to one cluster and each of the clusters represented by a centroid, which is the mean of the data points within that cluster. Where K-means [10-12] uses a pre-defined number of clusters K to set the centroids or the initial cluster centers by identifying the distance between the data points and the centroids. For each iteration of K-means, the distance between the points and new centroids is measured and assigns the data points to the closest centroids and this repeats until the values become unchanged. That is in each iteration, the value of centroids gets updated to get closer to the mean of each cluster. Where the methods to calculate the distance between the data points [13] can be measured using Euclidean distance, Manhattan distance, Jaccard Index, and a few more. The distance measure will determine the similarity between the data points [14], which influence the shape of the clusters.

To perform this cluster analysis, we need to set the optimal number of clusters K. For a small dataset, it is easy to define a K value. Since our data is large and there is no clue to set the K value, different methods like Elbow method, Silhouette method, Gap Statistic method is used to determine the optimal number of clusters. Based on the result from these methods we choose the Elbow method to pre-define the value of k. Once the value of K is determined, K means algorithm is performed, which results in the identification of the clustered points, and an exploratory data analysis [15] based on the clustered data shows the locations which have very high traffic at various time intervals.

This study analyses the traffic flow based on the average travel time between two sensors to identify the travel path of vehicles or the path which takes more time to travel for traffic optimization. The result of the study allows the traffic department authorities to understand how the travel time varies

from one location to another at a given time, which finally helps them to expand or restructure the road network for traffic optimization.

Literature Review

Data mining technology can be used to predict the traffic states of various locations by using historical data. Loop detectors and Bluetooth sensors are used to collect information about traffic data, to predict the patterns and the behavior of the traffic data. A traffic state includes an array of time-dependent characteristics that provide information about system information, which enables us to discover patterns and gain an understanding of traffic dynamics [16]. In data mining, machine learning techniques are used to observe a pattern, make intelligent decisions based on data, predict the future, and helps in the improvement of various traffic management and control strategies.

To perform a clustering technique, first, we need to set a pre-defined number of clusters. There are different methods to determine the value of K. The most popular and widely used one is the Elbow method [17]. This is a visual method, where a plot is drawn between the within-cluster sum of square (WSS) and the K values. When the K value becomes larger, the cluster will become smaller in size and thus it will reduce the intra-cluster distances. The value of K is obtained, when the WSS decreases abruptly and this produces an elbow effect i.e. the point where there is a bend (Figure 2.) in the plot is considered as the number of clusters. Another method to determine K is the Silhouette method [17], where it compares the within-cluster distances between cluster distances. The greater distance gives the best K value. i.e., the Silhouette score measures how similar an object to its cluster compare to the other cluster. The value ranges from -1 to 1, where the low value says there are too many or a smaller number of clusters and a high value indicates that the data points are matched to its cluster and poorly matched to the other clusters. A study [18] was proposed to determine the optimal number of clusters using different techniques like determining K using neighborhood measure, through visualization, by statistical measures, etc for the K-means clustering algorithm. The result shows that the proposed method suggests multiple values for K where different clustering results could be obtained with various levels of data. However, the methods are computationally expensive for large data sets.

Several kinds of research have been conducted to analyze the traffic pattern and behavior using different clustering techniques like K-means clustering, hierarchal clustering, probability-based clustering, and so on. [19] suggests a method using hierarchical data for univariate data with fixed dimensionality, uses probabilistic clustering of heterogeneous data types. The use of large data sets in the analysis of this technique makes the model complex. Another study [20] propose proposed an algorithm named, Grey Relational Membership Degree Rank Clustering (GMRC) to segregate the cluster ranking and to analyze the traffic condition from the traffic flow characteristics, velocity, density, and volume. The result of this study shows that the GMRC algorithm is much better than the k-mean algorithm.

A study [11] was conducted to estimate the traffic density with the help of location-based sensors, which detect the volume and speed of the vehicle passing through the sensors. The machine learning techniques such as k-Nearest Neighbour (k-NN), and Artificial Neural Network (ANN) are used for the estimation and prediction based on acceptable performance on the dataset and the result says that the use of ANN and k-NN techniques along with automated sensor data are useful for traffic state

estimation problems. However, combining these two techniques is not recommended because it did not show any improvement in the performance due to a reduction in the training data set.

[12] proposes an enhanced algorithm to make clustering more efficient & effective. A limitation of this algorithm is the number of clusters should be pre-defined, which is not so easy for large data sets. A study by [3] proposes an Improved k-mean clustering algorithm for prediction analysis to reduce the clustering time and to increase the efficiency, which can define automatically define the number of clusters, also assign required clusters to un-clustered data points. The major drawback of this study is that this proposed method can be only applied to smaller data sets, which gives better accuracy and consume calculation time in clustering, but this is not applicable for large data sets. A study based on Understanding the daily mobility pattern using traffic flow analytics [1], examines the characterize of traffic flow in urban road scenarios with an emphasis on the long term and the result says that by clustering the traffic flow in different points of a road network can be characterized and this characterization allows performing long-term predictions which is not as accurate for short-term forecasts. Even though extensive studies are going on to develop or replace K-mean clustering, still the selection of an optimum number of k clusters remains a significant issue limiting their analysis.

Numerous studies and researches are undergone and still undergoing to reduce the limitation of clustering techniques. The major challenge faced by many researchers is in identifying the exact number of clusters. For a small data set, we can assume, but for a larger data set with multivariate data, it is not possible. Secondly, the time is taken to cluster the data and the accuracy level in clustering is difficult to identify for large data sets.

This paper will propose a k-mean clustering algorithm for a large data set. How does the K-means algorithm work?

K-mean clustering is a popular data-clustering algorithm. The following is the K-means algorithm:

Step 1 − First, we need to divide the dataset and specify the number of clusters, K, need to be generated by this algorithm.

Step 2 − Next, Selecting K data points and assign each data point to the nearest cluster.

Step 3 − Now compute the cluster centroids by dividing the total by the number of members of the cluster.

Step 4 − Keep iterating the following until we find an optimal centroid that is not changing anymore

> **4.1** − First, the sum of squared distance between data points (Euclidean distance measure) and centroids would be computed.
>
> **4.2** − Now, assign each data point to the cluster that is closer than other clusters (centroid).
>
> **4.3** − Compute the centroids for the clusters by taking the average of all data points of that cluster.

The proposed method will first identify the number of clusters by using Elbow method and then perform a K-means clustering algorithm on the data given based on days and predict the traffic states at different Brisbane location over a time frame, which will help to manage or reduce the traffic congestions also help the road users to plan their trip in advance.

K-means Clustering Technique

Clustering is an unsupervised learning technique, which group objects that show similarities. I.e., by clustering we will minimize the intra-cluster distance while maximizing the inter-cluster distances. In the study, the K-means clustering technique is used to group the days which show similarities in the average travel time.

- **The Data**

  Queensland Government open data portal provided the traffic datasets, gives information about the Average Travel Times for key priority routes on the state-controlled road network (Brisbane data). The main characteristics of this traffic dataset are:
  - The datasets contain information collected by various Bluetooth sensors from July 2016 through December 2020 for every 15 minutes intervals.
  - The dataset contains 219 pairs of sensors situated in various locations of Brisbane.
  - The dataset contains the average travel time between the 219 sensors captured by various Bluetooth sensors
- **Approach**
  - Data pre-processing

    Data Filtering: The datasets contain information collected by various Bluetooth sensors during the period July 2016 — January 2020 which are broken down in 15 minutes regular intervals, with 219 pairs of sensors. Which is a very large data set? As the size of the dataset is very large, we are only considering Morning 7.00 AM Pacific Motorway data from July 2019- December 2019. The Weekdays and Weekend traffic pattern is different. In this study, we are only considering the weekdays. So, we filtered the data again by excluding Saturdays and Sunday's data.

    Data Cleaning: Removing the unnecessary quotes and symbols from the data for the ease of further analysis.

    Dealing with missing values: The Bluetooth sensor data contains missing values in the dataset represented as Zero. The reason for the missing values can be different like the Bluetooth won't detect the car passes through the sensors doesn't have any Bluetooth, If the sensor is not working, the Bluetooth device is turned off in the car all these will show missing values. As the dataset contains time-series data, we are not dropping the missing values. To deal with the missing value, first, we replace the zeros with NaN values and then apply Kalman filter to impute values to the missing data points. Where Kalman filter uses the estimate of previous state to determine the next state (missing point).
  - Exploratory Data Analysis

    An exploratory data analysis is performed to visualize the data points and to analyze how the data are distributed. Based on the data travel time Vs INTERVAL_END is plotted for morning 7.00 AM data.

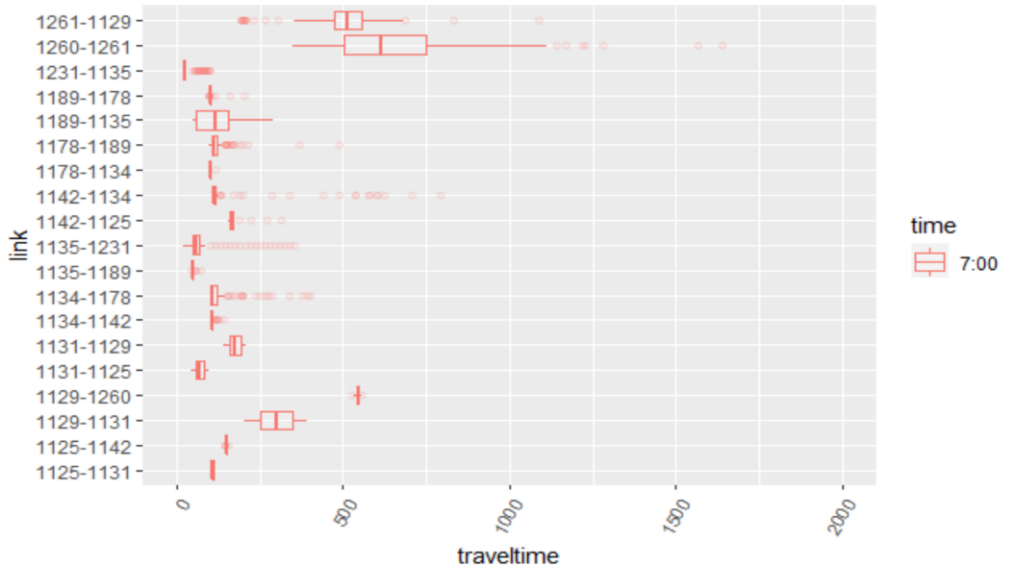


Figure 1. Travel Time Vs Time Interval

This plot shows the travel between the pair of links. The link 1260-1261 has very high traffic followed by the link 1261-1129 also the plot shows the link with lesser travel time.

- Data Normalization

  Standardization (Normalization) refers to the process of rescaling the values of the attributes in the dataset to develop a common scale where the values of the variables are different from one another. Clustering is a technique that is very sensitive to inputs on a different scale. Here we are using Euclidian distance to calculate the distance between the data points to find the cluster centers. With inputs on a different scale, Euclidian distance favors features on a larger scale. So, data normalization is performed.

- Determining the optimal number of clusters, K

  To perform clustering, we need to find the optimal number of clusters, K. Once the value of K is defined, group the data. There are different approaches to find the value of K. In this study, three different techniques were tried to find the optimal value.

  Elbow Method: This is the widely used method to set the K value. A plot is drawn between the K values and the WSS. When the K value becomes larger, the cluster will become smaller in size and thus it will reduce the intra-cluster distances. The value of K is obtained, when the WSS decreases abruptly, Where the value of the WSS shows the clustering error. Figure 1 shows the plot obtained by using the Elbow method, to determine the optimal number of clusters.

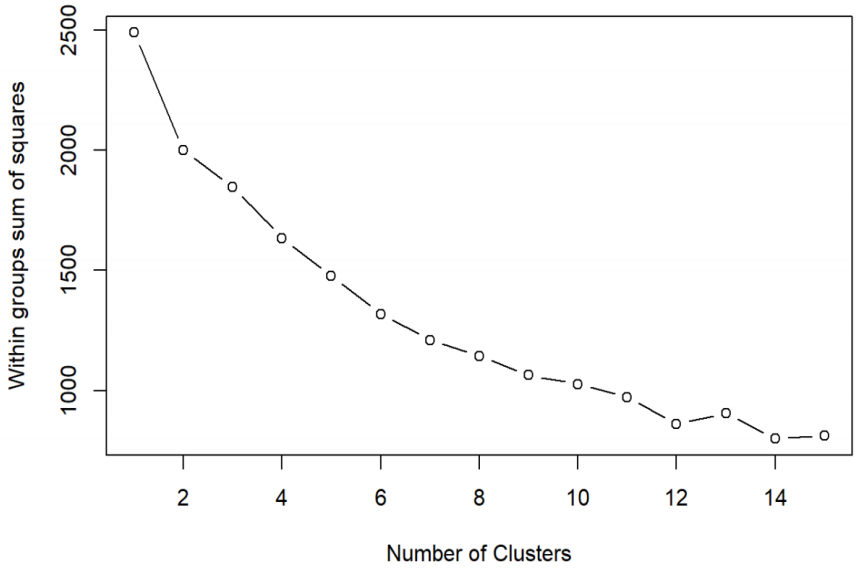


Figure 2. Elbow Method

Here, the elbow is somewhere between points 4 and 6. Either of the value can be selected as the optimal K. However, the elbow method is always not accurate. In some cases, it may plot obtained will be very smooth and show no recognizable values for k. So, an alternative method called Silhouette method is used

Silhouette Method: This method measures the similarities between the data points of its cluster and compared it to the other cluster. The value of the Silhouette method ranges from -1 to 1. Where a high-value show, the objects are matched with its cluster, and if the value is low or negative, then it says that the method has too many or very few clusters. Based on our dataset, the analysis shows that the number of clusters estimated by the average silhouette width is 10. Since the value obtained for K is different from the Elbow method, one more method is performed.

Gap Statistic: Gap statistic is the goodness of clustering measure, in which the range of clusters k, it compares two functions: log of the within-cluster sum of squares (WSS) with its expectation and the null reference distribution of the information. Based on our dataset, the analysis shows that the number of clusters estimated by the Gap statistic method is 8.

From the three approaches, based on the elbow method the optimum number of cluster is chosen 6,

- Clustering Data

Performing the clustering technique, once the value of the optimum number of clusters is obtained. Based on the three approaches above, six is the optimum number of clusters chosen for the traffic data analysis. Now, building the clustering model with K value as 6. In this study, we are grouping the days which show similarities in their average travel time. Here all the data points are assigned to the closest centroid and the centroid value gets updated until an optimal centroid value is obtained.

The clustering results in 6 clusters with size 46, 14, 1, 39, 18, 14. It is difficult to plot the result because of the high dimensional nature of the data. So, Principle Component Analysis (PCA) is performed to plot the K means result in two-dimensional data. Where the first dimension shows the variables with a higher variance and the second dimension shows the variables with less variance in their values. Below is the plot of the K-means result.

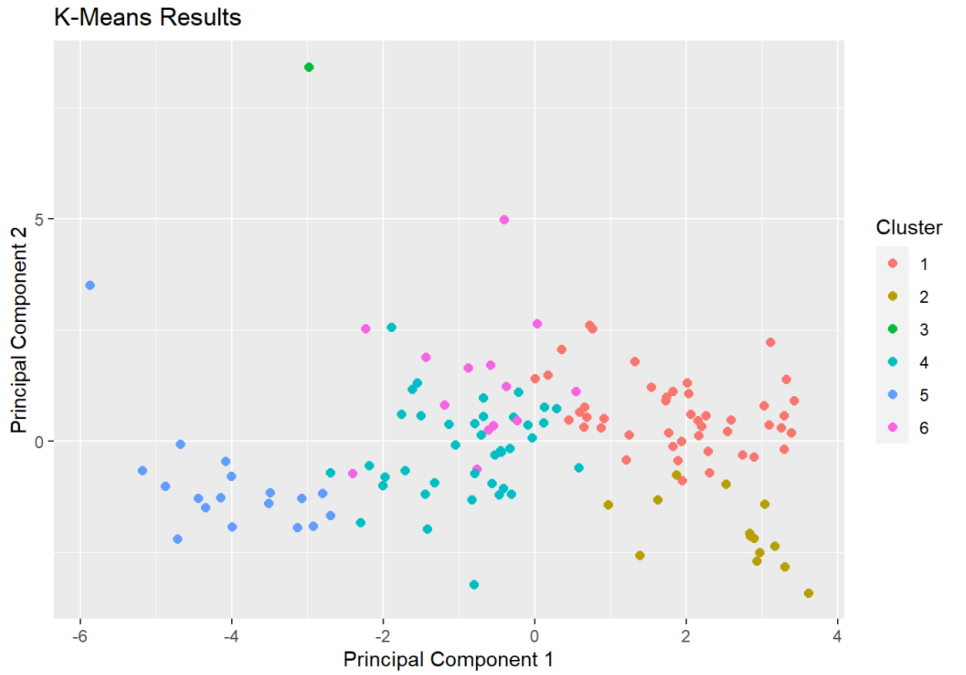


Figure 3. K-means analysis plot

In this study, the days which perform similarly are get clustered based on the average travel time taken in a link. The plot below shows, the days belong to which cluster.

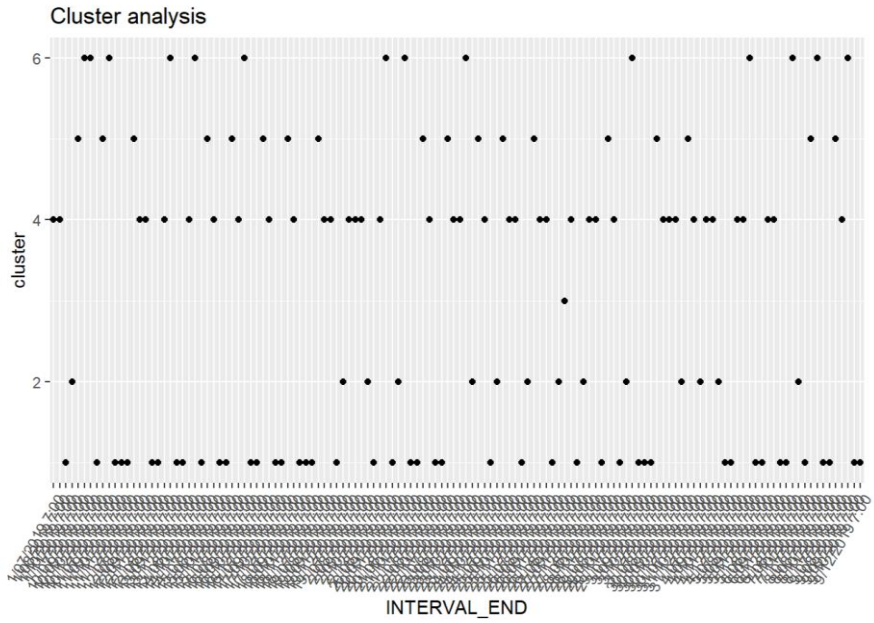


Figure 4. Cluster Analysis

Here it's not easy to identify which days belong to which cluster. So, the below plot will show the distribution of each cluster.

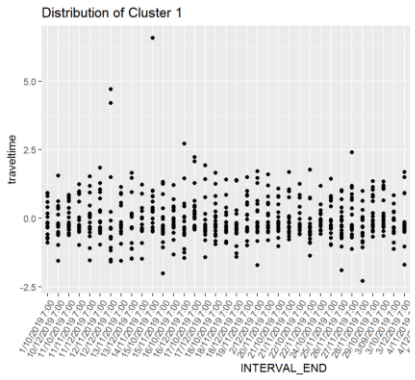


Figure 5. Distribution of Cluster 1

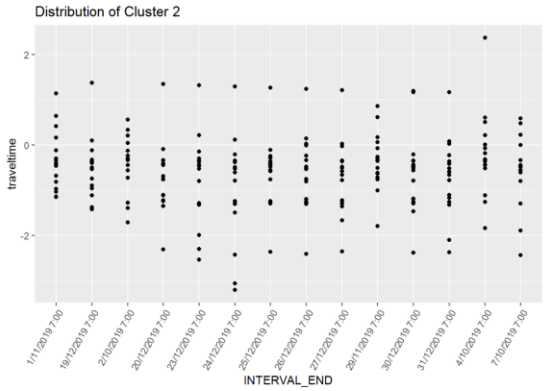


Figure 6.Distribution of Cluster 2

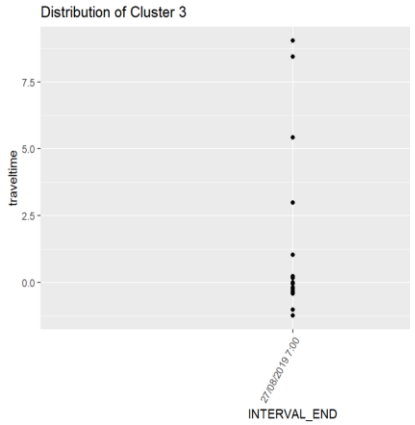


Figure 7. Distribution of Cluster 3

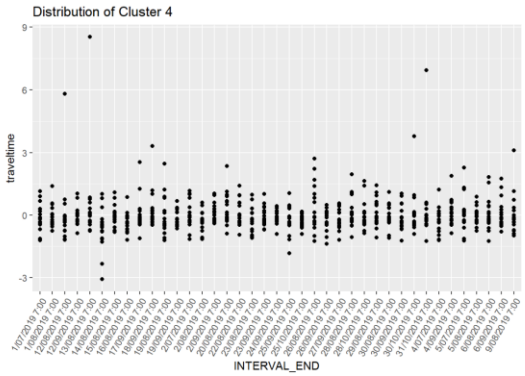


Figure 8. Distribution of Cluster 4

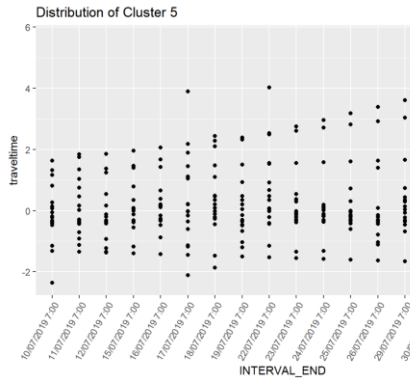


Figure 9.Distribution of Cluster 5

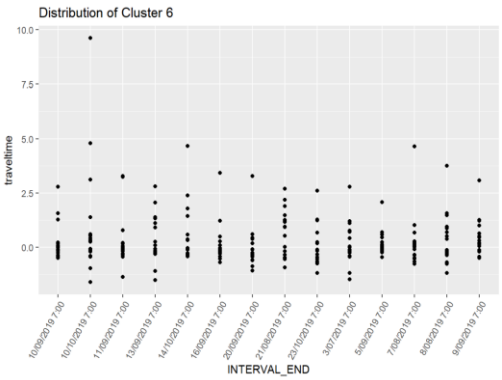


Figure 10. Distribution of Cluster 6

In this study, we analyze the average travel time through various links, to identify the link which has higher traffic by performing cluster analysis. The result of this analysis will show, how the travel time profile varies at a different time interval with respect to each cluster at 7.00 AM. This helps in the identification of the sensor which has very high traffic. The below-given plot Figure 11. will show this. To compare the travel time profile for the various link at 7.00 AM, the travel time profile at 8.00 AM is also plotted from the same dataset.

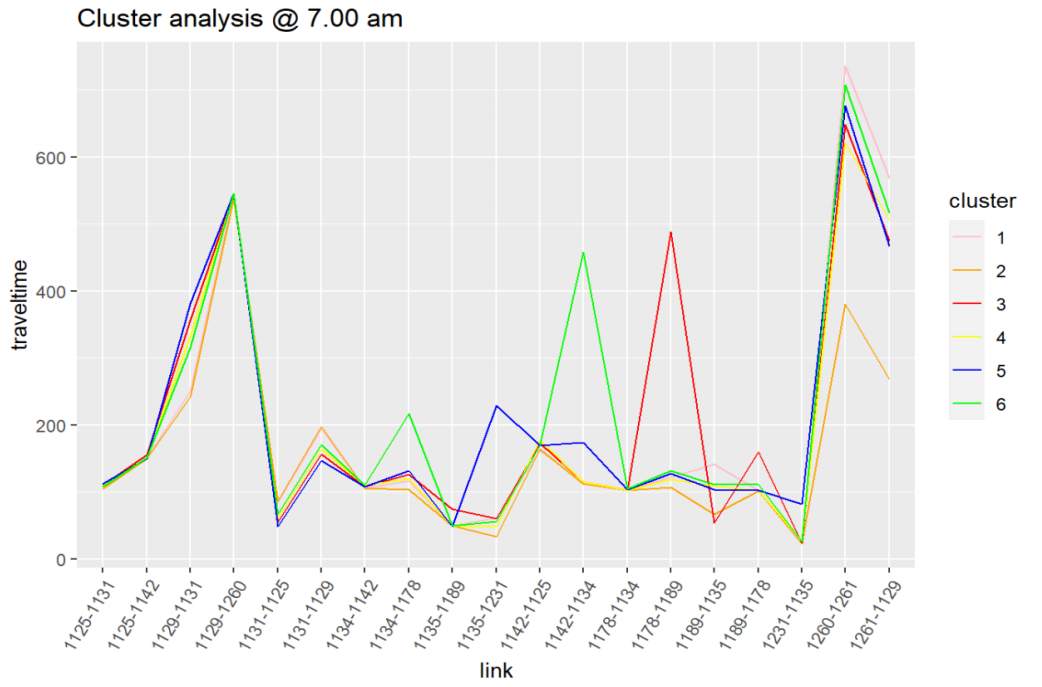


Figure 11. Cluster Analysis at Morning 7.00 AM

The result shows that the travel time profile at 7.00 AM for every link in each cluster. The clusters with a high traffic time show the link which has heavy traffic during the morning at 7.00 AM. This says that the link 1129-1131, 1129-1260, 1260-1261, 1261-1129 has a very high travel time at each cluster. i.e., on every day these links have heavy traffic congestion. This means that these links have heavy traffic on an everyday morning at 7.00 AM and on some days the days in cluster 3 have heavy traffic, maybe because of some unexpected events or other reasons cause this. Where 1129-1131 is Pacific Motorway South of Underwood Road Overpass (MET148), 1129-1260 is Pacific Motorway South of Underwood Road Overpass (MET148), 1260-1261 is Pacific Motorway NB south of Logan River (M5702). Theses are locations which have very high traffic during morning seven 7.00 AM in the pacific motorway.

For a comparison of morning 7.00 AM data with 7.15 AM and 8.00 AM, the plot is shown below, which is obtained through the same clustering process

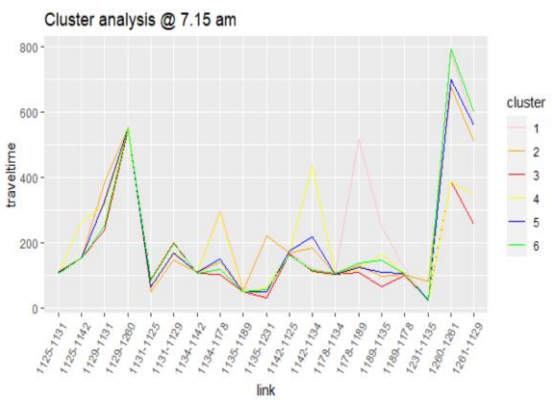


Figure 12. Cluster analysis at morning 7.15 AM

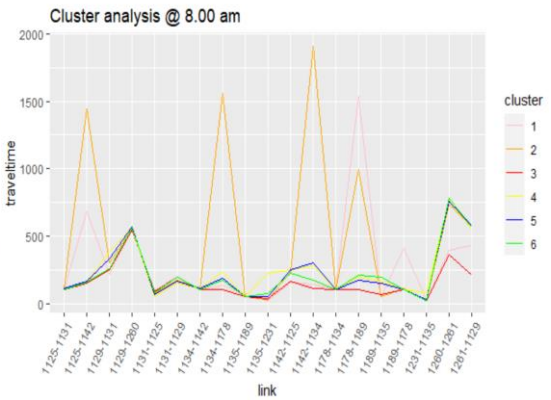


Figure 13. Cluster analysis at 8.00AM

The result shows that the travel time profile at 7.15 AM and 8.00 AM for comparison with the travel time profile at 7.00 AM data. The comparison result shows that morning 7.15 AM has the same traffic level of 7.00 AM data. Morning 8.00 AM also has similar traffic, but in the days of cluster 2 shows very high traffic.

- **Findings**

  The result of a proposed study of the Brisbane pacific motorway data study shows that the link 1129-1131, 1129-1260, 1260-1261, 1261-1129 (1129-1131 is Pacific Motorway South of Underwood Road Overpass (MET148), 1129-1260 is Pacific Motorway South of Underwood Road Overpass (MET148), 1260-1261 is Pacific Motorway NB south of Logan River (M5702)) has high traffic during morning time on every single day. The analysis result will help the authorities to take measures to optimize the traffic congestion in these locations.

- **Reflection**

  The result of the study helps in the identification of the travel time profile through various links of pacific motorway data in the morning at 7.00 AM. A k-means algorithm was used to cluster the data for the analysis. Which helps to identify the location which has very high traffic. Similarly, this can be done for every 15min time intervals also, to find the traffic state at time intervals. This analysis helps the traffic department authorities to build and implement new strategies to reduce road traffic congestion and to help the road users to plan their trip in advance.

  In this study, we perform a K-means clustering algorithm to identify the traffic of pacific motorway data during morning7.00AM. A further study can be performed by adding more sensors for different time intervals. This can be done by either K-means or by any other clustering techniques that help in the identification of the dynamic traffic states.

References

[1] I. Lana, J. Del Ser, and I. I. Olabarrieta, "Understanding daily mobility patterns in urban road networks using traffic flow analytics," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, 2016: IEEE, pp. 1157-1162.

[2] F. Gorunescu, *Data Mining: Concepts, models and techniques*. Springer Science & Business Media, 2011.

[3] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[4] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

[5] A. Bhaskar and E. Chung, "Fundamental understanding on the use of Bluetooth scanner as a complementary transport data," *Transportation Research Part C: Emerging Technologies,* vol. 37, pp. 42-72, 2013.

[6] S. M. Remias, A. M. Hainen, J. K. Mathew, L. Vanajakshi, A. Sharma, and D. M. Bullock, "Travel Time Observations Using Bluetooth MAC Address Matching: A Case Study on the Rajiv Gandhi Roadway: Chennai, India," 2017.

[7] P. Michaud, "Clustering techniques," *Future Generation Computer Systems,* vol. 13, no. 2-3, pp. 135-147, 1997.

[8] I. R. Rao, "Data mining and clustering techniques," in *DRTC Workshop on Semantic Web*, 2003, vol. 8.

[9] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*: Springer, 2006, pp. 25-71.

[10] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition,* vol. 36, no. 2, pp. 451-461, 2003.

[11] K. Alsabti, S. Ranka, and V. Singh, "An efficient k-means clustering algorithm," 1997.

[12] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," in *ICML*, 1998, vol. 98: Citeseer, pp. 91-99.

[13] S. Pandit and S. Gupta, "A comparative study on distance measuring approaches for clustering," *International Journal of Research in Computer Science,* vol. 2, no. 1, pp. 29-31, 2011.

[14] A. Vimal, S. R. Valluri, and K. Karlapalem, "An Experiment with Distance Measures for Clustering," in *COMAD*, 2008, pp. 241-244.

[15] J. W. Tukey, *Exploratory data analysis*. Reading, Mass., 1977.

[16] A. Paz, C. Gaviria, C. Arteaga, and J. Torres-Jimenez, "Mining Dynamic Network-Wide Traffic States," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018: IEEE, pp. 999-1004.

[17] T. M. Kodinariya and P. R. Makwana, "Review on determining number of Cluster in K-Means Clustering," *International Journal,* vol. 1, no. 6, pp. 90-95, 2013.

[18] D. T. Pham, S. S. Dimov, and C. D. Nguyen, "Selection of K in K-means clustering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* vol. 219, no. 1, pp. 103-119, 2005.

[19] I. V. Cadez and P. Smyth, *Probabilistic Clustering Using Hierarchial Models*. Information and Computer Science, University of California, Irvine, 1999.

[20] Y. Zhang, N. Ye, R. Wang, and R. Malekian, "A method for traffic congestion clustering judgment based on grey relational analysis," *ISPRS International Journal of Geo-Information,* vol. 5, no. 5, p. 71, 2016.

Appendix 1

IFN703-Advanced Project-n10434411

Anju Jose

Reading data

From the dataset downloaded from the Queensland open data portal, we consider the data from July 2019 - December 2019 for the analysis

```
library(tidyverse)
library(chron)

sensor_data_dec <- read_csv("Priority-Route-Bluetooth-Travel-Times-Dec-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   INTERVAL_END = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
sensor_data_nov <- read_csv("Priority-Route-Bluetooth-Travel-Times-Nov-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   INTERVAL_END = col_character()
## )
## See spec(...) for full column specifications.
```

```
sensor_data_oct <- read_csv("Priority-Route-Bluetooth-Travel-Times-Oct-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   INTERVAL_END = col_character()
## )
## See spec(...) for full column specifications.
```

```
sensor_data_sep <- read_csv("Priority-Route-Bluetooth-Travel-Times-Sep-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   INTERVAL_END = col_character()
## )
## See spec(...) for full column specifications.
```

```
sensor_data_aug <- read_csv("Priority-Route-Bluetooth-Travel-Times-Aug-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   INTERVAL_END = col_character()
## )
## See spec(...) for full column specifications.
```

```
sensor_data_jul <- read_csv("Priority-Route-Bluetooth-Travel-Times-July-2019.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   INTERVAL_END = col_character()
## )
## See spec(...) for full column specifications.
```

```
df <- read_csv("Priority-Route-Link-Details-2020.csv")
```

```
## Parsed with column specification:
## cols(
##   LINK_DETAILS = col_character(),
##   ORIGIN_DESC = col_character(),
##   ORIGIN_LONGITUDE = col_double(),
##   ORIGIN_LATITUDE = col_double(),
##   DEST_DESC = col_character(),
##   DEST_LONGITUDE = col_double(),
##   DEST_LATITUDE = col_double()
## )
```

Data Pre-processing

Creating a single data frame

```
sensor_data <- rbind(sensor_data_dec,sensor_data_nov,sensor_data_oct,sensor_data_sep,sensor_data_aug,sensor_
data_jul)
```

get data from monday = 1 to Friday = 5 (5 days of the week)

Traffic flow is different for weekdays and weekends. Here we are only considering the data from Monday to friday for the traffic data analysis

```
sensor_data <- sensor_data[!chron::is.weekend(as.Date(sensor_data$INTERVAL_END, "%d/%m/%Y")), ]
```

Cleaning the data

Cleaning the daat by removing the unnecessary symbols, for the ease of further analysis and coding

```
# Converting data from wide to long format
library(tidyr)
sensor_data_long <- sensor_data %>% gather(link, traveltime, -c(INTERVAL_END))

#Cleaning the data
sensor_data_long$link <- gsub("'","",sensor_data_long$link)
sensor_data_long$link <- gsub(">","",sensor_data_long$link)
```

Dealing with missing values

The data contain missing values, represented as Zero. As we considering a time-series data for the analysis, we are not dropping the observations with missing values. Because dropping the missing values in the time series data may cause variation in th efinal result. So we replace Zero with NaN and then uses Kalman filter to impute the missing values.

```
library(dplyr)
sensor_data_long<-na_if(sensor_data_long, 0)
#is.na(sensor_data_long$traveltime)
sum(is.na(sensor_data_long$traveltime))/length(sensor_data_long$traveltime)
```

```
## [1] 0.2663316
```

```
library(imputeTS)
```

```
## Warning: package 'imputeTS' was built under R version 3.6.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
sensor_data_long <-na_kalman(sensor_data_long)
```

Filtering Pacific Motorway Links

The data contain 219 pair of sensors, which is a very large dataset. So we are only considering the pacific motoway links for the traffic data analysis

```
list_ids <- c("1142-1134","1129-1260","1131-1129","1131-1125",
              "1134-1178","1125-1131","1134-1142","1135-1231",
              "1189-1178","1129-1131","1178-1134","1261-1129",
              "1142-1125","1135-1189","1178-1189","1189-1135",
              "1125-1142","1231-1135","1260-1261")
pacific_motorway_data <- subset(sensor_data_long, link %in% list_ids)
```

Filtering data from mng 7.00am

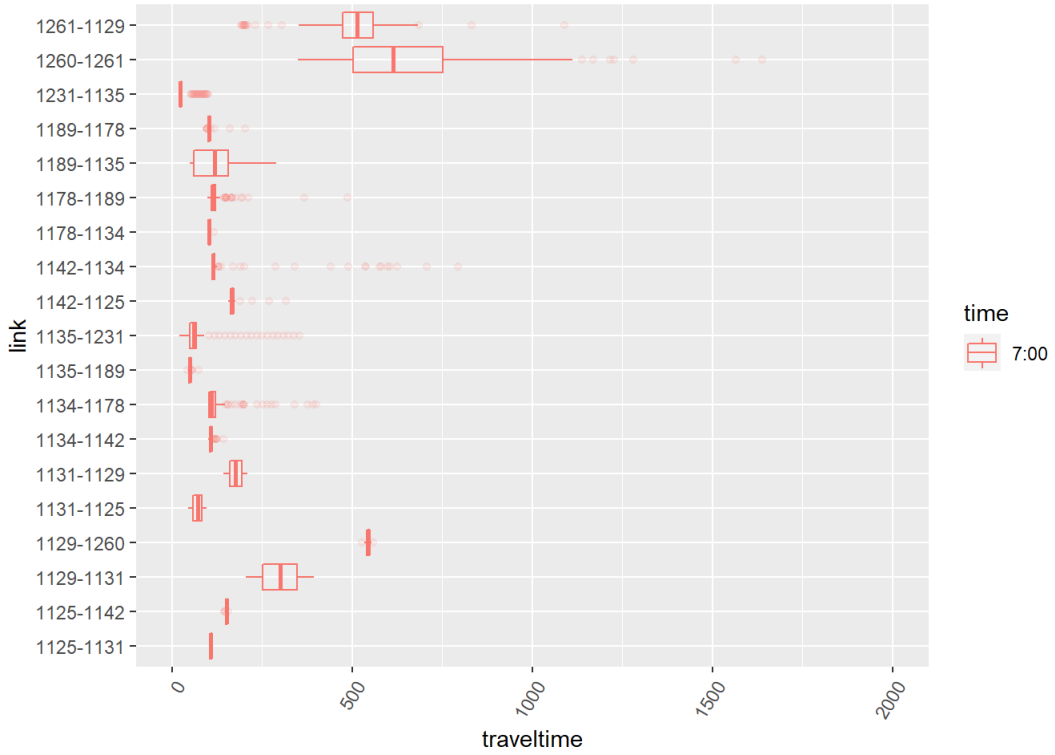
The average travel time varies at different time intervals. Here, only 7.00 am data is filtered for the analysis.

```
pacific_motorway_data$interval = pacific_motorway_data$INTERVAL_END
pacific_motorway_data <- tidyr::separate(pacific_motorway_data, interval, c("date", "time"), sep = " ")
```

```
time_peak <- c("7:00")
pacific_motorway_data <- subset(pacific_motorway_data, time %in% time_peak)
```

Exploratory Data Analysis

```
#Interval End Vs travel time
ggplot(data = pacific_motorway_data, mapping = aes(x = traveltime, y = link)) +
  geom_boxplot(alpha = 0.1, aes(color = time))+
  scale_x_continuous(limits = c(0,2000))+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```




```
#scale_y_log10()
```

Normalising the data

Standardization (Normalization) refers to the process of rescaling the values of the attributes in a dataset to develop a common scale where the values of the variables are different from one another.Clustering is sensitive to inputs on different scale.Here we are using Euclidian distance to calculate the distance between the data points in order to find the cluster centres. With inputs on different scale, Euclidian distance favors features on larger scale. So, we performed scaling before performing clustering, to normalise the variables.

```
# Long to wide format
pacific_motorway_data <- pacific_motorway_data[-c(4:5)]

pacific_motorway_data <- spread(pacific_motorway_data, link, traveltime)
pacific_motorway_data_original <- pacific_motorway_data

data <- tibble::rowid_to_column(pacific_motorway_data, "No.")

pacific_motorway_data <- data[-c(1,2)]
```

```
#Standerdisation
i <- c(1:19)
pacific_motorway_data[ ,i] <- apply(pacific_motorway_data[ ,i], 2,          function(x) as.numeric(as.charac
ter(x)))
sapply(pacific_motorway_data, class)
```

```
## 1125-1131 1125-1142 1129-1131 1129-1260 1131-1125 1131-1129 1134-1142
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
## 1134-1178 1135-1189 1135-1231 1142-1125 1142-1134 1178-1134 1178-1189
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
## 1189-1135 1189-1178 1231-1135 1260-1261 1261-1129
## "numeric" "numeric" "numeric" "numeric" "numeric"
```

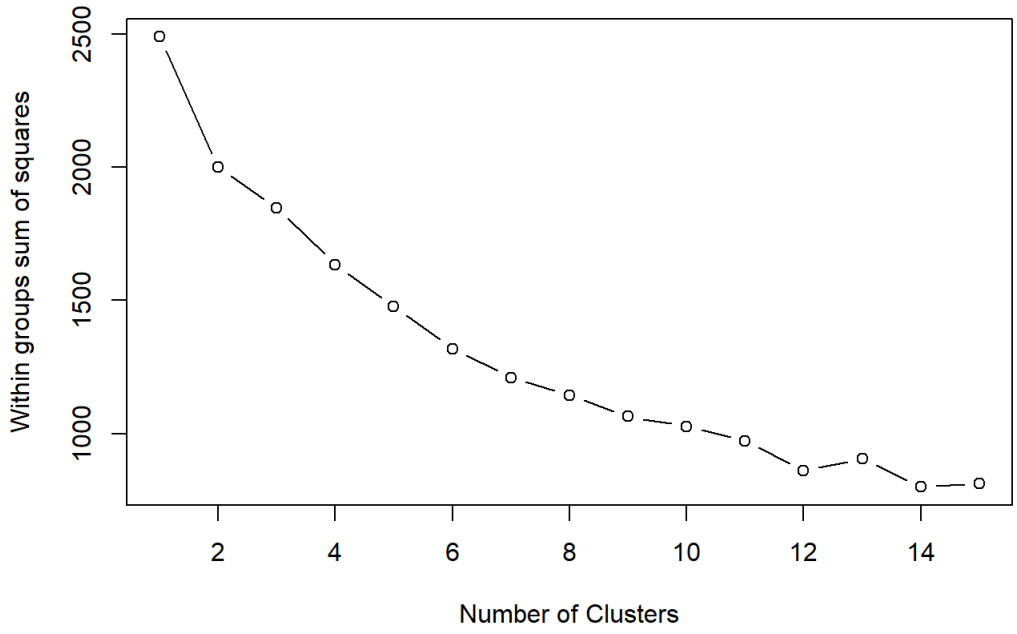
```
pacific_motorway_dataN <- as.data.frame(scale(pacific_motorway_data))
```

Determing the optimal number of clusters

1. Elbow Method

The elbow method maps the within-cluster sum of squares onto the number of possible clusters. As a rule of thumb, pick the number for which you see a significant decrease in the within-cluster dissimilarity, or so called ‘elbow

```
wss <- (nrow(pacific_motorway_dataN)-1)*sum(apply(pacific_motorway_dataN,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(pacific_motorway_dataN,
                                     centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```



2. Silhouette Method

The silhouette plots display a measure of how close each point in one cluster is to points in the neighboring clusters. This measure ranges from -1 to 1, where 1 means that points are very close to their own cluster and far from other clusters and -1 indicates that points are close

to the neighbouring clusters.

```
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 3.6.3
```

```
pamk.best2 <- pamk(pacific_motorway_dataN)
cat("number of clusters estimated by optimum average silhouette width:", pamk.best2$nc, "\n")
```

```
## number of clusters estimated by optimum average silhouette width: 10
```

3. Gap Statistics

Gap statistic is a goodness of clustering measure, where 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.

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.6.3
```

```
clusGap(pacific_motorway_dataN, kmeans, 10, B = 100, verbose = interactive())
```

```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = pacific_motorway_dataN, FUNcluster = kmeans, K.max = 10,     B = 100, verbose = interactive()
)
## B=100 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
##  --> Number of clusters (method 'firstSEmax', SE.factor=1): 8
##           logW   E.logW       gap     SE.sim
##  [1,] 5.228332 5.968475 0.7401433 0.01001746
##  [2,] 5.095746 5.892916 0.7971697 0.01092244
##  [3,] 5.014378 5.841975 0.8275967 0.01179792
##  [4,] 4.967842 5.801012 0.8331704 0.01229583
##  [5,] 4.926819 5.768084 0.8412650 0.01195745
##  [6,] 4.871655 5.740176 0.8685213 0.01230266
##  [7,] 4.829381 5.713880 0.8844992 0.01216728
##  [8,] 4.790698 5.689708 0.8990102 0.01183884
##  [9,] 4.786505 5.667521 0.8810153 0.01140572
## [10,] 4.742477 5.645810 0.9033333 0.01077620
```

```
pacific_motorway_data_clustermean<-pacific_motorway_dataN
```

Clustering

Clustering means to separate the datasets into different groups. Where the data points in the same group are similar to the other data points and dissimilar to the data points in other groups. K-means clustering will start by picking random points as the initial cluster centers (centroids)and for each iteration of K-means, all objects are assigned to the closest centroids. Each centroids are then updated to get closer to the mean of each cluster and this process continuous until the value of cebntroid becomes constant. Thus it helps in determining the size of each cluster and the cluster means.

```
pacific_motorway_data_cluster <- kmeans(pacific_motorway_dataN, centers = 6, nstart = 25)
print(pacific_motorway_data_cluster)
```

```
## K-means clustering with 6 clusters of sizes 46, 14, 1, 39, 18, 14
##
## Cluster means:
##    1125-1131   1125-1142  1129-1131   1129-1260  1131-1125  1131-1129
## 1 -0.8911793  0.03557813 -0.8911793  0.09419042  0.8911793  0.8911793
## 2 -1.0513573 -0.69378299 -1.0513573 -1.31031682  1.0513573  1.0513573
## 3  1.0327714  2.98165708  1.0327714  0.15795213 -1.0327714 -1.0327714
## 4  0.6103658  0.30813682  0.6103658  0.11592672 -0.6103658 -0.6103658
## 5  1.4902488 -0.30673806  1.4902488  0.35083465 -1.4902488 -1.4902488
## 6  0.2894095 -0.10009571  0.2894095  0.21553988 -0.2894095 -0.2894095
##      1134-1142   1134-1178   1135-1189  1135-1231   1142-1125  1142-1134
## 1 -0.419611962 -0.22245803 -0.08471688 -0.2204876 -0.36866605 -0.3250101
## 2 -0.412207844 -0.44750462 -0.07339121 -0.6332633 -0.25570811 -0.3438773
## 3 -0.009413426 -0.05581511  9.05607104 -0.2414930  0.23189133 -0.3230785
## 4  0.436674480 -0.15731316  0.04550178 -0.4174214  0.50278409 -0.3226578
## 5  0.047632614  0.05232308 -0.38081589  2.2090517  0.01355463  0.1053061
## 6  0.513912979  1.55338193  0.06774991 -0.3024207  0.03243555  2.1982834
##     1178-1134   1178-1189   1189-1135   1189-1178  1231-1135   1260-1261
## 1  0.32025117 -0.03581268  0.53781758 -0.12571093 -0.4471256  0.35612769
## 2 -0.58295131 -0.39393162 -0.97593268 -0.23786186 -0.4577945 -1.16459740
## 3 -0.37149006  8.45358043 -1.25009528  5.41613688 -0.4276836 -0.01985581
## 4 -0.31260357 -0.13034747 -0.11421565 -0.07435298 -0.2631644 -0.13338168
## 5  0.01293065  0.07259700 -0.23464264 -0.15835429  2.3346330  0.10301783
## 6  0.41143159  0.17754649 -0.08203414  0.67476969 -0.3111000  0.23499360
##     1261-1129
## 1  0.56110540
## 2 -1.82872500
## 3 -0.18657154
## 4  0.05992666
## 5 -0.24963112
## 6  0.15243525
##
## Clustering vector:
##   [1] 4 4 1 2 5 6 6 1 5 6 1 1 1 5 4 4 1 1 4 6 1 1 4 6 1 5 4 1 1 5 4 6 1 1 5
##  [36] 4 1 1 5 4 1 1 1 5 4 4 1 2 4 4 4 2 1 4 6 1 2 6 1 1 5 4 1 1 5 4 4 6 2 5
##  [71] 4 1 2 5 4 4 1 2 5 4 4 1 2 3 4 1 2 4 4 1 5 4 1 2 6 1 1 1 5 4 4 4 2 5 4
## [106] 2 4 4 2 1 1 4 4 6 1 1 4 4 1 1 6 2 1 5 6 1 1 5 4 6 1 1
##
## Within cluster sum of squares by cluster:
## [1] 353.60159  68.89776   0.00000 406.04740 199.84971 268.08659
##  (between_SS / total_SS =  47.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

Plotting

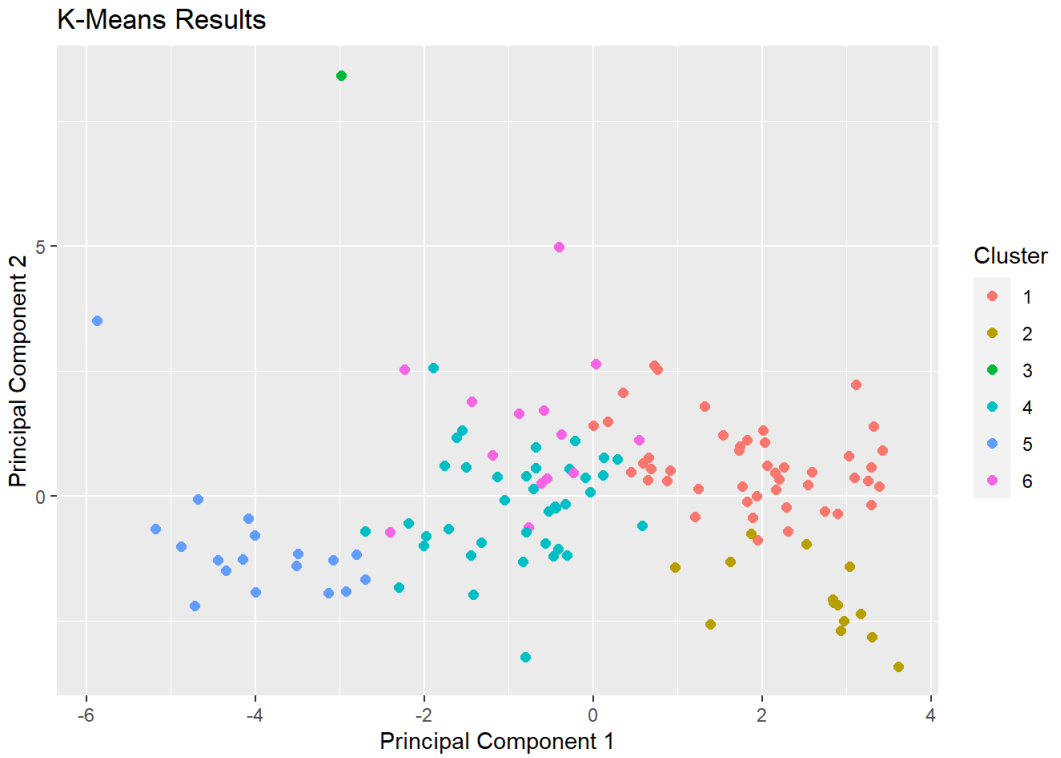
The result of K-means clustering can be difficult to plot because of the high dimensional nature of the data. To overcome this, the plot.kmeans function performs multidimensional scaling to project the data into two dimensions and then color codes the points according to cluster membership. Where the data in the first dimesion show the variables with a high variance and the variables with less variance in the Y axis.

```
library(useful)
```

```
## Warning: package 'useful' was built under R version 3.6.3
```

```
## Registered S3 methods overwritten by 'useful':
##   method       from
##   autoplot.acf forecast
##   fortify.ts   forecast
```

```
plot(pacific_motorway_data_cluster, data=pacific_motorway_dataN)
```



Finding the mean of the clusters.

Determining the mean of each cluster, to perform an exploratory analysis on the basis of the clustered data to interpret the result.

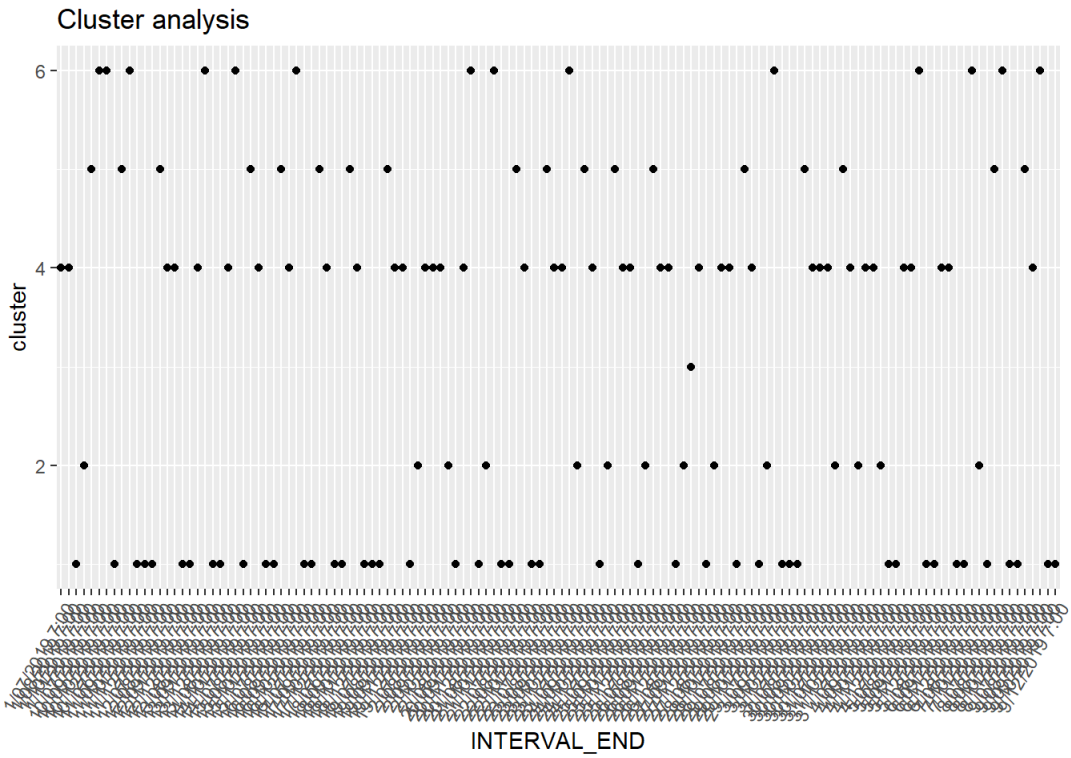
```
cluster_mean <- aggregate(pacific_motorway_data, by=list(cluster=pacific_motorway_data_cluster$cluster), mea
n)
```

```
pacific_motorway_data_withclusters <- cbind(pacific_motorway_dataN, cluster = pacific_motorway_data_cluster$
cluster)

pacific_motorway_data_withclusters$INTERVAL_END <- pacific_motorway_data_original$INTERVAL_END

pacific_motorway_data_withclusters_long <- pacific_motorway_data_withclusters %>% gather(link, traveltime, -
c(INTERVAL_END,cluster))
```

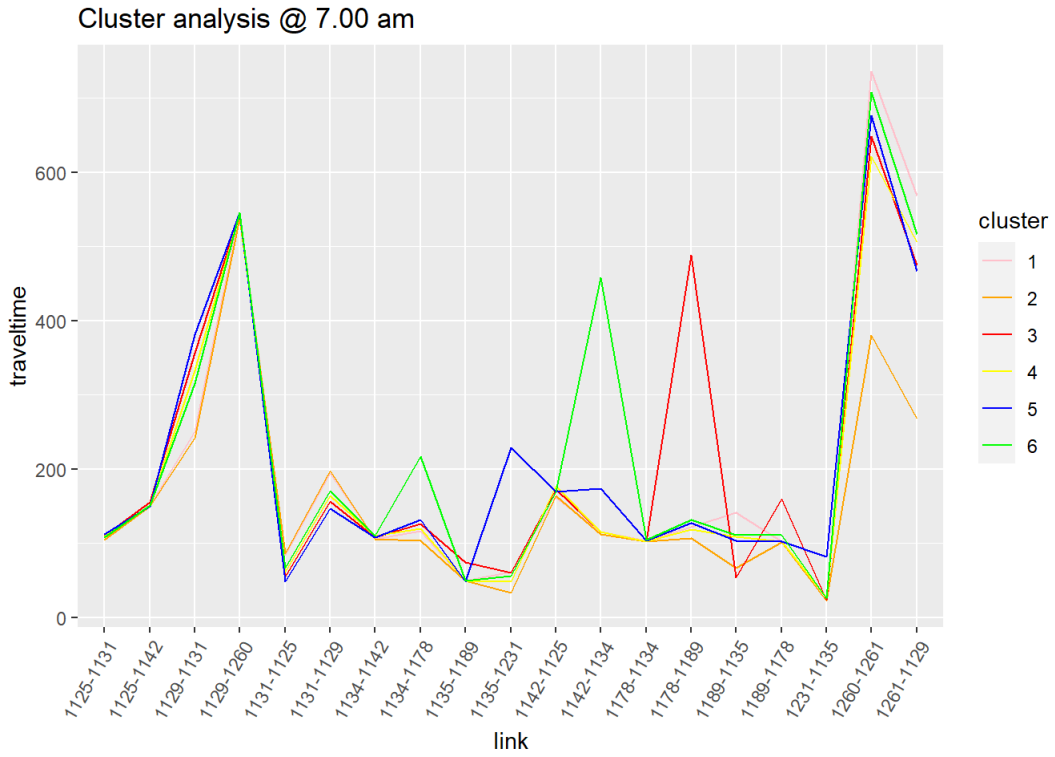
```
cluster_mean_long <- cluster_mean %>% gather(link, traveltime, -c(cluster))
```

Visualising the clustering results

```
#Cluster Vs Interval
#Cluster Vs Interval
ggplot(pacific_motorway_data_withclusters_long, aes(INTERVAL_END,cluster)) +
  geom_point()+
  ggtitle("Cluster analysis")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```




```
cluster_mean_long$cluster <- as.factor(cluster_mean_long$cluster)

#Cluster Vs Link
ggplot(cluster_mean_long, aes(link,traveltime)) +
  geom_line(aes(group=cluster,colour=cluster,))+
  scale_colour_manual(values = c('pink','orange','red',
                                 'yellow','blue','green'))+
  ggtitle("Cluster analysis @ 7.00 am")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```

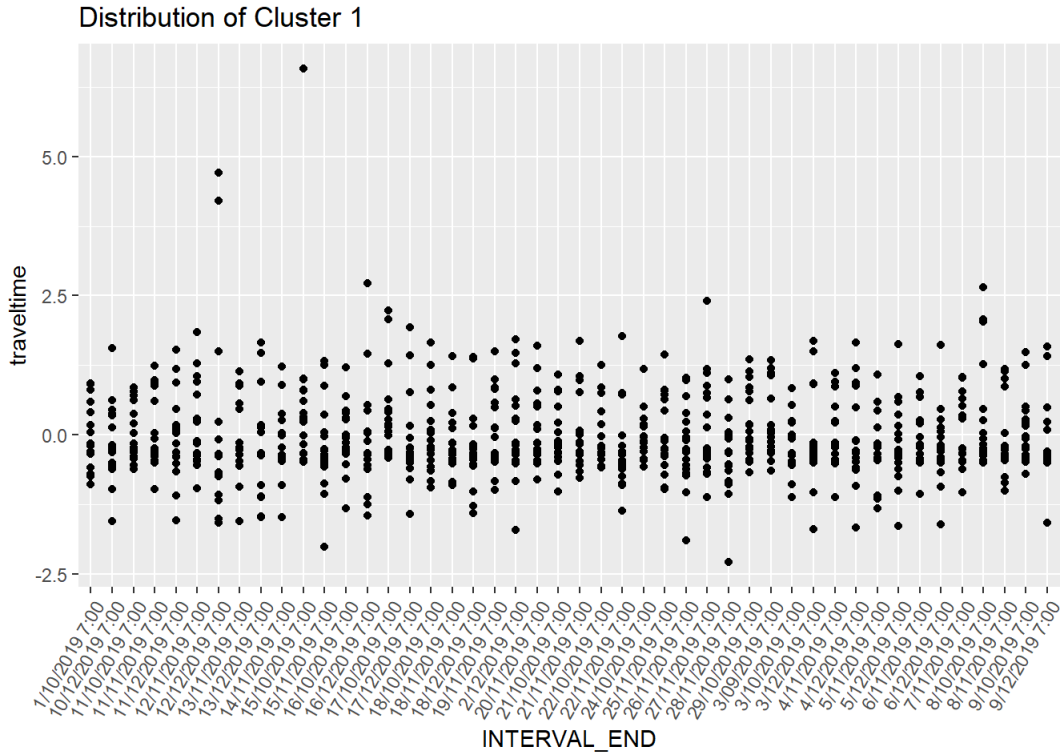


Distribution of each cluster

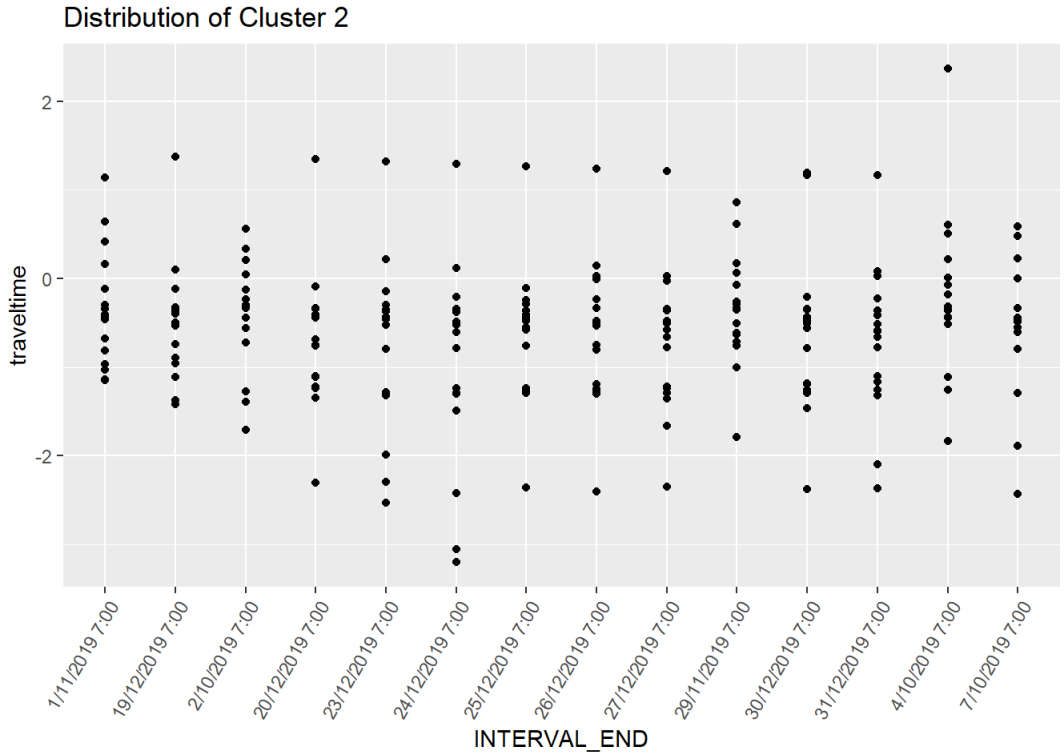
This shows which are the days that are in each of the cluster.

```
cluster_1 <- filter(pacific_motorway_data_withclusters_long, cluster=="1")
cluster_2 <- filter(pacific_motorway_data_withclusters_long, cluster=="2")
cluster_3 <- filter(pacific_motorway_data_withclusters_long, cluster=="3")
cluster_4 <- filter(pacific_motorway_data_withclusters_long, cluster=="4")
cluster_5 <- filter(pacific_motorway_data_withclusters_long, cluster=="5")
cluster_6 <- filter(pacific_motorway_data_withclusters_long, cluster=="6")
```

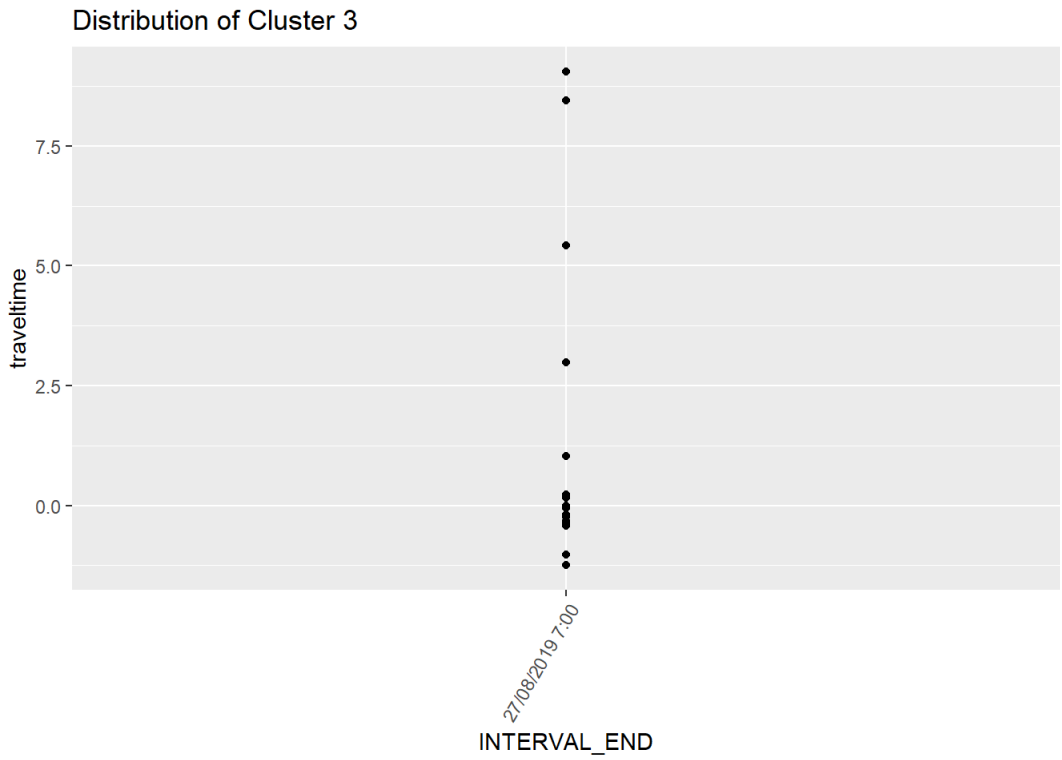
```
cluster_1 %>%
  ggplot(aes(INTERVAL_END,traveltime)) + geom_point() +
ggtitle("Distribution of Cluster 1")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



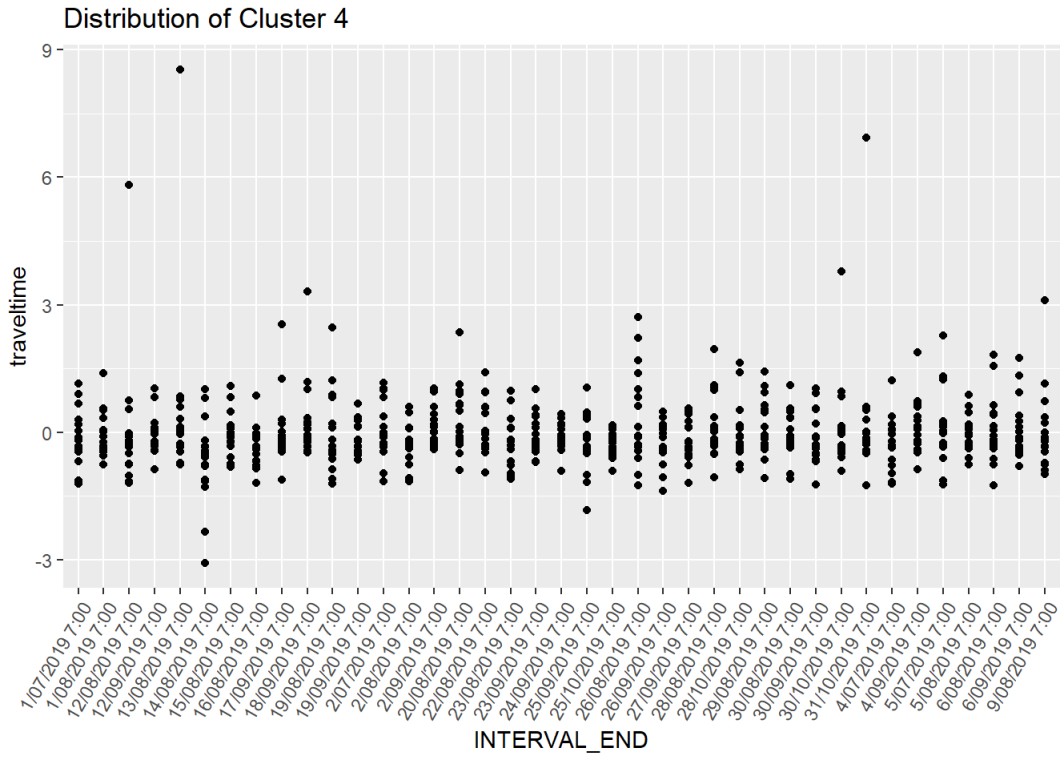

```
cluster_2 %>%
  ggplot(aes(INTERVAL_END,traveltime)) + geom_point() +
ggtitle("Distribution of Cluster 2")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



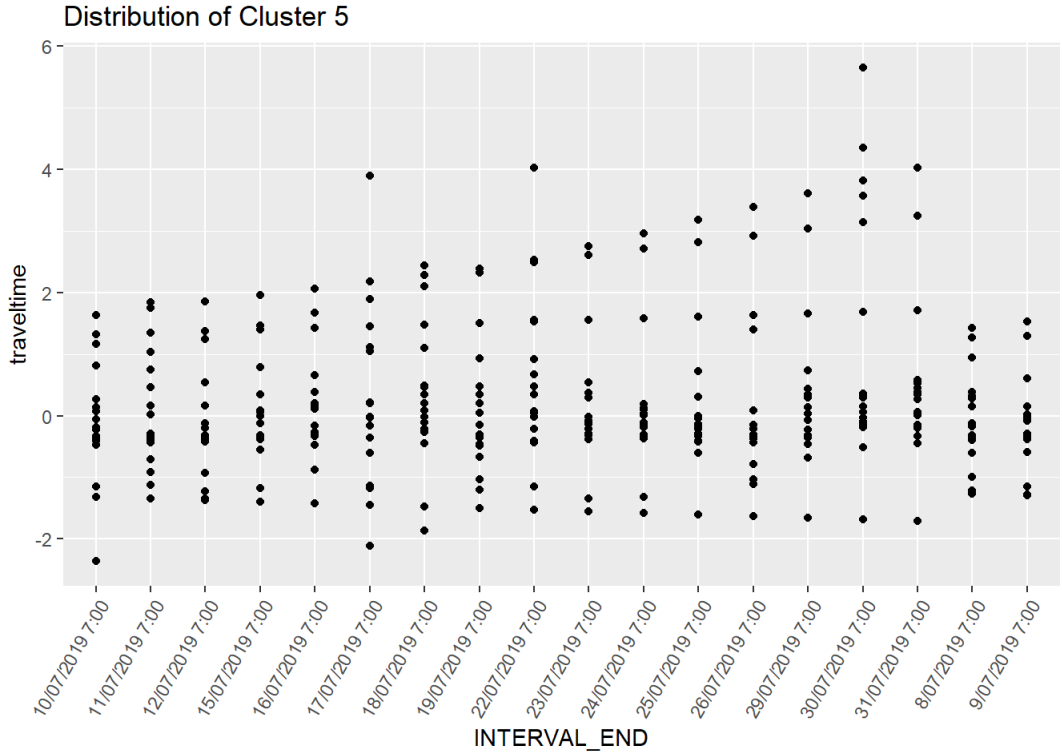
```
cluster_3 %>%
  ggplot(aes(INTERVAL_END,traveltime)) + geom_point() +
ggtitle("Distribution of Cluster 3")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



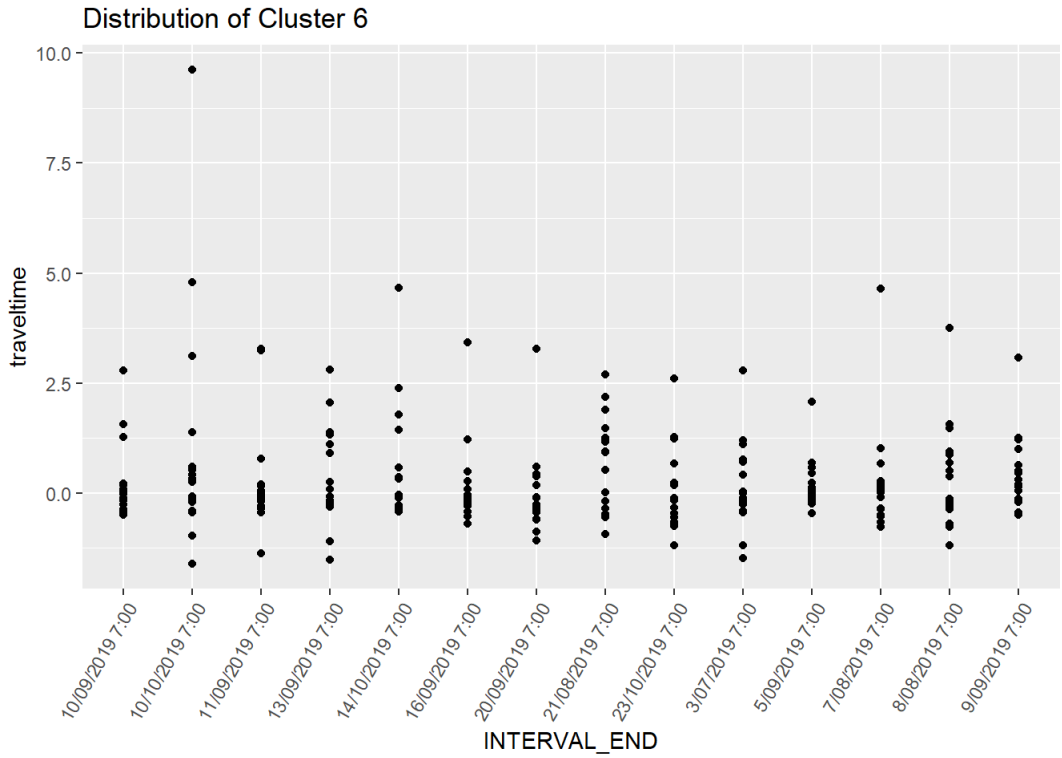

```
cluster_4 %>%
  ggplot(aes(INTERVAL_END,traveltime)) + geom_point() +
ggtitle("Distribution of Cluster 4")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```




```
cluster_5 %>%
  ggplot(aes(INTERVAL_END,traveltime)) + geom_point() +
ggtitle("Distribution of Cluster 5")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```




```
cluster_6 %>%
  ggplot(aes(INTERVAL_END,traveltime)) + geom_point() +
ggtitle("Distribution of Cluster 6")+theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



Note: To compare the result of morning 7.00AM traffic data, 7.15 AM and 8.00 AM traffic data was also plotted using the same code.