Kmeans Clustering

Kumar Rahul

23/05/2022

K means Clustering

More info at: https://uc-r.github.io/kmeans_clustering

```
#setwd("/Users/Rahul/Documents/Rahul Office/IIMB/Concepts/R/ML_using_R/R_code/R_clustering")
knitr::opts_knit$set(root.dir = "/Users/Rahul/Documents/Rahul Office/IIMB/Concepts/R/ML_using_R/R_code/
```

Preparing Data

Read data from a specified location

```
car_df = read.csv('./data/Kmeans_Car.csv',header = TRUE,sep = ",",
                  na.strings = c("","  ", "NA"), row.names = c(2))
```

```
head(car_df)
```

```
##                      Brand Price..INR. Mileage Seating.Capacity Vehicle.Type
## Tata Nano Std BSIII  Tata      141898    25.4                4    Hatchback
## Tata Nano Std        Tata      145000    25.4                4    Hatchback
## Tata Nano 2013 STD   Tata      150000    25.4                4    Hatchback
## Tata Nano Cx BSIII   Tata      171489    25.4                4    Hatchback
## Tata Nano Cx         Tata      191125    25.4                4    Hatchback
## Tata Nano 2013 CX    Tata      198000    25.4                4    Hatchback
##                      Fuel.Type Transmission Parking.Sensor Airbag Cruise.Control
## Tata Nano Std BSIII    Petrol       Manual             No     No             No
## Tata Nano Std          Petrol       Manual             No     No             No
## Tata Nano 2013 STD     Petrol       Manual             No     No             No
## Tata Nano Cx BSIII     Petrol       Manual             No     No             No
## Tata Nano Cx           Petrol       Manual             No     No             No
## Tata Nano 2013 CX      Petrol       Manual             No     No             No
##                      Keyless.Entry Alloy.wheels ABS Climate.Control Rear.AC.Vent
## Tata Nano Std BSIII             No           No  No              No           No
## Tata Nano Std                   No           No  No              No           No
## Tata Nano 2013 STD              No           No  No              No           No
## Tata Nano Cx BSIII              No           No  No              No           No
## Tata Nano Cx                    No           No  No              No           No
## Tata Nano 2013 CX               No           No  No              No           No
##                      Power.Steering
## Tata Nano Std BSIII              No
## Tata Nano Std                    No
## Tata Nano 2013 STD               No
## Tata Nano Cx BSIII               No
## Tata Nano Cx                     No
## Tata Nano 2013 CX                No
```

Summary of the data

Summary of the data on which model is built and Standardizing the variables

```
str(car_df)
```

```
## 'data.frame':    1008 obs. of  16 variables:
##  $ Brand           : chr  "Tata" "Tata" "Tata" "Tata" ...
##  $ Price..INR.     : int  141898 145000 150000 171489 191125 198000 198605 216238 223500 237831 ...
##  $ Mileage         : num  25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4 ...
##  $ Seating.Capacity: int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Vehicle.Type    : chr  "Hatchback" "Hatchback" "Hatchback" "Hatchback" ...
##  $ Fuel.Type       : chr  "Petrol" "Petrol" "Petrol" "Petrol" ...
##  $ Transmission    : chr  "Manual" "Manual" "Manual" "Manual" ...
##  $ Parking.Sensor  : chr  "No" "No" "No" "No" ...
##  $ Airbag          : chr  "No" "No" "No" "No" ...
##  $ Cruise.Control  : chr  "No" "No" "No" "No" ...
##  $ Keyless.Entry   : chr  "No" "No" "No" "No" ...
##  $ Alloy.wheels    : chr  "No" "No" "No" "No" ...
##  $ ABS             : chr  "No" "No" "No" "No" ...
##  $ Climate.Control : chr  "No" "No" "No" "No" ...
##  $ Rear.AC.Vent    : chr  "No" "No" "No" "No" ...
##  $ Power.Steering  : chr  "No" "No" "No" "No" ...
```

```
car_df = na.omit(car_df)
car_scaled = scale(car_df[,c(2:4)])
```

Kmeans clustering - cluster package

The kmeans function has an nstart option that attempts multiple initial configurations and reports on the best one. We are setting nstart = 10 and thus it will generate 10 initial configurations. This approach is often recommended.

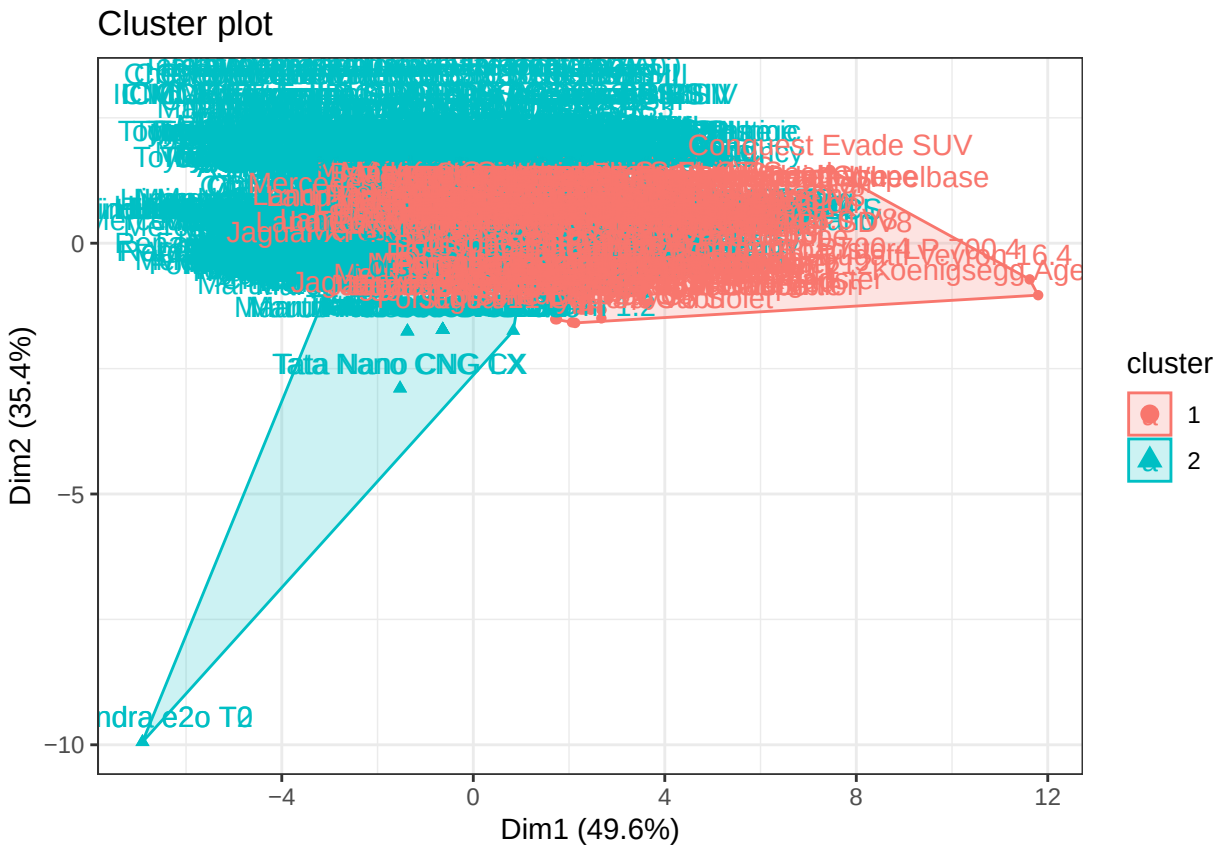
```
car_kmeans2 = kmeans(car_scaled, centers = 2, nstart = 10)
```

The output of kmeans is a list with several bits of information. The most important being:

- cluster: A vector of integers (from 1:k) indicating the cluster to which each point is allocated.
- centers: A matrix of cluster centers.
- totss: The total sum of squares.
- withinss: Vector of within-cluster sum of squares, one component per cluster.
- tot.withinss: Total within-cluster sum of squares, i.e. sum(withinss).
- betweenss: The between-cluster sum of squares, i.e. $totss - tot.withinss$.
- size: The number of points in each cluster.

```
options(repr.plot.width=15, repr.plot.height=10)
```

```
fviz_cluster(car_kmeans2, data = car_scaled) +
theme_bw()
```



Clusters with different K

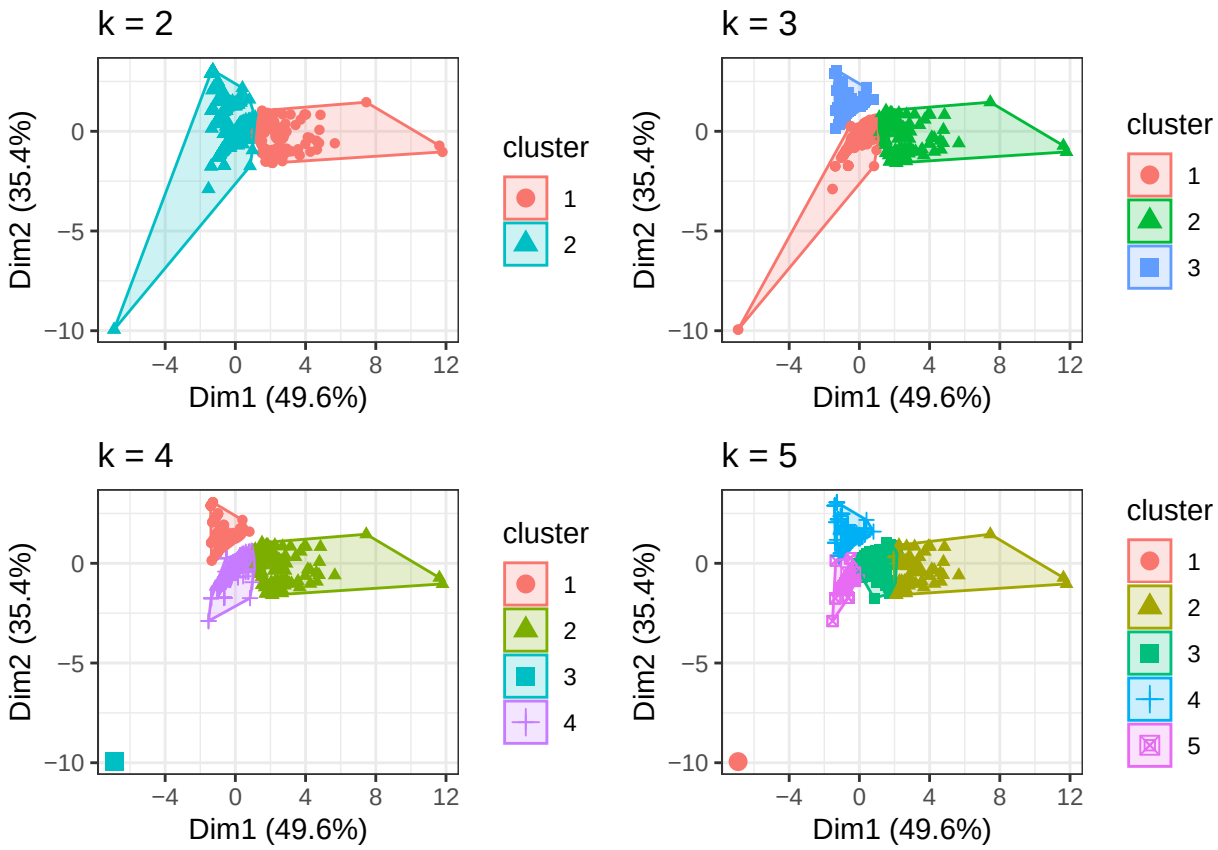
Using centers as 3,4,5 and visulaizing the clusters.

```
car_kmeans3 = kmeans(car_scaled, centers = 3, nstart = 10)
car_kmeans4 = kmeans(car_scaled, centers = 4, nstart = 10)
car_kmeans5 = kmeans(car_scaled, centers = 5, nstart = 10)

# plots to compare
p1 = fviz_cluster(car_kmeans2, geom = "point", data = car_scaled) + ggtitle("k = 2") + theme_bw()
p2 = fviz_cluster(car_kmeans3, geom = "point",  data = car_scaled) + ggtitle("k = 3")+ theme_bw()
p3 = fviz_cluster(car_kmeans4, geom = "point",  data = car_scaled) + ggtitle("k = 4")+ theme_bw()
p4 = fviz_cluster(car_kmeans5, geom = "point",  data = car_scaled) + ggtitle("k = 5")+ theme_bw()

library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```

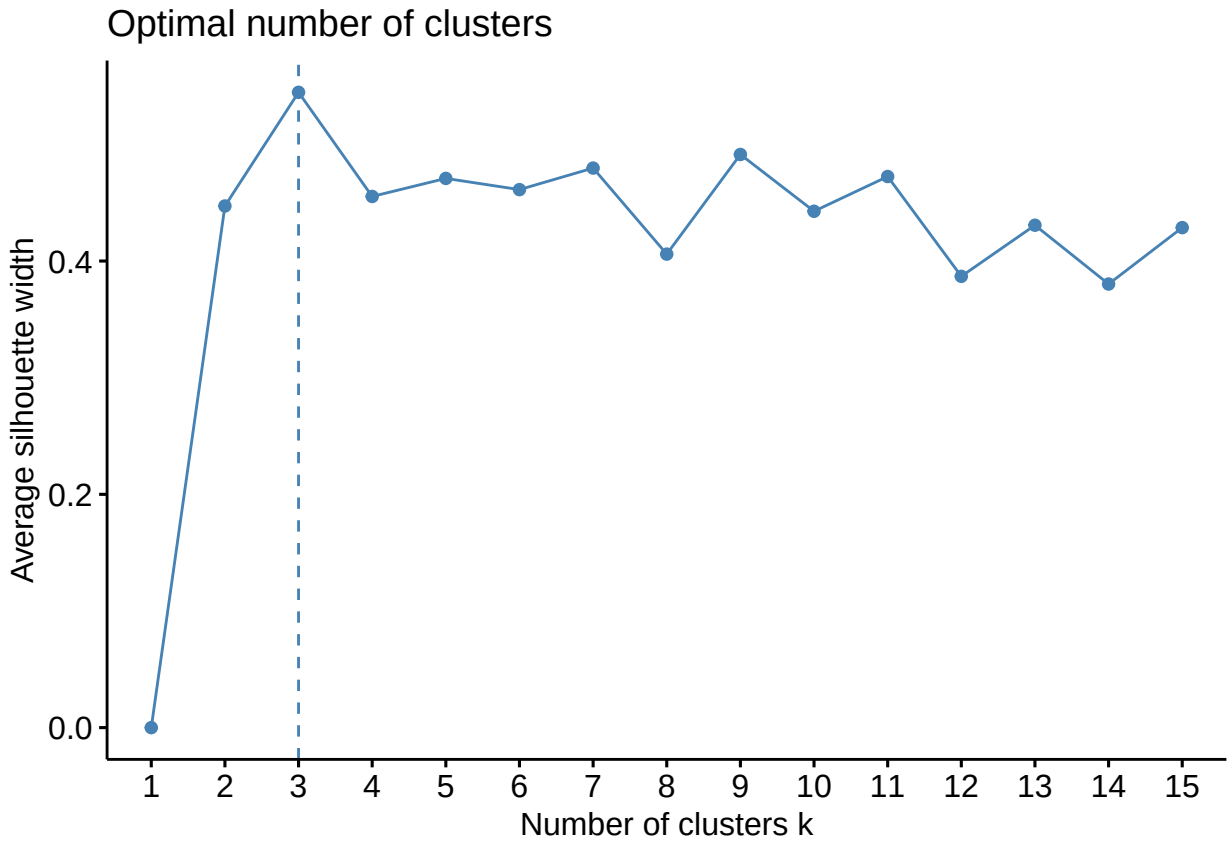


Optimal cluster - Average Silhouette width

Using average silhouette width as the metric, the optimal number of clusters k is the one that maximizes the average silhouette over a range of possible values for k.

The results show that 2 clusters maximize the average silhouette values with 3 clusters coming in as second optimal number of clusters.

```
fviz_nbclust(car_scaled, FUNcluster = kmeans,
             method = "silhouette", k=15)
```

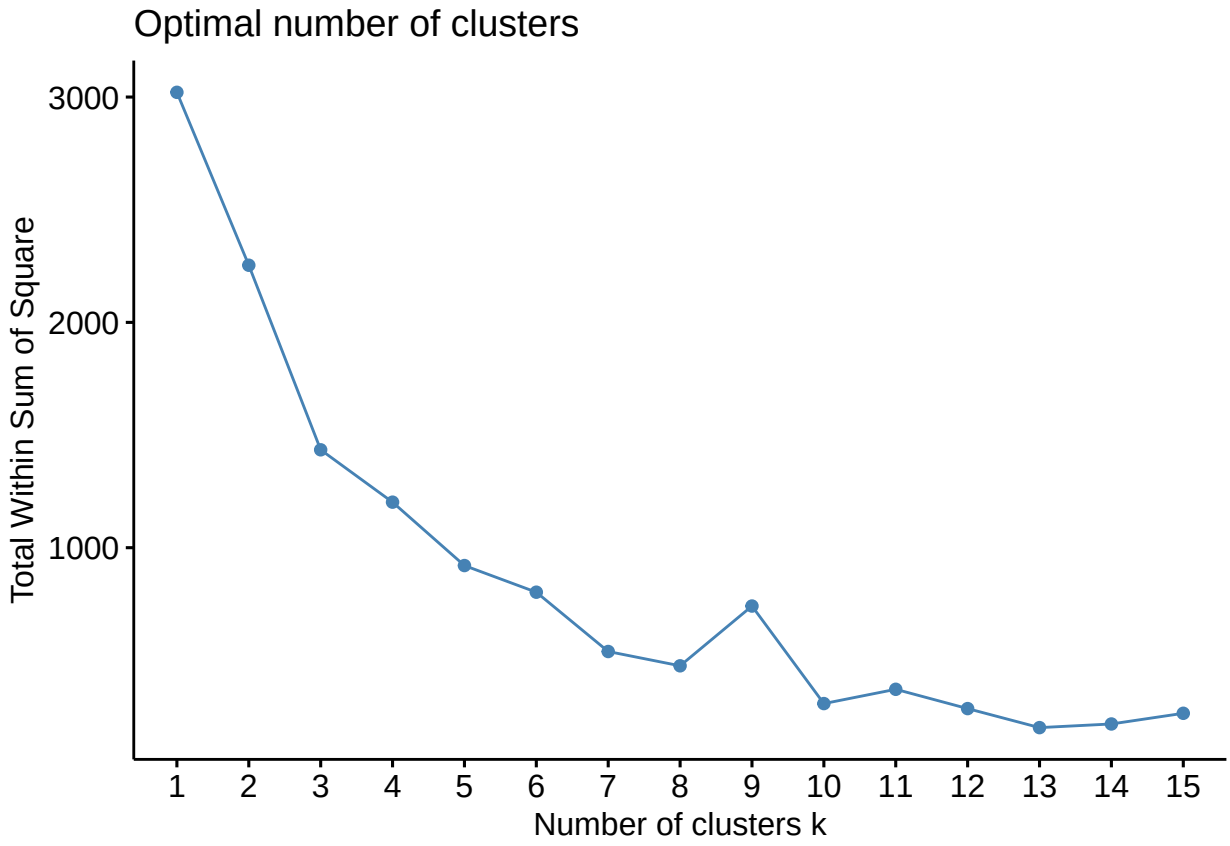


Optimal Cluster - Elbow method (wss)

The total within-cluster sum of square (wss) measures the compactness of the clustering and we want it to be as small as possible. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

Result shows k = 3 to be the optimal.

```
fviz_nbclust(car_scaled, FUNcluster = kmeans, method = "wss", k=15)
```



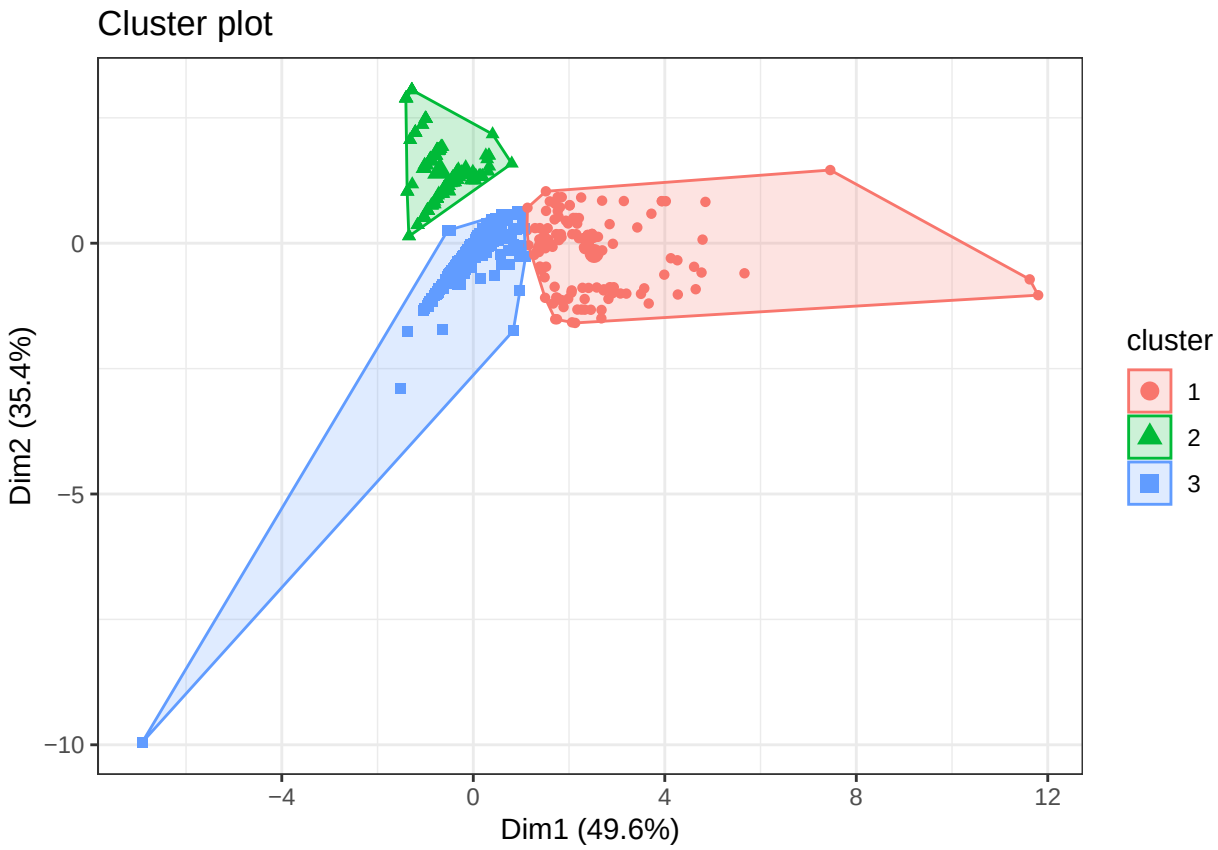
In order to identify sub-groups (i.e. clusters), we will set k =3

```
set.seed(123)
car_kmeans_final = kmeans(car_scaled, 3, nstart = 10)
#print(car_kmeans_final)
```

Visualize final cluster

We can visualize the cluster as below:

```
fviz_cluster(car_kmeans_final, geom="point", data = car_scaled) +
theme_bw()
```



The summary staistics of the 3 clsuters are as below:

```
car_df[,c(2:4)] %>%
  mutate(Cluster = car_kmeans_final$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 3 x 4
##   Cluster Price..INR. Mileage Seating.Capacity
##     <int>       <dbl>   <dbl>            <dbl>
## 1       1   19841783.    9.06             3.52
## 2       2    1513744.   13.8              7.57
## 3       3    1457427.   18.3              4.96
```

The cluster can be tagged to the original dataframe as below:

```
car_df %>%
mutate(cluster = car_kmeans_final$cluster, name = row.names(car_df)) %>%
head(10)
```

```
##    Brand Price..INR. Mileage Seating.Capacity Vehicle.Type Fuel.Type
## 1   Tata      141898    25.4                4    Hatchback    Petrol
## 2   Tata      145000    25.4                4    Hatchback    Petrol
## 3   Tata      150000    25.4                4    Hatchback    Petrol
## 4   Tata      171489    25.4                4    Hatchback    Petrol
## 5   Tata      191125    25.4                4    Hatchback    Petrol
## 6   Tata      198000    25.4                4    Hatchback    Petrol
## 7   Tata      198605    25.4                4    Hatchback    Petrol
## 8   Tata      216238    25.4                4    Hatchback    Petrol
```

```
## 9   Tata      223500    25.4               4    Hatchback    Petrol
## 10  Tata      237831    25.4               4    Hatchback    Petrol
##    Transmission Parking.Sensor Airbag Cruise.Control Keyless.Entry Alloy.wheels
## 1        Manual             No     No             No            No           No
## 2        Manual             No     No             No            No           No
## 3        Manual             No     No             No            No           No
## 4        Manual             No     No             No            No           No
## 5        Manual             No     No             No            No           No
## 6        Manual             No     No             No            No           No
## 7        Manual             No     No             No            No           No
## 8        Manual             No     No             No           Yes           No
## 9        Manual             No     No             No           Yes           No
## 10       Manual             No     No             No           Yes           No
##    ABS Climate.Control Rear.AC.Vent Power.Steering cluster               name
## 1   No              No           No              No       3 Tata Nano Std BSIII
## 2   No              No           No              No       3       Tata Nano Std
## 3   No              No           No              No       3  Tata Nano 2013 STD
## 4   No              No           No              No       3  Tata Nano Cx BSIII
## 5   No              No           No              No       3        Tata Nano Cx
## 6   No              No           No              No       3   Tata Nano 2013 CX
## 7   No              No           No              No       3  Tata Nano Lx BSIII
## 8   No              No           No              No       3        Tata Nano Lx
## 9   No              No           No              No       3   Tata Nano 2013 LX
## 10  No              No           No             Yes       3  Tata Nano Twist XT
```

```
car_df %>%
mutate(cluster = car_kmeans_final$cluster, name = row.names(car_df)) %>%
write.csv("cars_kmeans_output.csv")
```