K-means implementation in R

In this lecture

- Case study
  - Problem statement
- Solve the case study using R
  - Read the data from a ".csv" file
  - Understand the data
  - k-means() function
  - Interpret the results

Clustering of trips: a case study

Clustering of trips: Problem statement

An Uber cab driver has attended 91 Trips in a week (5 days). He has a facility which continuously monitors the following parameters for each trip

Trip length, Max speed, Most frequent speed, Trip duration, number of times brakes are used, idling time and number times the horn is being honked.

Uber wants to group the trips in to certain number of categories based on the details collected during the trip for some business plan. They have consulted Mr. Sam, a data scientist to perform this job and the details of trips are shared in a ".csv" format file with name "tripDetails.csv"

Solution to case study using R

Getting things ready

- Setting working directory, clearing variables in the workspace

```
##### k-means clustering #########
# Set the working directory as the directory
which contains the data files
# setwd("Path of the directory with data files")
rm(list=ls()) # to clear the environment
```

Reading the data

- Data for this case study is provided to you file with name "tripDetails.csv"
- To read the data from a ".csv" file we use `read.csv()` function

read.csv()

Reads a file in table format and creates a data frame from it

SYNTAX

```
read.csv(file,row.names=1)
```

file	the name of the file which the data are to be read from. Each row of the table appears as one line of the file.
row.names	a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names.

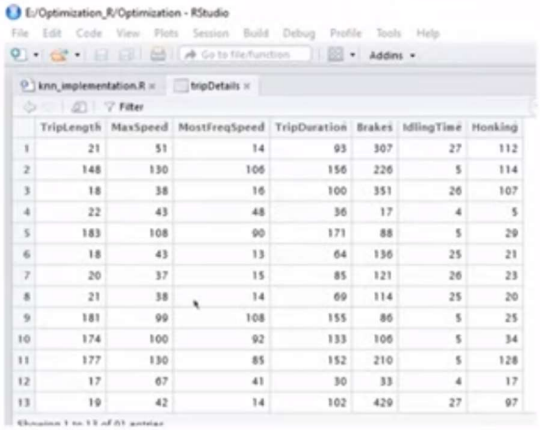
Reading the data

- Data for this case study is provided to you file with name "tripDetails.csv"

```
#Reading the data
tripDetails = read.csv("tripDetails.csv",
                       row.names=1)
```

Viewing the data

- View(tripDetails)



	TripLength	MaxSpeed	MostFreqSpeed	TripDuration	Brakes	IdlingTime	Honking
1	21	51	14	93	307	27	112
2	148	130	106	156	226	5	114
3	18	38	16	100	351	26	107
4	22	43	48	36	17	4	5
5	183	108	90	171	88	5	29
6	18	43	13	64	136	25	21
7	20	37	15	85	121	26	23
8	21	38	14	69	114	25	20
9	181	99	108	155	86	5	25
10	174	100	92	133	106	5	34
11	177	130	85	152	210	5	128
12	17	67	41	30	33	4	17
13	19	42	14	102	429	27	97

Understanding the data

Variables →

	Trip length	Max. Speed	Most Freq. speed	Trip duration	Brakes	Idling time	Honking
1	21	51	14	93	307	27	112
2	148	130	106	156	226	5	114
3	18	38	16	100	351	26	107
4	22	43	48	36	17	4	5
5	183	108	90	171	88	5	29
6	18	43	13	64	136	25	21
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...

Data contains 91 Trips where 7 variables (columns) named

Trip length, Max speed, Most Freq. speed, Trip duration, Brakes, Idling time and Honking are noted for each trip

91 observations

Structure of the data

- Structure of data
  - Variables and their data types
- `str()`

Compactly display the internal structure of an R object

SYNTAX

```
str(object)
```

object	any R object about which you want to have some information.

Structure of tripDetails

```
> str(tripDetails)
'data.frame':   91 obs. of  7 variables:
 $ TripLength   : int  21 148 18 22 183 18 20 21 181 174 ...
 $ MaxSpeed     : int  51 130 38 43 108 43 37 38 99 100 ...
 $ MostFreqSpeed: int  14 106 16 48 90 13 15 14 108 92 ...
 $ TripDuration : int  93 156 100 36 171 64 85 69 155 133 ...
 $ Brakes       : int  307 226 351 17 88 136 121 114 86 106 ...
 $ IdlingTime   : int  27 5 26 4 5 25 26 25 5 5 ...
 $ Honking      : int  112 114 107 5 29 21 23 20 25 34 ...
```

Summary of the data

- Summary of data
  - Five point summary of the numeric varaibles
- summary()

Summary is a generic function used to produce result summaries of the results of various model fitting functions and five point summaries of numeric R objects

SYNTAX

```
summary(object)
```

object	any R object about which you want to have some information.

Summary of tripDetails

```
> summary(tripDetails)
   TripLength         MaxSpeed      MostFreqSpeed
 Min.   : 16.00   Min.   : 35.00   Min.   : 12.00
 1st Qu.: 20.00   1st Qu.: 42.00   1st Qu.: 15.50
 Median : 21.00   Median : 54.00   Median : 42.00
 Mean   : 70.77   Mean   : 70.36   Mean   : 50.65
 3rd Qu.:163.00   3rd Qu.:105.50   3rd Qu.: 89.00
 Max.   :210.00   Max.   :138.00   Max.   :118.00
  TripDuration        Brakes        IdlingTime
 Min.   : 22.00   Min.   : 14.0   Min.   : 4.00
 1st Qu.: 34.50   1st Qu.: 36.5   1st Qu.: 5.00
 Median : 88.00   Median :100.0   Median : 5.00
 Mean   : 87.37   Mean   :135.4   Mean   :11.59
 3rd Qu.:133.00   3rd Qu.:198.0   3rd Qu.:24.00
 Max.   :171.00   Max.   :429.0   Max.   :32.00
    Honking
 Min.   :  4.00
 1st Qu.: 20.00
 Median : 25.00
 Mean   : 49.92
 3rd Qu.: 97.50
 Max.   :155.00
```

K-means clustering

- Given the dataset of trip details, Mr. Sam's job is to segregate these trips into clusters
  - We seek an answer through k-means clustering
- Using k-means clustering on data
  - k-means clustering in R can be applied on data using "`kmeans()`" function

kmeans()

```
object = kmeans(x, centers, iter.max = 10, nstart = 1)
```

Arguments

x	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
centers	either the number of clusters, say k, or a set of initial (distinct) cluster centers. If a number, a random set of (distinct) rows in x is chosen as the initial centers.
iter.max	the maximum number of iterations allowed.
nstart	if centers is a number, how many random sets should be chosen?
object	an R object of class "kmeans", typically the result "ob" of ob < − kmeans(..).

Implementing K-means

- Clustering data using k-means and seeing the clusters details

```
# k-means clustering using kmeans
command
tripCluster <- kmeans(tripDetails,3)
```

Results

tripCluster has the following information

```
> tripCluster
K-means clustering with 3 clusters of sizes 46, 15, 30

Cluster means:
  TripLength  MaxSpeed MostFreqSpeed TripDuration    Brakes
1   19.91304  48.21739      32.82609     50.13043  59.93478
2   20.26667  45.06667      14.46667     88.73333 350.13333
3  174.00000 116.96667      96.06667    143.80000 143.86667
  IdlingTime  Honking
1  11.413043 15.60870
2  25.400000 97.73333
3   4.966667 78.63333

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
 2  3  2  1  3  1  1  1  3  3  3  1  2  1  1  3  1  1  2  1  3
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
 1  2  2  2  1  1  2  3  3  1  1  1  3  1  3  3  3  1  2  3  1
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 1  1  3  1  2  1  3  1  1  1  1  1  2  3  1  1  3  1  1  1  3
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84
 3  2  1  1  1  3  1  1  2  1  1  1  3  3  3  1  1  3  3  3  3
85 86 87 88 89 90 91
 1  1  2  3  2  3  1
```

Data science for Engineers

Results

tripCluster has the following information

```
Within cluster sum of squares by cluster:
[1] 160740.2  25986.8 194647.0
 (between_SS / total_SS =  83.3 %)

Available components:

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

Results

tripCluster has the following information

```
> tripCluster
K-means clustering with 3 clusters of sizes 46, 15, 30

Cluster means:
  TripLength  MaxSpeed MostFreqSpeed TripDuration    Brakes
1   19.91304  48.21739      32.82609     50.13043  59.93478
2   20.26667  45.06667      14.46667     88.73333 350.13333
3  174.00000 116.96667      96.06667    143.80000 143.86667
  IdlingTime  Honking
1  11.413043 15.60870
2  25.400000 97.73333
3   4.966667 78.63333

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
 2  3  2  1  3  1  1  1  3  3  3  1  2  1  1  3  1  1  2  1  3
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
 1  2  2  2  1  1  2  3  3  1  1  1  3  1  3  3  3  1  2  3  1
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 1  1  3  1  2  1  3  1  1  1  1  1  2  3  1  1  3  1  1  1  3
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84
 3  2  1  1  1  3  1  1  2  1  1  1  3  3  3  1  1  3  3  3  3
85 86 87 88 89 90 91
 1  1  2  3  2  3  1
```

Results

tripCluster has the following information

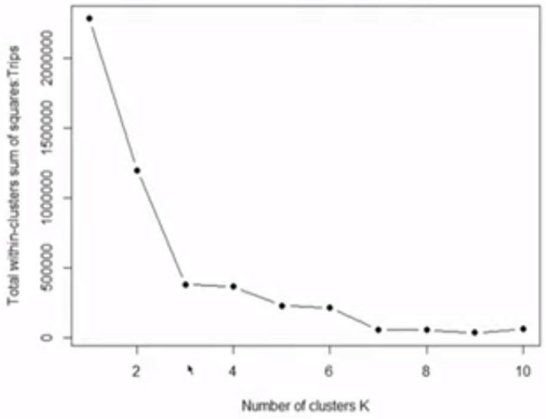
```
Within cluster sum of squares by cluster:
[1] 160740.2  25986.8 194647.0
 (between_SS / total_SS =  83.3 %)

Available components:

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

Results: k calculation

```
# Method to calculate optimal k
k.max <- 10 # Maximum 10 clusters assumed
wss <- rep(NA, k.max)
nClust <- list()
for (i in 1:k.max){
  driveClasses <- kmeans(tripDetails, i)
  wss[i] <- driveClasses$tot.withinss
  nClust[[i]] <- driveClasses$size
}
plot(1:k.max, wss,
     type="b", pch = 19,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares:Trips")
```

Data science for Engineers

Results: k calculation



K-means implementation in R

Summary

- K-means is an unsupervised algorithm
- `kmeans()`
- Elbow method