R Workshop: Segmentation

Winter 2024

Contents

This handout is designed to help you practice the statistical analyses on Segmentation. In order to run these statistical methods, you need to install these R packages:

- mclust
- cluster

Cluster Analysis refers to a class of techniques used to classify individuals into groups such that:

- Individuals within a group should be as similar as possible
- Individuals belonging to different groups should be as dissimilar as possible

This handout shows three different cluster analysis techniques:

1. Hierarchical Clustering: *hclust()*
2. Mean-Based Clustering: *kmeans()*
3. Model-based Clustering: *mclust()*

1 The steps of clustering

Clustering analysis requires two stages: *finding a proposed cluster solution* and *evaluating that solution* for one's business needs. For each method, we go through the following steps:

- **Transform the data if needed for a particular clustering method.** For instance, some methods require all numeric data (e.g. *kmeans(), mclust()*)
- **Compute a distance matrix if needed.** Some methods require a pre-computed matrix of similarity in order to group observations (e.g. *hclust()*)
- **Apply the clustering method and save its result to an object.** For some methods, this requires specifying the number (K) of groups desired (e.g. *kmeans()).
- **Examine the solution in the model object about the underlying data and consider whether it answers a business question**. This is the most challenging part. It is up to you to figure out whether the solution tells a meaningful story for your data.

2 Segmentation data

Costco is a subscription-based retailer and offers individual membership for customers. They have collected data from N=300 potential or existing customers on some demographic variables, including:

- age (*age*)
- gender (*gender*)
- income (*income*)
- number of children (*kids*)
- whether they *own or rent* their homes (*ownHome*)
- whether they currently *subscribe* to the offered membership or not (*subscribe*).

The company is interested in the insights from this data on how to segment the market and target the promising groups.

```r
seg.df <- read.csv("Data_Segmentation.csv", stringsAsFactors = TRUE)
```

Check the data after loading:

```r
head(seg.df, n=8)
```

2.1 Recode factor into numeric data

We use *ifelse()* to recode the binary factors:

```
seg.df$gender <- ifelse(seg.df$gender=="Male",0,1)
seg.df$ownHome <- ifelse(seg.df$ownHome == "ownNo", 0,1)
seg.df$subscribe <- ifelse(seg.df$subscribe == "subNo", 0,1)

head(seg.df)
```

2.2 Rescaling the data

It is very important that all the variables are measured at a similar scale. A common procedure is to *center* each variable by subtracting its mean from every observation and *rescale* those centered values as units of standard deviation. This is commonly called *standardizing*, *normalizing*, or *Z-scoring* the data. We use *scale()* to rescale all variables at once.

```
seg.df.sc <- seg.df

seg.df.sc[, c(1,3,4)] <- scale(seg.df[, c(1,3,4)])
# We only need to standardize continuous variables.

head(seg.df.sc)
```

```
summary(seg.df.sc, digits = 2)
```

```
##       age             gender          income             kids           ownHome    
##  Min.   :-1.73   Min.   :0.00   Min.   :-2.787   Min.   :-0.90   Min.   :0.00  
##  1st Qu.:-0.64   1st Qu.:0.00   1st Qu.:-0.560   1st Qu.:-0.90   1st Qu.:0.00  
##  Median :-0.13   Median :1.00   Median : 0.054   Median :-0.19   Median :0.00  
##  Mean   : 0.00   Mean   :0.52   Mean   : 0.000   Mean   : 0.00   Mean   :0.47  
##  3rd Qu.: 0.53   3rd Qu.:1.00   3rd Qu.: 0.520   3rd Qu.: 0.52   3rd Qu.:1.00  
##  Max.   : 3.09   Max.   :1.00   Max.   : 3.145   Max.   : 4.07   Max.   :1.00  
##    subscribe   
##  Min.   :0.00  
##  1st Qu.:0.00  
##  Median :0.00  
##  Mean   :0.13  
##  3rd Qu.:0.00  
##  Max.   :1.00
```

3 Hierarchical clustering: *hclust()*

Hierarchical clustering is a popular method that groups observations according to their similarity. The *hclust()* method is one way to perform this analysis in R.

3.1 Distance

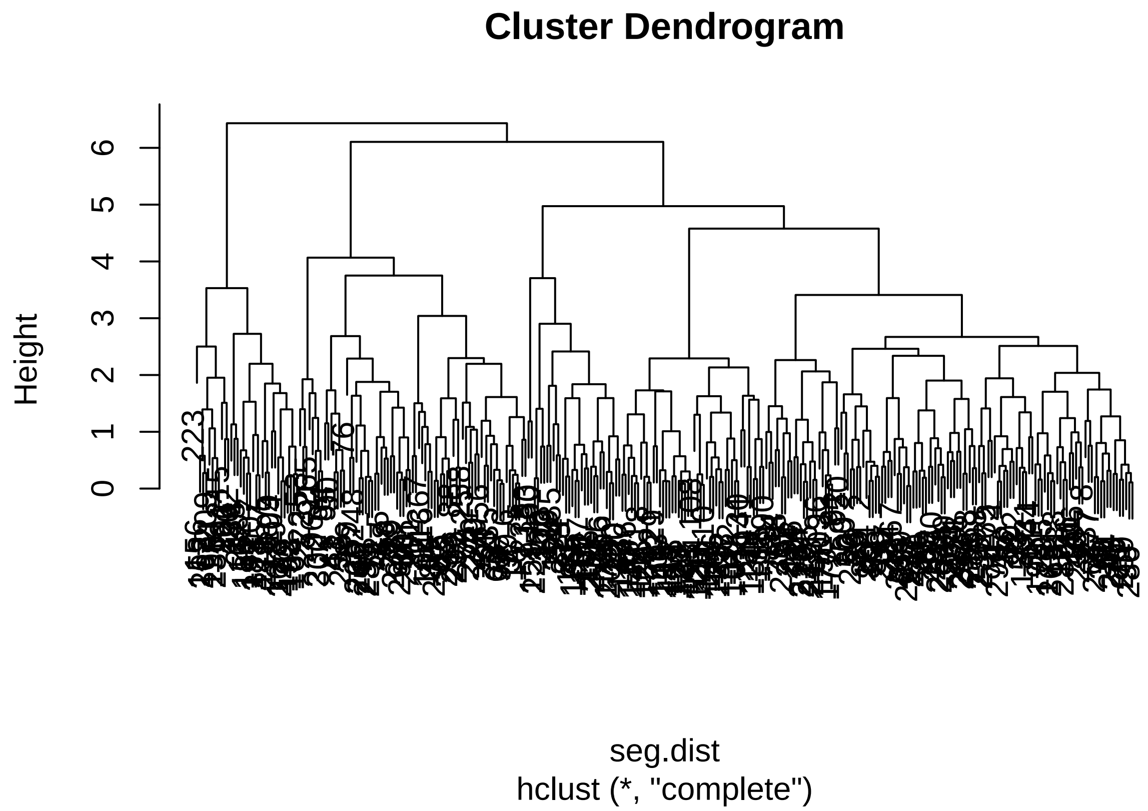
The primary information in hierarchical clustering is the *distance* between observations. The *Euclidean distance* is one of the best-known methods of calculating distance. This can be done with the *disc()* function.

```
seg.dist <- dist(seg.df.sc)
as.matrix(seg.dist)[1:5, 1:5]
```

```
##          1        2        3        4        5
## 1 0.000000 1.885319 1.786323 2.139937 2.225970
## 2 1.885319 0.000000 1.245679 2.705915 2.883670
## 3 1.786323 1.245679 0.000000 2.463979 2.936121
## 4 2.139937 2.705915 2.463979 0.000000 1.762212
## 5 2.225970 2.883670 2.936121 1.762212 0.000000
```

3.2 Clustering

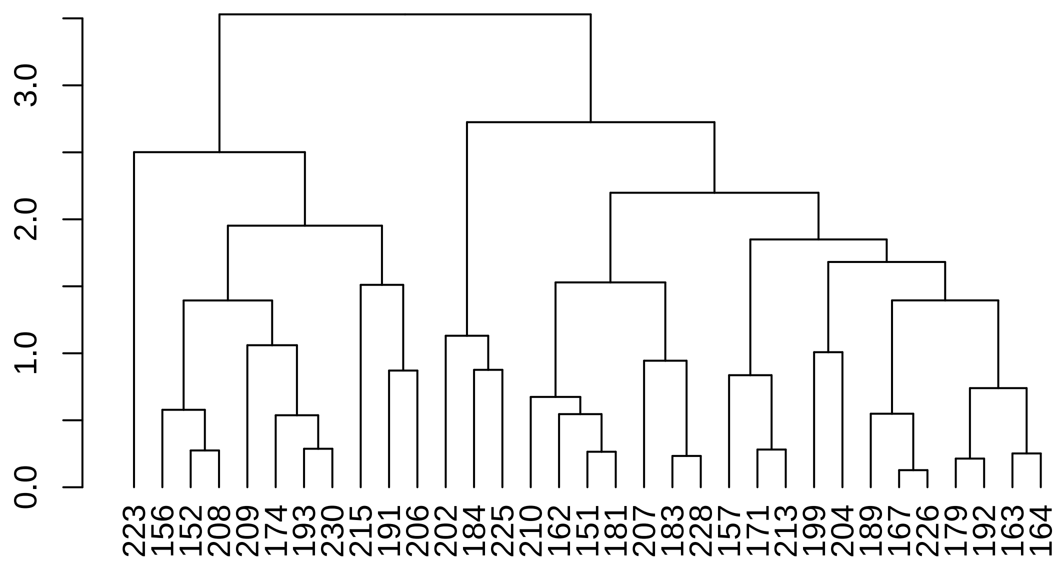
```
seg.hc <- hclust(seg.dist, method="complete")
plot(seg.hc)
```



A hierarchical dendrogram is interpreted primarily by height and where observations are joined.

It is sometimes helpful to zoom in on one section of the chart. We can cut it at a specific location and plot just one branch as follows. We coerce it to a dendrogram object, cut it at a certain height, and select the resulting branch that we want.

```
plot(cut(as.dendrogram(seg.hc), h = 4)$lower[[1]])
```



We can check the similarity of observations by selecting a few rows. Observations 101 and 107 are represented as being quite similar because they are linked at a very low height, as are observations 278 and 294. on the other hand, observations 173 and 141 are only joint at the highest level of this branch and thus should be relatively dissimilar. We can check those directly:

```
seg.df[c(156, 152),]   #similar
```

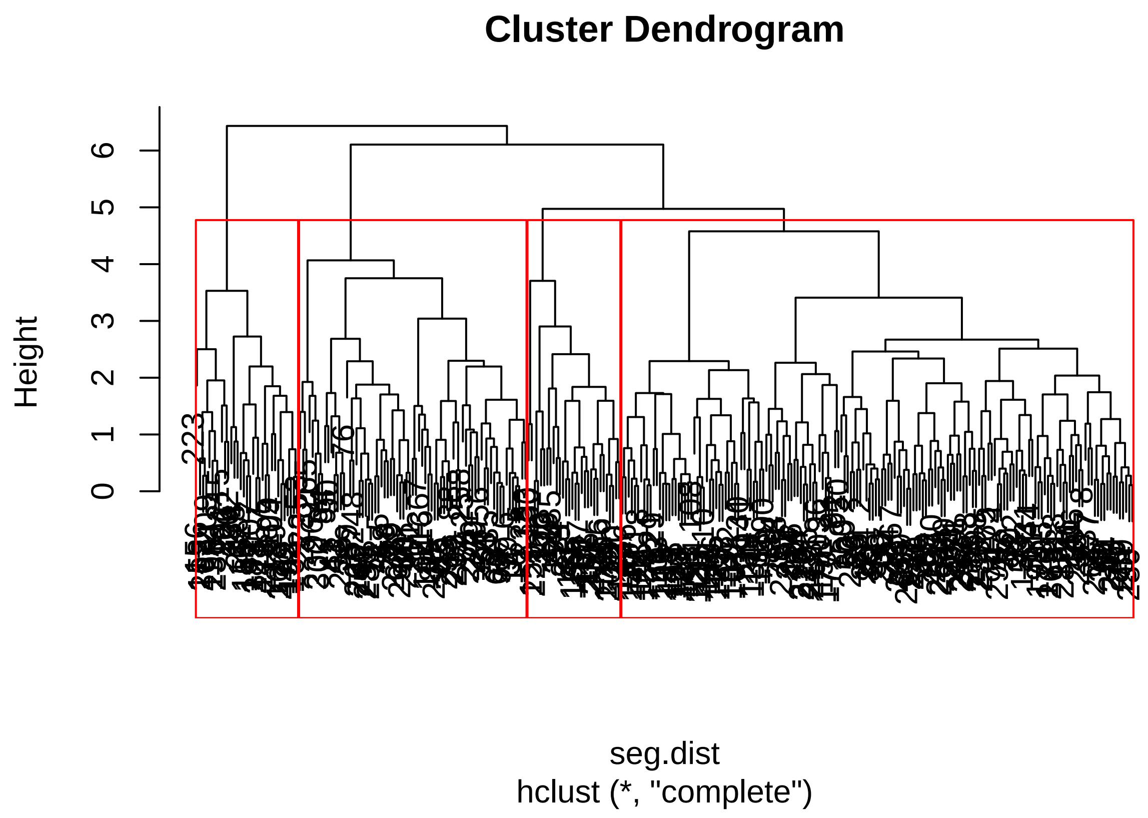
```
seg.df[c(156, 183),]   #less similar
```

As we can see, the first set is similar in all variables (age, gender, income, etc.). The second set differs substantially in some variables.

3.3 Getting groups

How do we get specific segment assignments? A dendrogram can be cut into clusters at any height desired, resulting in different numbers of groups. We must specify the number of groups desired. We can see where the dendrogram would be cut by overlaying its *plot()* with *rect.hclust()*, specifying the number of groups we want by using *(k=. . .)*:

```
plot(seg.hc)
rect.hclust(seg.hc, k=4, border = "red")
```



We obtain the assignment vector for observations using *cutree()*

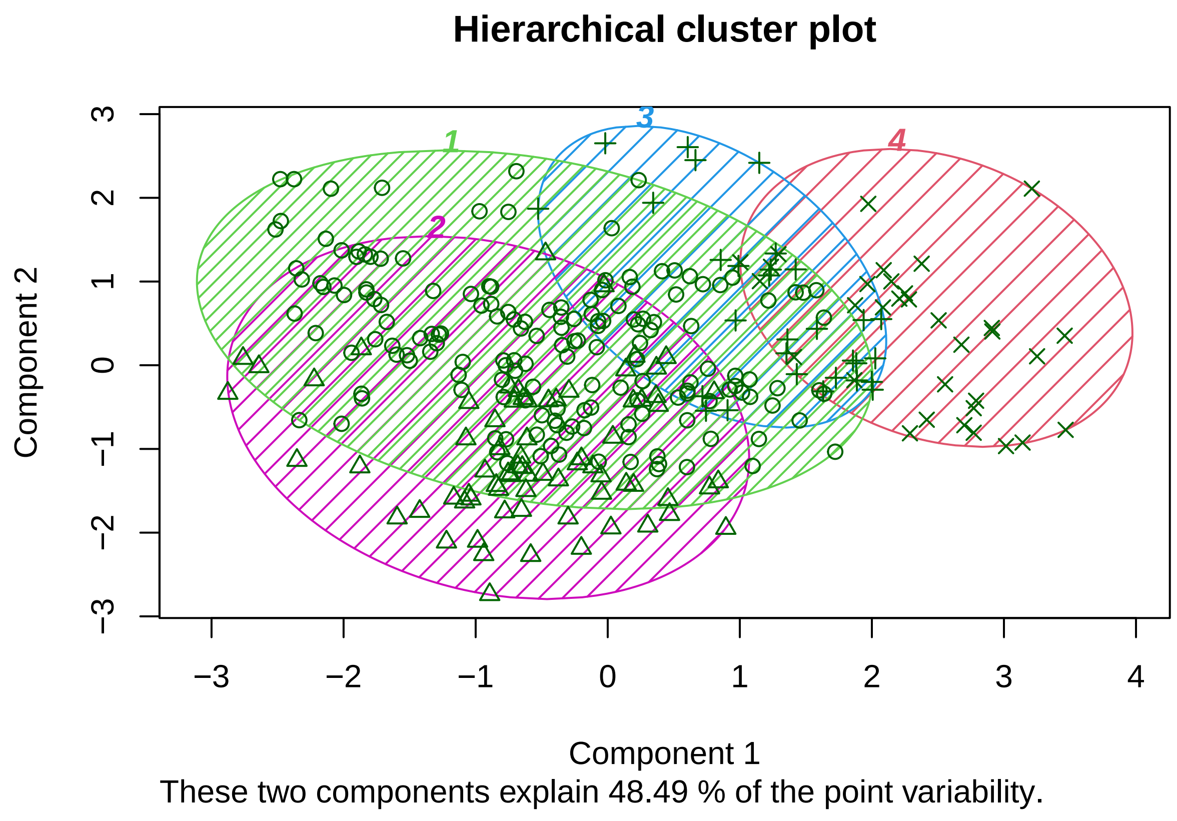
```
seg.hc.segment <- cutree(seg.hc, k=4) #membership vector for 4 groups
table(seg.hc.segment) #counts
```

```
## seg.hc.segment
##   1   2   3   4
## 164  73  30  33
```

3.4 Describing clusters

We can visualize the clusters by plotting them against a dimensional plot, by using *cluspot* from the *cluster* package.

```
library(cluster)
clusplot(seg.df, seg.hc.segment,
        color = TRUE, #color the groups
        shade = TRUE, #shade the ellipses for group membership
        labels = 4, #label only the groups, not the individual points
        lines = 0, #omit distance lines between groups
        main = "Hierarchical cluster plot", # figure title
)
```



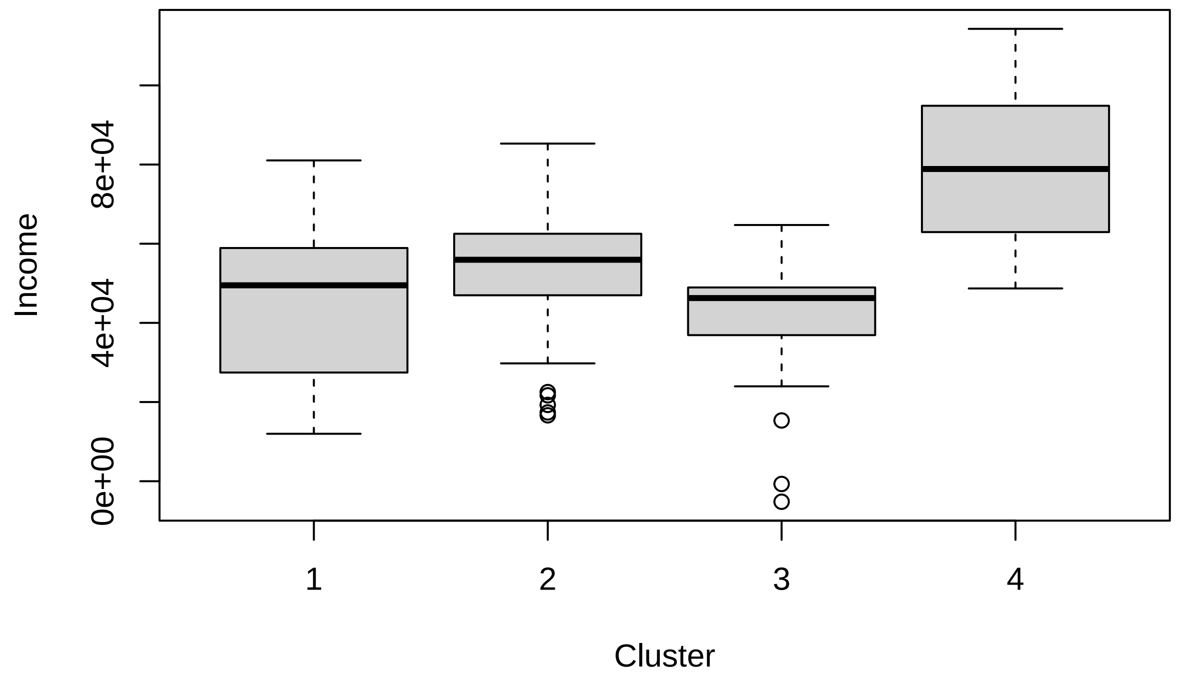
We can then describe the variables in seg.df with reference to the four clusters:

```r
aggregate(seg.df, list(seg.hc.segment), mean)
```

Try to interpret the groups. Gender? Homeowner? Is this interesting from a business point of view?

We can visually check the distribution of different variables according to segment using *boxplot()*:

```r
boxplot(seg.df$income ~ seg.hc.segment, ylab = "Income", xlab = "Cluster")
```



The result shows substantial differences in income by segment.

4 Mean-based clustering: *kmeans()*

4.1 Clustering

We now run *kmeans()*, which specifically requires specifying the number of clusters to find. We ask for four clusters with *centers=4*.

```
set.seed(96743)
seg.k <- kmeans(seg.df.sc, centers = 4)  #use standardized variables
```

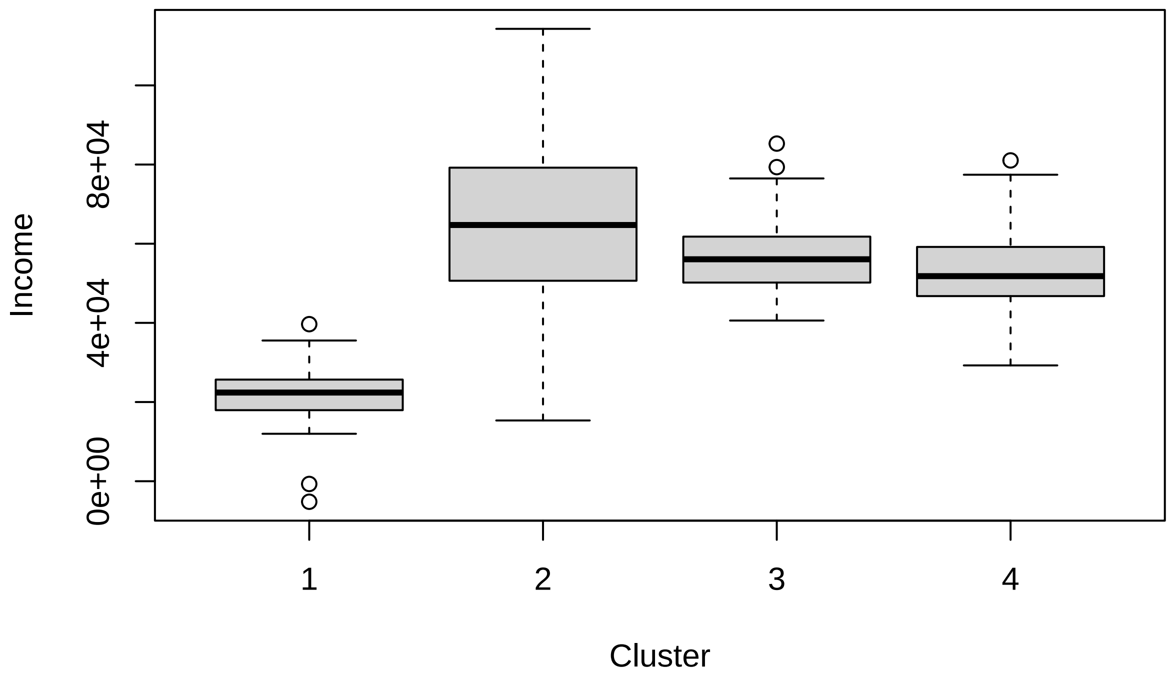
4.2 Describing clusters

We can do a quick check of the data by the proposed groups, where cluster assignments are found in the *$cluster* vector inside the *seg.k* model:

```
aggregate(seg.df, list(seg.k$cluster), mean)
```

We see some interesting differences; the groups appear to vary by age, gender, kids, income, and home ownership. For instance, we can visually check the distribution of income according to segment using *boxplot()*:

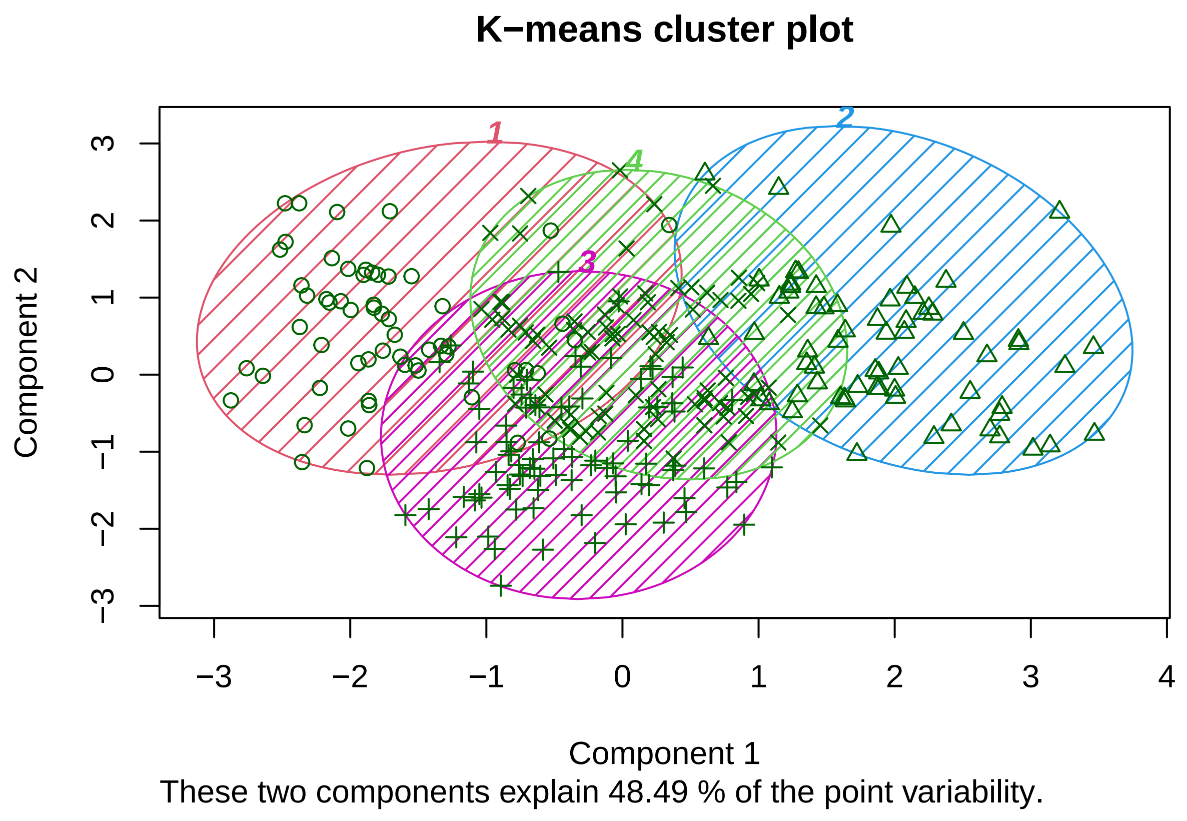
```
boxplot(seg.df$income ~ seg.k$cluster, ylab = "Income", xlab = "Cluster")
```



The result shows substantial differences in income by segment.

We visualize the clusters

```
clusplot(seg.df, seg.k$cluster,
        color = TRUE, #color the groups
        shade = TRUE, #shade the ellipses for group membership
        labels = 4, #label only the groups, not the individual points
        lines = 0, #omit distance lines between groups
        main = "K-means cluster plot", # figure title
)
```



This may suggest a business strategy. In the present case, for instance, we see that group 2 is modestly well-differentiated and has the highest average income. That may make it a good target for a potential campaign. Many other strategies are possible, too; the key point is that the analysis provides interesting options to consider.

A limitation of k-means analysis is that it requires specifying the number of clusters, and it can be difficult to determine whether one solution is better than another. If we were to use k-means for the present problem, we would repeat the analysis for k=3, 4, 5 and so forth and determine which solution gives the most useful result for our business goals.

5 Model-based clustering: *Mclust()*

The key idea here is that observations come from groups with different statistical distributions (such as different means and variances). The algorithms try to find the best set of such underlying distributions to explain the observed data.

5.1 Clustering

```r
library(mclust) #install if needed
```

```
## Package 'mclust' version 6.0.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
seg.mc <- Mclust(seg.df.sc)
summary(seg.mc)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VEV (ellipsoidal, equal shape) model with 3 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1291.139 300 73 -2998.655 -2998.655
##
## Clustering table:
##   1   2   3
## 163  71  66
```

The result shows that the data are estimated to have three clusters with the sizes shown in the table.

We also see log-likelihood information, which we can use to compare models. For instance, We try a 4-cluster solution.

```r
seg.mc4 <- Mclust(seg.df.sc, G =4) #specifying the number of clusters
summary(seg.mc4)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VII (spherical, varying volume) model with 4 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1690.304 300 31 -3557.425 -3597.143
##
## Clustering table:
##   1   2   3   4
##  51  74 119  56
```

Forcing it to find four clusters resulted in quite a different model, with a lower log-likelihood.

5.2 Comparing models

We can compare the 3-cluster and 4-cluster models using the Bayesian information criterion (BIC).

```r
BIC(seg.mc, seg.mc4)
```

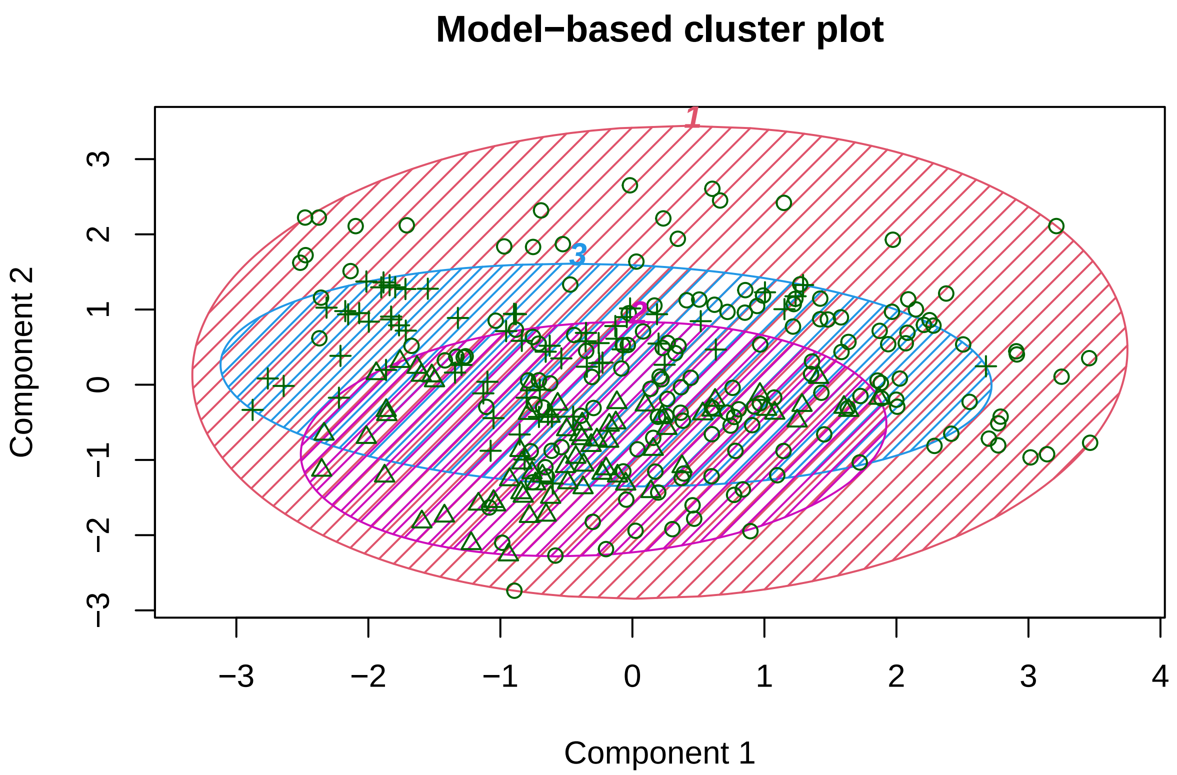
The low value of BIC indicates a better fit (e.g. 60 is better than 90; -1,000 is better than -990). In our case, we can see that the 3-cluster solution is better than the 4-cluster solution.

5.3 Describing clusters

Will the 3-cluster solution provide useful insight for the business? We check the quick summary and plot the clusters:

```r
aggregate(seg.df, list(seg.mc$classification), mean)
```

```r
library(cluster)
clusplot(seg.df, seg.mc$classification, color = TRUE, shade = TRUE,
         labels = 4, lines = 0, main = "Model-based cluster plot")
```



These two components explain 48.49 % of the point variability.

6 Recap

We have covered different methods to identify potential groups of observations in a data set.

- Different methods are likely to yield different solutions, and in general, there is no absolute "right" answer. It is recommended to try multiple clustering methods with different potential numbers of clusters.
- The results of segmentation are primarily about business value, and solutions should be evaluated in terms of both model fit and business utility. Although the model fit is an important criterion and should not be overlooked, it is ultimately necessary that an answer can be communicated to and used by stakeholders.