Non-Hierarchical Clustering – II

K Means Method

Contents

1. K-Means Clustering in R
2. Elbow Method to Select K
3. K-Median Clustering in R

Case Study

Background

- A FMCG company has recorded information of customers based on their buying behaviour for a period of 1 year and would like to implement strategies by segmenting these customers into tiers.

Objective

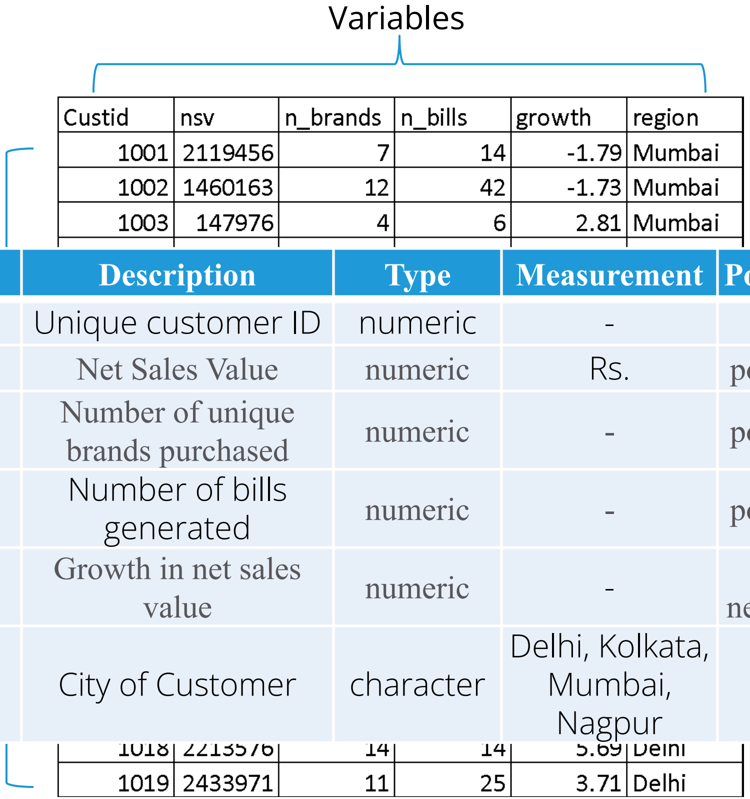
- To create segment of customers.

Available Information

- Sample size is 1158.
- Variables : Custid, nsv, n_brands , n_bills, growth, region

Data Snapshot

RETAILERS DATA



Variables

Custid	nsv	n_brands	n_bills	growth	region
1001	2119456	7	14	-1.79	Mumbai
1002	1460163	12	42	-1.73	Mumbai
1003	147976	4	6	2.81	Mumbai
1018	2213576	14	14	5.69	Delhi
1019	2433971	11	25	3.71	Delhi

Columns	Description	Type	Measurement	Possible values
Custid	Unique customer ID	numeric	-	-
nsv	Net Sales Value	numeric	Rs.	positive values
n_brands	Number of unique brands purchased	numeric	-	positive values
n_bills	Number of bills generated	numeric	-	positive values
growth	Growth in net sales value	numeric	-	Positive & negative values
region	City of Customer	character	Delhi, Kolkata, Mumbai, Nagpur	4

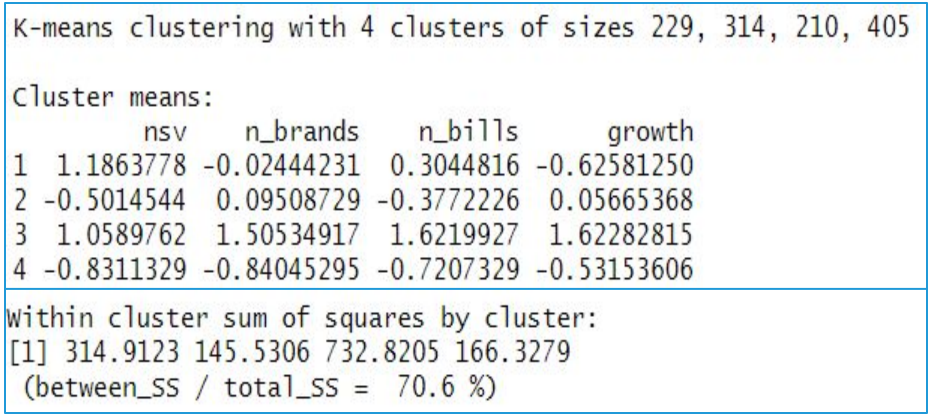
K-Means Method in R

```
# Importing Data
custsales<-read.csv("RETAILERS DATA.csv",header=T)
custsales_cl<-subset(custsales,select=c(-Custid,-region))
```

```
# Scale (standardize) all variables.(subtract mean and divide by
standard deviation)
custsales_cl<-scale(custsales_cl)
CL<-kmeans(custsales_cl,4)
CL
```

kmeans() perform k-means clustering on a data matrix cutsales_cl for 4 clusters.

# Output



```
K-means clustering with 4 clusters of sizes 229, 314, 210, 405

Cluster means:
         nsv    n_brands    n_bills      growth
1  1.1863778 -0.02444231  0.3044816 -0.62581250
2 -0.5014544  0.09508729 -0.3772226  0.05665368
3  1.0589762  1.50534917  1.6219927  1.62282815
4 -0.8311329 -0.84045295 -0.7207329 -0.53153606

Within cluster sum of squares by cluster:
[1] 314.9123 145.5306 732.8205 166.3279
 (between_SS / total_SS =  70.6 %)
```

Interpretation :

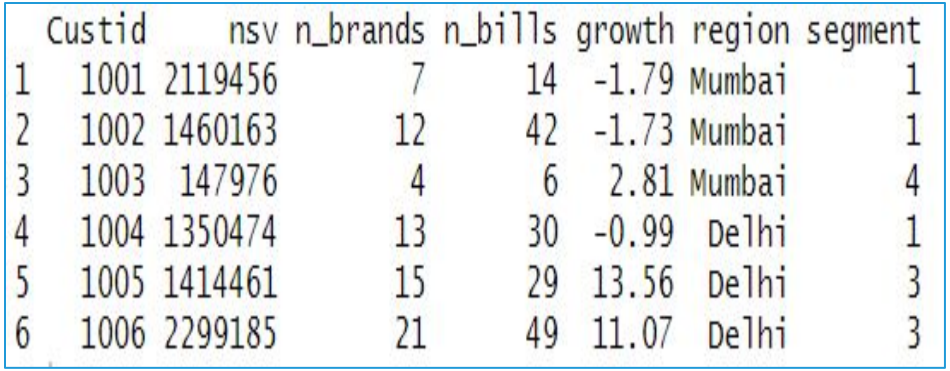
- Cluster 3 looks platinum customers group.

K-Means Method in R
Append Segment Variable

```
# Adding New column “segment” :
```

```
custsales$segment <- CL$cluster
head(custsales)
```

```
# Output
```



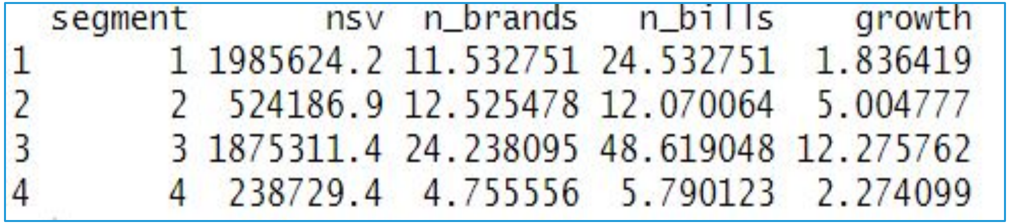
```
  Custid     nsv n_brands n_bills growth region segment
1   1001 2119456        7      14  -1.79 Mumbai       1
2   1002 1460163       12      42  -1.73 Mumbai       1
3   1003  147976        4       6   2.81 Mumbai       4
4   1004 1350474       13      30  -0.99  Delhi       1
5   1005 1414461       15      29  13.56  Delhi       3
6   1006 2299185       21      49  11.07  Delhi       3
```

K-Means Method in R : Summarize Clusters Using Original Variables

```
# Aggregating data based on segments
```

```
aggregate(cbind(nsv,n_brands,n_bills,growth)~segment,data=custsales,
FUN=mean)
```

```
# Output
```



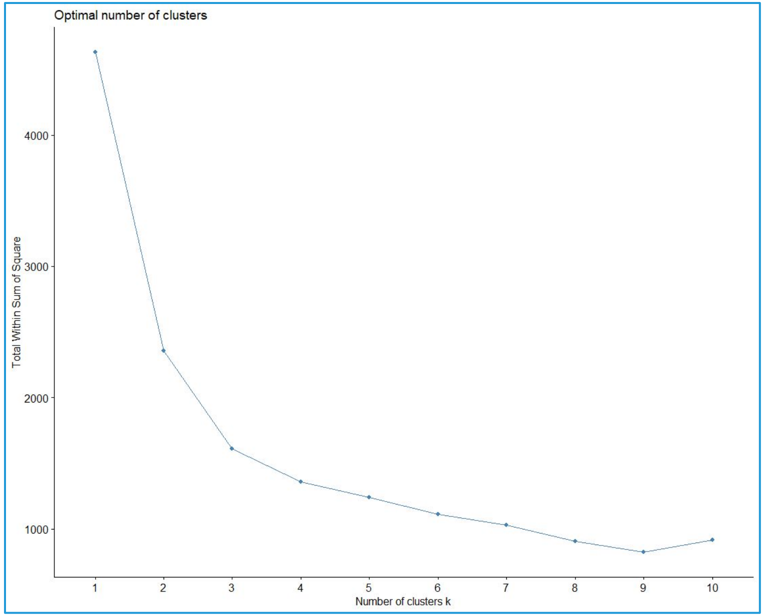
```
  segment       nsv  n_brands   n_bills    growth
1       1 1985624.2 11.532751 24.532751  1.836419
2       2  524186.9 12.525478 12.070064  5.004777
3       3 1875311.4 24.238095 48.619048 12.275762
4       4  238729.4  4.755556  5.790123  2.274099
```

Interpretation :

- Cluster 3 is group of ‘Platinum’ clusters.
- Cluster 4 is a group of ‘non-performers’

K-Means Method in R
Elbow Method

```
# Install & load package “factoextra”
install.packages("factoextra")
library(factoextra)
fviz_nbclust(custsales_cl, kmeans,method = "wss")
# Output
```



❑ **fviz_nbclust()** determines & visualize the optimal number of clusters using different methods, here we use within cluster sums of squares(wss)

Interpretation :

- The location of a bend in the plot is generally considered as an indicator of the appropriate number of clusters.
- Here K= 3 or 4 is a good solution.
- The method is termed as Elbow Method.

kmeansruns() in "fpc" Package
Finding Best K

- Package: fpc: Flexible Procedures for Clustering

- Performs K-means method for different values of ‘K’ and provides best value of K.

```
library(fpc)
CL1<-kmeansruns(custsales_cl,krange=2:10)
CL1$bestk
```

- ❑ **kmeansruns(data,krange)**
- ❑ **data** A numeric matrix of data, or an object that can be coerced to such a matrix
- ❑ **krange=** integer vector. Numbers of clusters which are to be compared

* Use this as only indicative

K-Means Clustering in R

kmeans() output includes :

cluster	A vector of integers (from 1:k) indicating the cluster to which each point is allocated.
centers	A matrix of cluster centres.
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.
iter	The number of (outer) iterations.
ifault	integer: indicator of a possible algorithm problem – for experts.

K Means Algorithms

- The algorithm of Hartigan and Wong (1979) is used by default.

- Note that some authors use k-means to refer to a specific algorithm rather than the general method: most commonly the algorithm given by

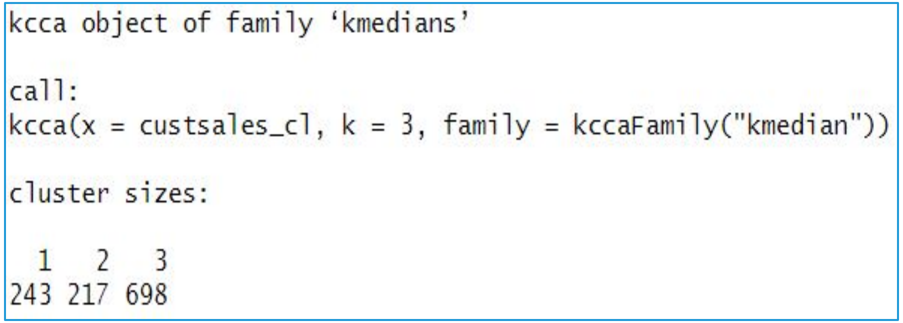
  MacQueen (1967) or sometimes that given by Lloyd (1957) and Forgy (1965).

K-Median Clustering in R

```
# Install and load package “flexclust”
# K-Median Clustering
install.packages("flexclust")
library(flexclust)

kmedian<-kcca(custsales_cl,3,family=kccaFamily("kmedian"))
kmedian
```

- ❑ **kcca()** performs k-centroid clustering on data matrix. The first two arguments are **data object** and **number of clusters** to be formed.
- ❑ **family=kccaFamily()** specifies object of class **kccaFamily**. Other options are **"kmeans"**, **"angle"**, **"jaccard"**, or **"ejaccard"**.

```
# Output
```



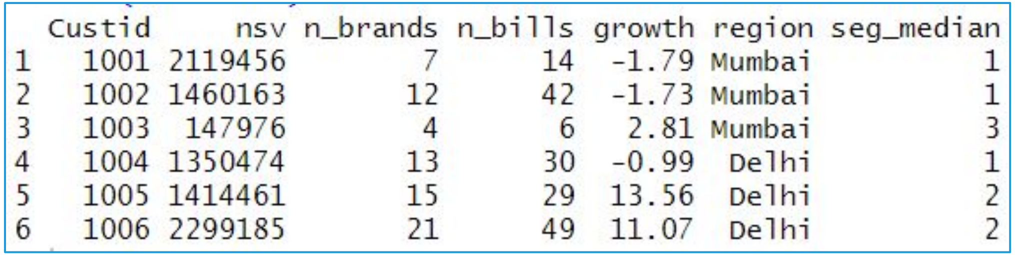
```
kcca object of family ‘kmedians’

call:
kcca(x = custsales_cl, k = 3, family = kccaFamily("kmedian"))

cluster sizes:

  1   2   3 
243 217 698 
```

K-Median Clustering in R

```
# Adding New column “segment” :
```

```
custsales$seg_median <- kmedian@cluster
head(custsales)
```

```
# Output
```



```
  Custid     nsv n_brands n_bills growth region seg_median
1   1001 2119456        7      14  -1.79 Mumbai          1
2   1002 1460163       12      42  -1.73 Mumbai          1
3   1003  147976        4       6   2.81 Mumbai          3
4   1004 1350474       13      30  -0.99  Delhi          1
5   1005 1414461       15      29  13.56  Delhi          2
6   1006 2299185       21      49  11.07  Delhi          2
```

Quick Recap

K-Means Clustering in R

- **`kmeans()`** function in base R performs K-Means Clustering
- **`kmeansruns()`** from package **`fpc`** can also be used for finding number of clusters.

K-Median Clustering in R

- **`kcca()`** function from package **`flexclust`** performs k-centroid clustering on data matrix.
- **`kccaFamily()`** specifies object of class **`kccaFamily`**.