Chapter 7. Clustering

An unsupervised learning technique: more challenging, more subjective, often more difficult to perform evaluation of the results obtained: no cross-validation

Two techniques: Hierarchical clustering, and K-means clustering

Further readings:

James et al. (2013) Sections 10.3 & 10.5,

Provost and Fawcett (2013) Chapter 6.

Classification: finding groups of objects that differ with respect to some target characteristics of interest. So the groups are known in prior, defined as the target characteristics.

Clustering: finding groups of objects such that the objects within each group are similar, but the objects in different groups are not so similar. So the number of groups and the groups themselves are unknown in prior.

- Do our customers naturally fall into different groups, so we can develop better products, better marketing campaigns, better sales methods, or better customer service by understanding the natural subgroups?

- A search engine might choose what search results to display to a particular individual based on the click histories of other individuals with similar search patterns.

- Clustering single-malt-scotch whiskeys: natural groupings by taste – run a small shop as ‘a place to go for single-malt scotch‘ in a well-to-do neighbourhood

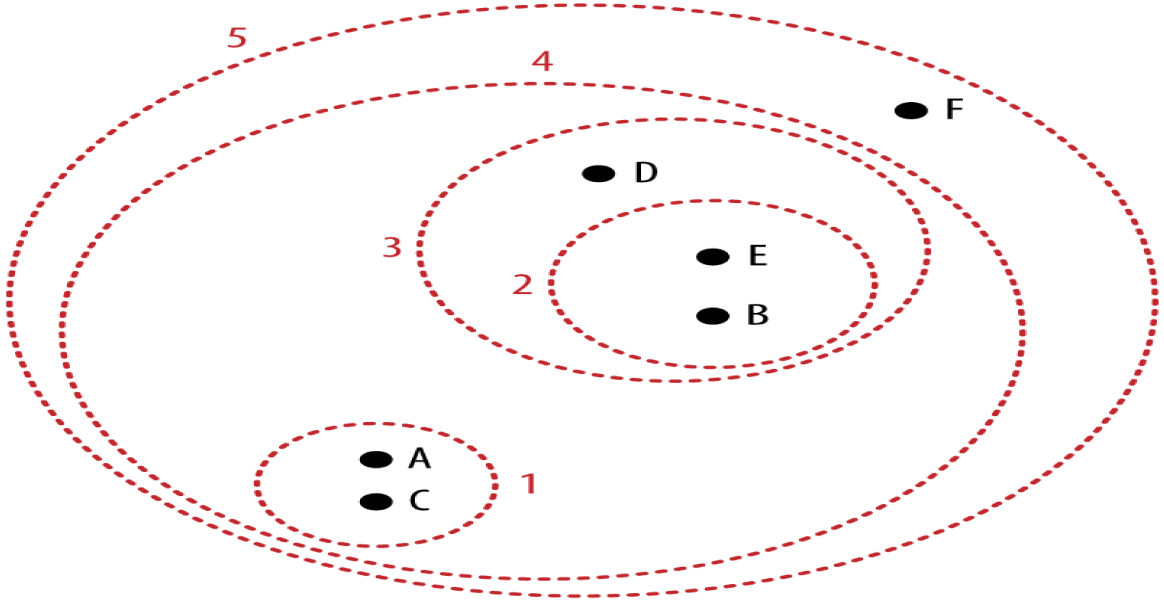
Hierarchical Clustering: clustering by similarity, dendrogram

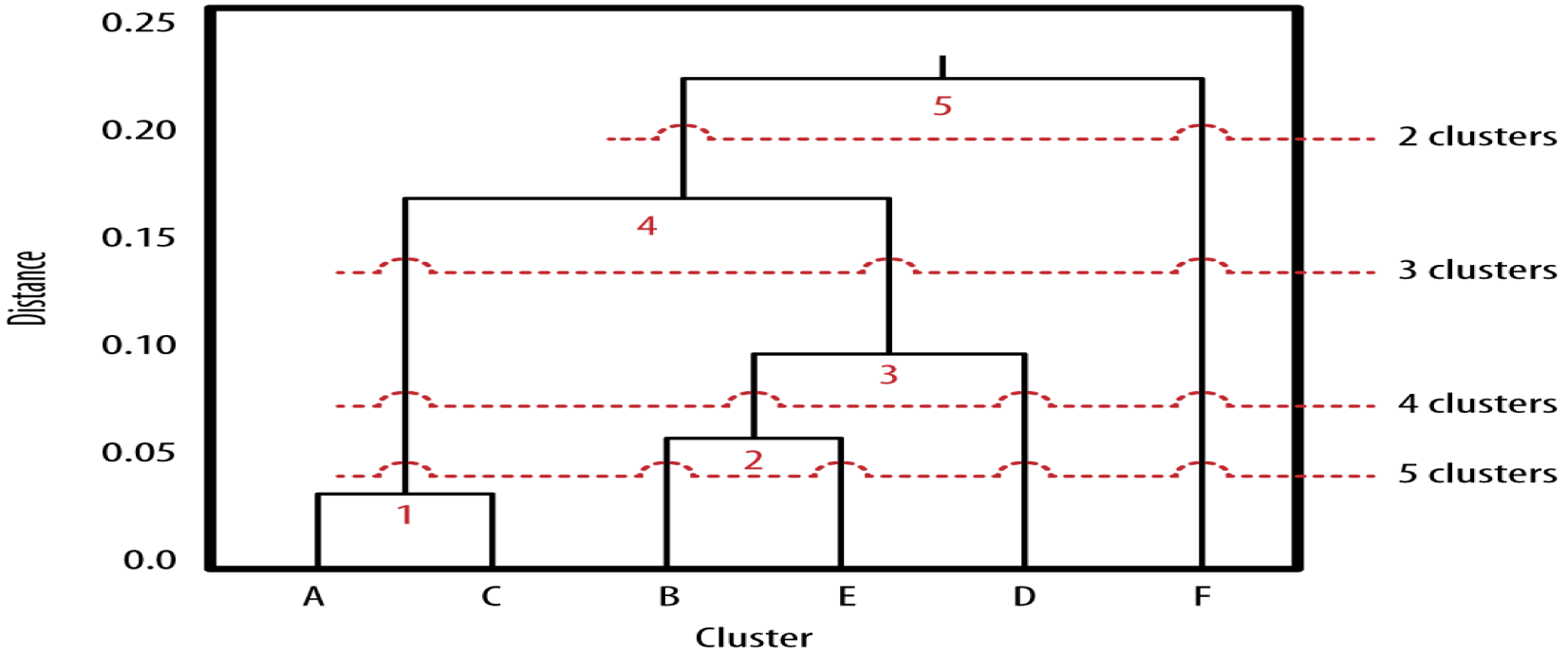
1. Calculate an appropriate distance for any two objects. At distance 0, put each object into a separate group.

2. Put two groups with the smallest distance together. Update the distances between any two groups using one of the four linkages:

- *Complete*: the maximum pairwise distance between the objects from two groups
- *Single*: the minimum pairwise distance between the objects from two groups
- *Average*: the average pairwise distance between the objects from two groups
- *Centroid*: the distance between the centroids (i.e. the mean vectors) of the two groups.

3. Repeat Step 2 above until all the objects are in one group.


5
4
3
2
1
F
D
E
B
A
C


0.25
0.20
0.15
0.10
0.05
0.0
Distance
5
4
3
2
1
2 clusters
3 clusters
4 clusters
5 clusters
A
C
B
E
D
F
Cluster

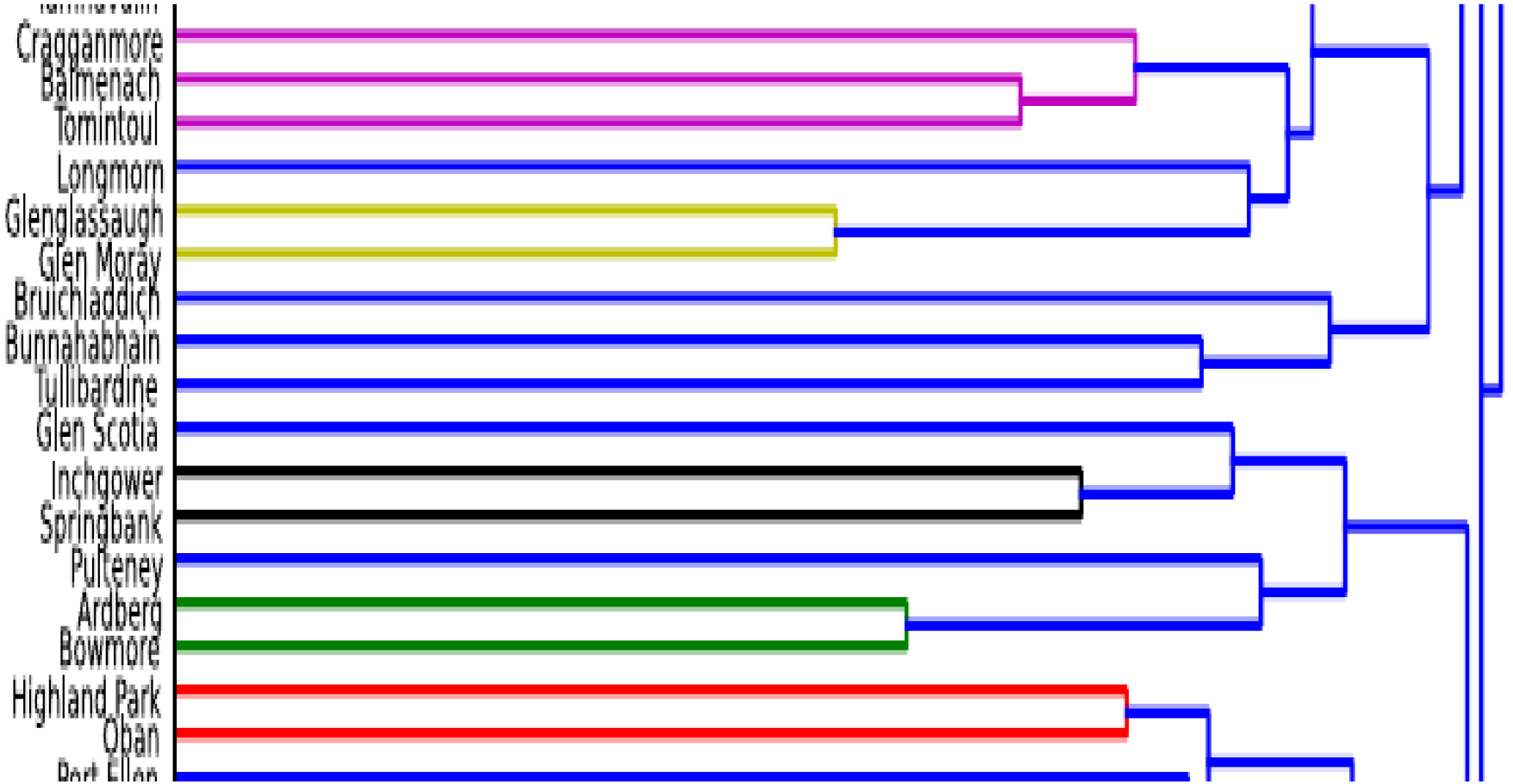
Dendrogram: a landscape of data similarity

Any number of clusters can be obtained by cutting dendrogram at an appropriate height (i.e. distance)

In practice, the number of clusters is often chosen by looking at dendrogram.

For the example on previous page, 3 clusters could be selected as there is a relatively long distance between Group 3 (0.10) and Group 4 (0.17).

An Excerpt of hierarchical clustering of single malt Scotch whiskeys.


Cragganmore
Balmenach
Tomintoul
Longmorn
Glenglassaugh
Glen Moray
Bruichladdich
Bunnahabhain
Tullibardine
Glen Scotia
Inchgower
Springbank
Pulteney
Ardberg
Bowmore
Highland Park
Oban

Choice of Dissimilarity Measure and Linkage Function: very important, strong effect on the resulting dendrogram

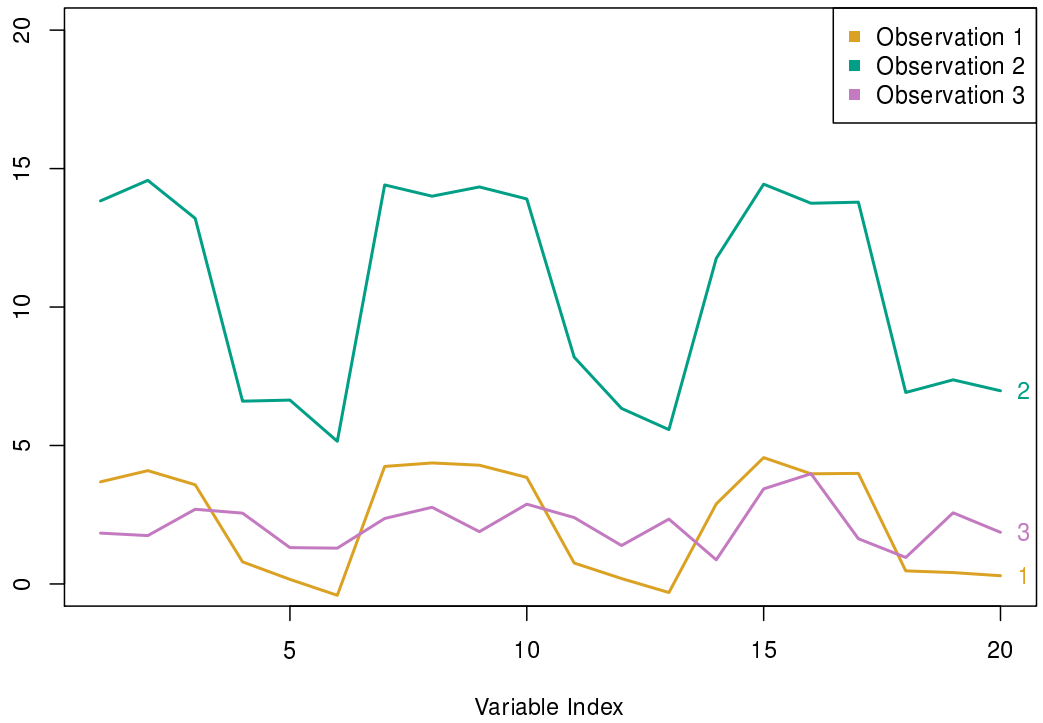
Type of data, business question in hand should be taken into account.

Clustering online shoppers: Data form a matrix with shoppers being rows and items available for purchase being columns, the elements of the matrix indicate the number of the times a given shopper has purchased a given item

Using Euclidean distance clusters together the shoppers who have bought very few items, which is not desirable.

Correlation-based distance clusters together shoppers with similar preferences (e.g. bought A and B but not C and D etc).

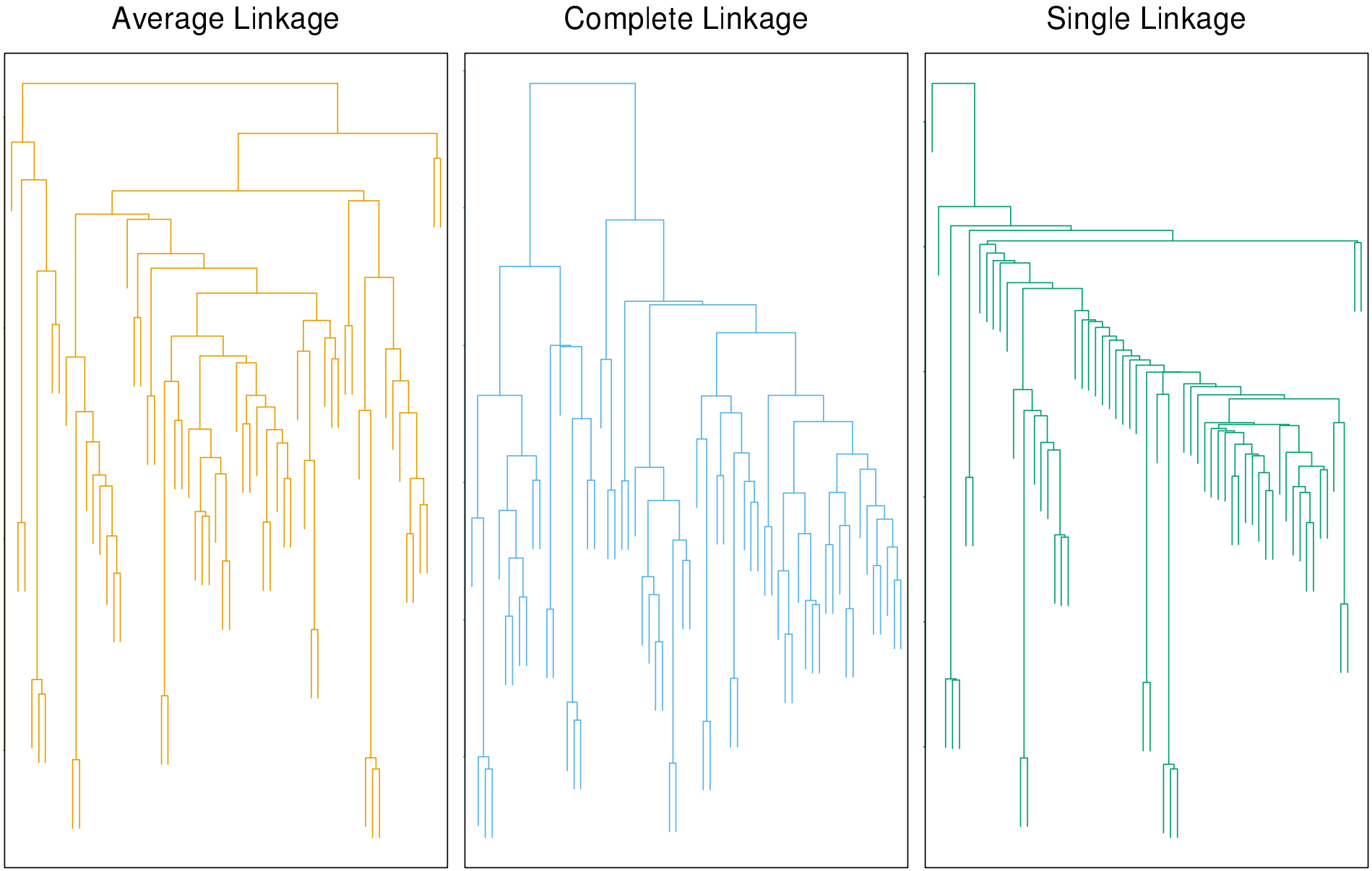
Complete linkage would make the shoppers in the same cluster more homogeneous, such that same ads can be shown to each clusters.



Each observation is measured on 20 attributes.

Euclidean distance between Observations 1 and 3 is small.

Correlation between Observations 1 and 2 is large, so the correlation based distance is small.


Average Linkage
Complete Linkage
Single Linkage

K-means clustering: partition the whole objects into K distinct, non-overlapping clusters.

An nearest neighbour approach: each object is represented by a p-feature vector.

K needs to be pre-specified.

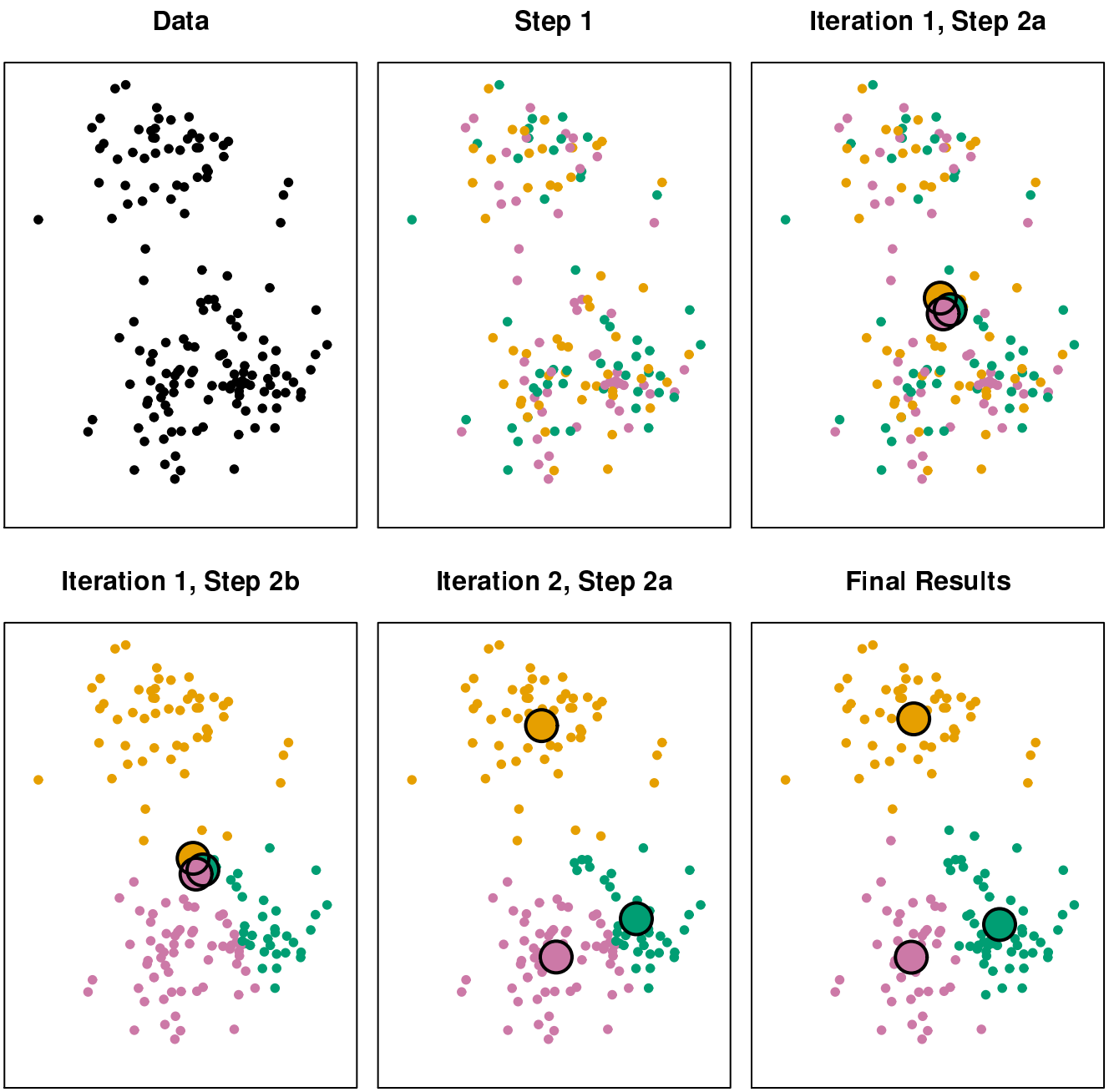
1. Randomly assign all the objects into one of K clusters

2. Iterate until the cluster assignments stop changing:

   (a) Calculate the centroid (i.e. a p-feature mean vector) for each of the K clusters

   (b) Assign each object to the cluster whose centroid is closest, measured using, e.g. Euclidean distance.

Remark. Step 1 assigns an initial value. When p is large, using a good initial value is important.



K-means clustering with $K = 3$ for randomly generated 150 points.

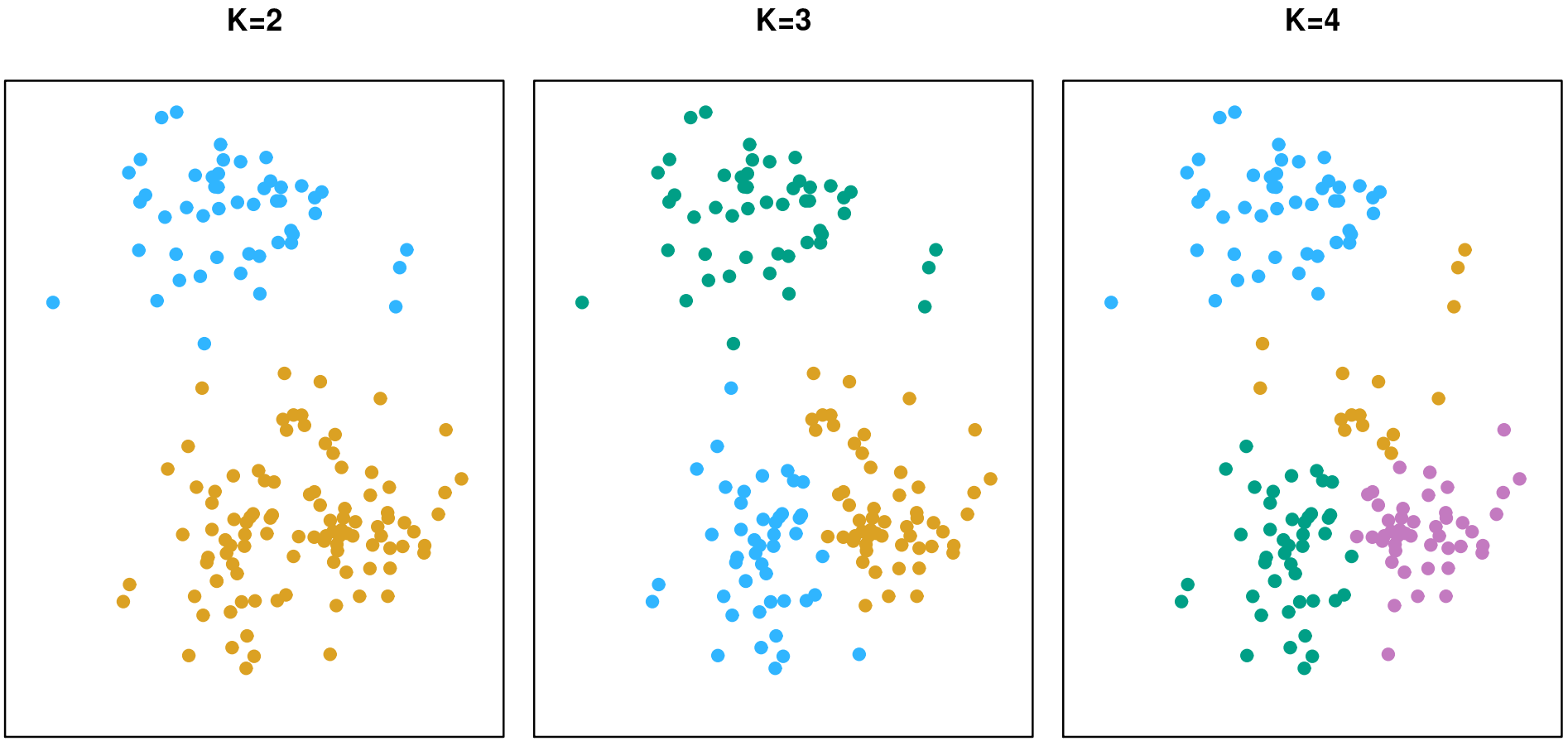
Step 1. Randomly assign 150 points to 3 groups

Step 2(a) Calculate centroids for each of 3 groups

Step 2(b) Assign each point to the cluster with the closest centroid

Performing 6 times the K-means clustering with $K = 3$.

3 different local optima were obtained


K=2
K=3
K=4

Example: Clustering Business News Stories

Objective: identify different groupings of news stories released on a particular company.

Purposes: gain a quick understanding of the news on a company without having to read every news story; categorize forthcoming news stories for a news prioritization process; or simply to understand data better before undertaking deeper learning (such as to relate business news stories to stock performance).

Data: Thomson Reuters Text Research Collection (TRC2)

`http://trec/nist/gov/data/reuters/reuters.html`

1,800,370 news stories from January 2008 to February 2009 (14 months)

312 news stories whose headlines specifically mentioned company Apple or its stock symbol APPL from the above corpus

Data Preparation:

remove HTML, URL and other stop words

text case-normalized

eliminate words which occurred rarely (fewer than 2 documents)

eliminate words which occurred too commonly (more than 50% documents)

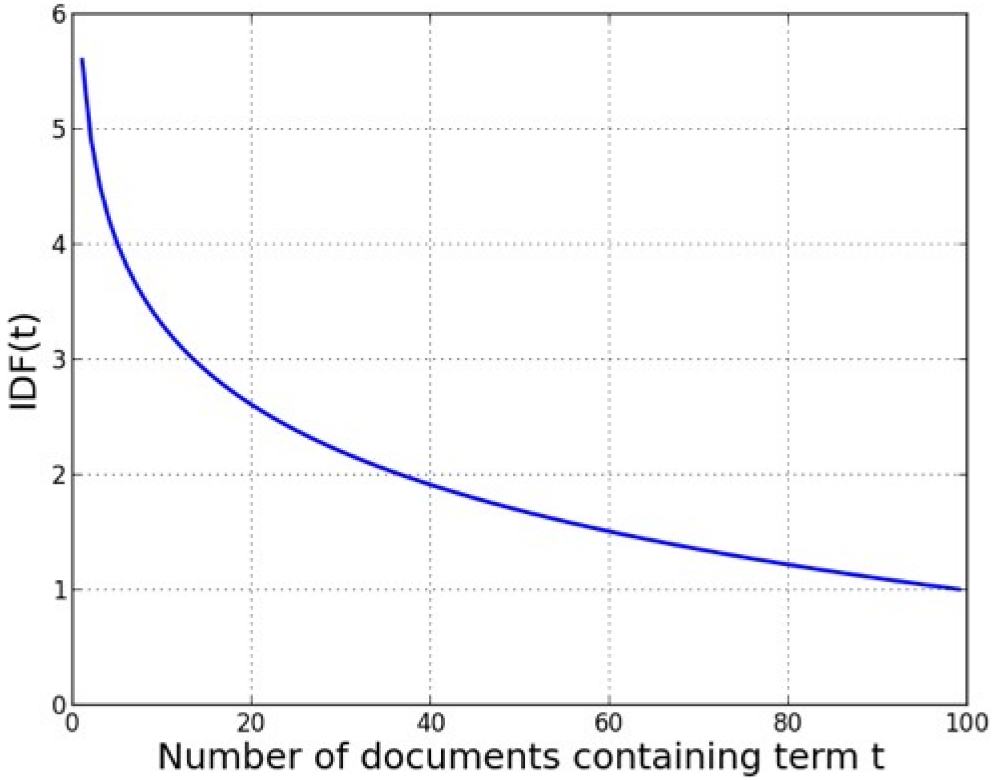
The remaining words form the *vocabulary*.

Each document is represented by a long numerical vector consisting of "TFIDF score" for each word in the vocabulary.

$$\text{TFIDF} = \text{TF (term frequency)} \times \text{IDF (inverse document frequency)}$$

$$\text{TF(t, d)} = \text{No. of times of word t occurring in document d}$$

$$\text{IDF(t)} = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing word t}}\right)$$



IDF of a term t within a corpus of 100 documents

9 clusters for the Apple news stories, derived by K-means clustering method.

Below we present a description of the 9 clusters, along with some headlines of the stories contained in each clusters. Note that entire news story was used in the clustering, not just the headline.

Cluster 1. These stories are analyst's announcements concerning ratings changes and price target adjustments:

```
* RBC RAISES APPLE <AAPL.O> PRICE TARGET TO $200 FROM $190; KEEPS OUT
PERFORM RATING
* THINKPANMURE ASSUMES APPLE <AAPL.O> WITH BUY RATING; $225 PRICE TARGET
* AMERICAN TECHNOLOGY RAISES APPLE <AAPL.O> TO BUY FROM NEUTRAL
* CARIS RAISES APPLE <AAPL.O> PRICE TARGET TO $200 FROM $170; RATING
ABOVE AVERAGE
* CARIS CUTS APPLE <AAPL.O> PRICE TARGET TO $155 FROM $165; KEEPS ABOVE
```

AVERAGE RATING

Cluster 2. This cluster contains stories about Apple's stock price movements, during and after each day of trading:

* Apple shares pare losses, still down 5 pct
* Apple rises 5 pct following strong results
* Apple shares rise on optimism over iPhone demand
* Apple shares decline ahead of Tuesday event
* Apple shares surge, investors like valuation

Cluster 3. In 2008, there were many stories about Steve Jobs, Apple's charismatic CEO, and his struggle with pancreatic cancer. Jobs' declining health was a topic of frequent discussion, and many business stories speculated on how well Apple would continue without him. Such stories clustered here:

* ANALYSIS-Apple success linked to more than just Steve Jobs

* NEWSMAKER-Jobs used bravado, charisma as public face of Apple
* COLUMN-What Apple loses without Steve: Eric Auchard
* Apple could face lawsuits over Jobs' health
* INSTANT VIEW 1-Apple CEO Jobs to take medical leave
* ANALYSIS-Investors fear Jobs-less Apple

Cluster 4. This cluster contains various Apple announcements and releases. Superficially, these stories were similar, though the specific topics varied:

* Apple introduces iPhone "push" e-mail software
* Apple CFO sees 2nd-qtr margin of about 32 pct
* Apple says confident in 2008 iPhone sales goal
* Apple CFO expects flat gross margin in 3rd-quarter
* Apple to talk iPhone software plans on March 6

Cluster 5. This cluster's stories were about the iPhone and deals to sell iPhones in other countries:

* MegaFon says to sell Apple iPhone in Russia
* Thai True Move in deal with Apple to sell 3G iPhone
* Russian retailers to start Apple iPhone sales Oct 3
* Thai AIS in talks with Apple on iPhone launch
* Softbank says to sell Apple's iPhone in Japan

Cluster 6. One class of stories reports on stock price movements outside of normal trading hours (known as Before and After the Bell):

* Before the Bell-Apple inches up on broker action
* Before the Bell-Apple shares up 1.6 pct before the bell
* BEFORE THE BELL-Apple slides on broker downgrades
* After the Bell-Apple shares slip
* After the Bell-Apple shares extend decline

Centroid 7. This cluster contained little thematic consistency:

* ANALYSIS-Less cheer as Apple confronts an uncertain 2009

* TAKE A LOOK - Apple Macworld Convention
* TAKE A LOOK-Apple Macworld Convention
* Apple eyed for slim laptop, online film rentals
* Apple's Jobs finishes speech announcing movie plan

Cluster 8. Stories on iTunes and Apple's position in digital music sales formed this cluster:

* PluggedIn-Nokia enters digital music battle with Apple
* Apple's iTunes grows to No. 2 U.S. music retailer
* Apple may be chilling iTunes competition
* Nokia to take on Apple in music, touch-screen phones
* Apple talking to labels about unlimited music

Cluster 9. A particular kind of Reuters news story is a News Brief, which is usually just a few itemized lines of very terse text (e.g. 'Says

purchase new movies on iTunes same day as dvd release'). The contents of these New Briefs varied, but because of their very similar form they clustered together:

```
* BRIEF-Apple releases Safari 3.1
* BRIEF-Apple introduces ilife 2009
* BRIEF-Apple announces iPhone 2.0 software beta
* BRIEF-Apple to offer movies on iTunes same day as DVD release
* BRIEF-Apple says sold one million iPhone 3G's in first weekend
```

Some of these clusters are interesting and thematically consistent while others are not. Some are just collections of superficially similar text.

Syntactic similarity is not semantic similarity.

We should not expect every cluster to be meaningful and interesting. Nevertheless, clustering is often a useful tool to uncover structure in our data that we did not foresee. Clusters can suggest new and interesting data mining opportunities.

Small Decision with Big Consequences

- Should the observations or features first be standardized in some way? For example, make the mean of each feature 0, the STD 1.

- For hierarchical clustering,

  What distance measure should be used?
  What linkage should be used?
  Where should one cut the dendrogram in order to obtain clusters?

- For K-means clustering,

  How many clusters should we set for?
  What distance measure should be used?

In practice, try different choices, and look for the one with most useful or interpretable solution.

There is no single right answer for Clustering. Any solution which exposes some interesting aspects of the data should be considered.

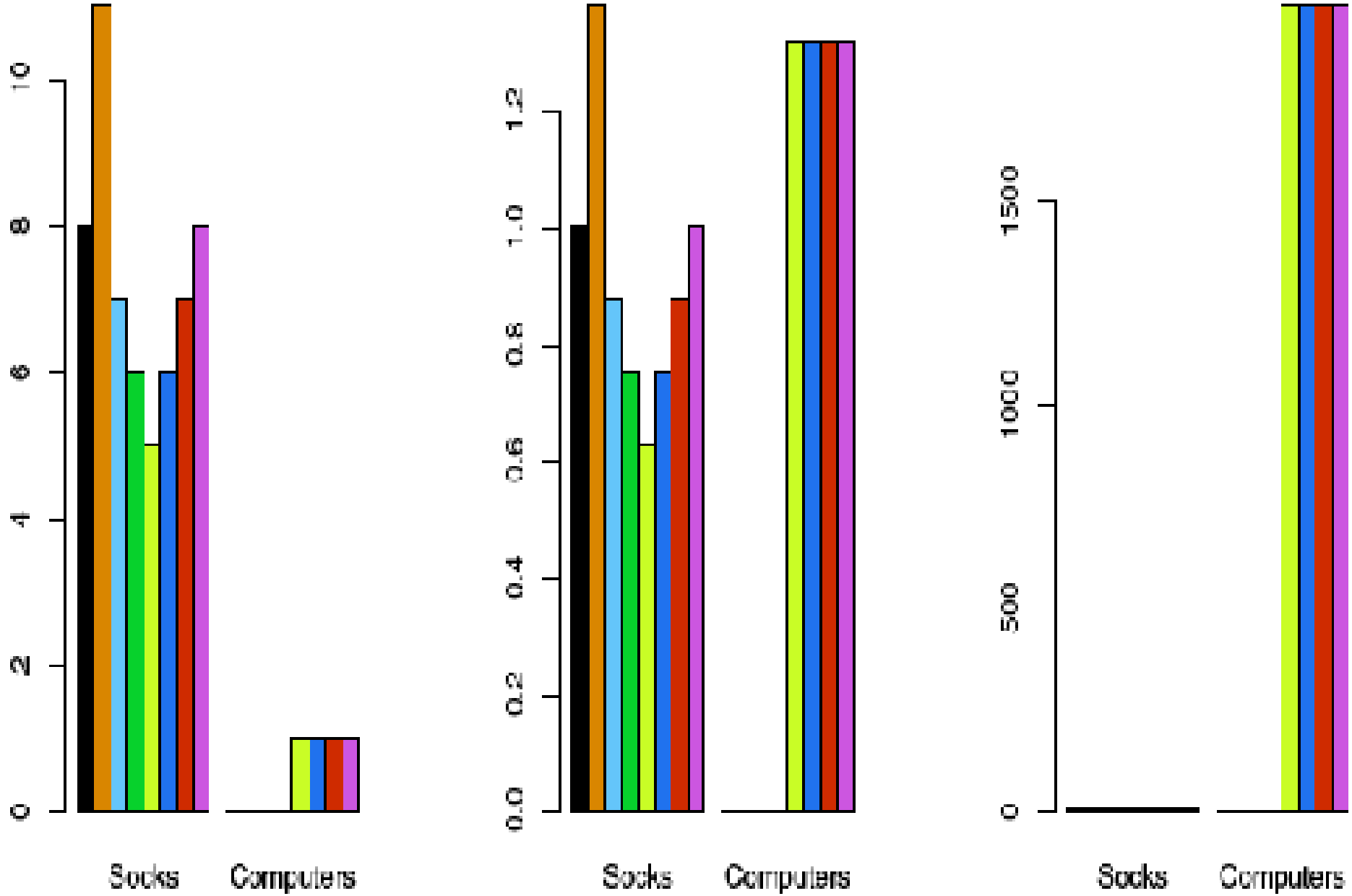
An eclectic online retailer sells two items: socks and computers

Left the number of pairs of socks, and computers, purchased by eight online shoppers is displayed. Each shopper is shown in a different color. If inter-observation dissimilarities are computed using Euclidean distance on the raw variables, then the number of socks purchased by an individual will drive the dissimilarities obtained, and the number of computers purchased will have little effect. This might be undesirable, since (1) computers are more expensive than socks and so the online retailer may be more interested in encouraging shoppers to buy computers than socks, and (2) a large difference in the number of socks purchased by two shoppers may be less informative about the shoppers′ overall shopping preferences than a small difference in the number of computers purchased.

Center the same data is shown, after scaling each variable by its standard deviation. Now the number of computers purchased will

have a much greater effect on the inter-observation dissimilarities obtained.

Right the same data are displayed, but now the y-axis represents the number of dollars spent by each online shopper on socks and on computers. Since computers are much more expensive than socks, now computer purchase history will drive the inter-observation dissimilarities obtained.


0
2
4
6
8
10
Socks
Computers
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Socks
Computers
0
500
1000
1500
Socks
Computers

Understanding the Results of Clustering

Clustering is often used as a data-exploratory analysis.

Understanding and Interpretation of the Results: domain knowledge, intuition and creativity

For the whiskey example, 12 clusters are resulted by cutting dendrogram at a certain height, here are two of them:

- Group A: `Aberfeldy, Glenugie, Laphroaig, Scapa`
- Group H: `Bruichladdich, Deanston, Fettercairn, Glenfiddich, Glen Mhor, Glen Spey, Glentauchers, Ladyburn, Tobermory`

We can simply look into the names of whiskeys in each clusters, which may make sense for whiskey-lovers/experts.

However such an approach does not make sense for, e.g., the customer clusters of a large retailer.

One can go further:

- Group A
  * Scotches: Aberfeldy, Glenugie, Laphroaig, Scapa
  * The best of its class: Laphroaig (Islay), 10 years, 86 points
  * Average characteristics: full gold; fruity, salty; medium; oily, salty, sherry; dry
- Group H
  * Scotches: Bruichladdich, Deanston, Fettercairn, Glenfiddich, Glen Mhor, Glen Spey, Glentauchers, Ladyburn, Tobermory
  * The best of its class: Bruichladdich (Islay), 10 years, 76 points
  * Average characteristics: white wyne, pale; sweet; smooth, light; sweet, dry, fruity, smoky; dry, light

Now those information is useful for whiskey retailers/shops and not-experts.

Two additional pieces of info:

Best whiskey in the class: from Jackson (1989) – unused in clustering

Average characteristics: extract from the centroid of each cluster (i.e. those with averages close to 1).

Using Supervised Learning to Generate Cluster Descriptions

A cluster centroid, in effect, reflects the average characteristics of the members in the cluster.

Those characteristics may be descriptive, represent the commonalities of the cluster members. But they do not tell how the clusters differ.

Basic Idea: Introduce a label (responsive) variable; each individual is given a label according to the cluster it belongs to. We then derive, for example, a decision tree, for 'classifying the clusters'.

Two approaches: classify k-classes (i.e. one class per cluster), or k separate classification problems: each classify one cluster from the others.

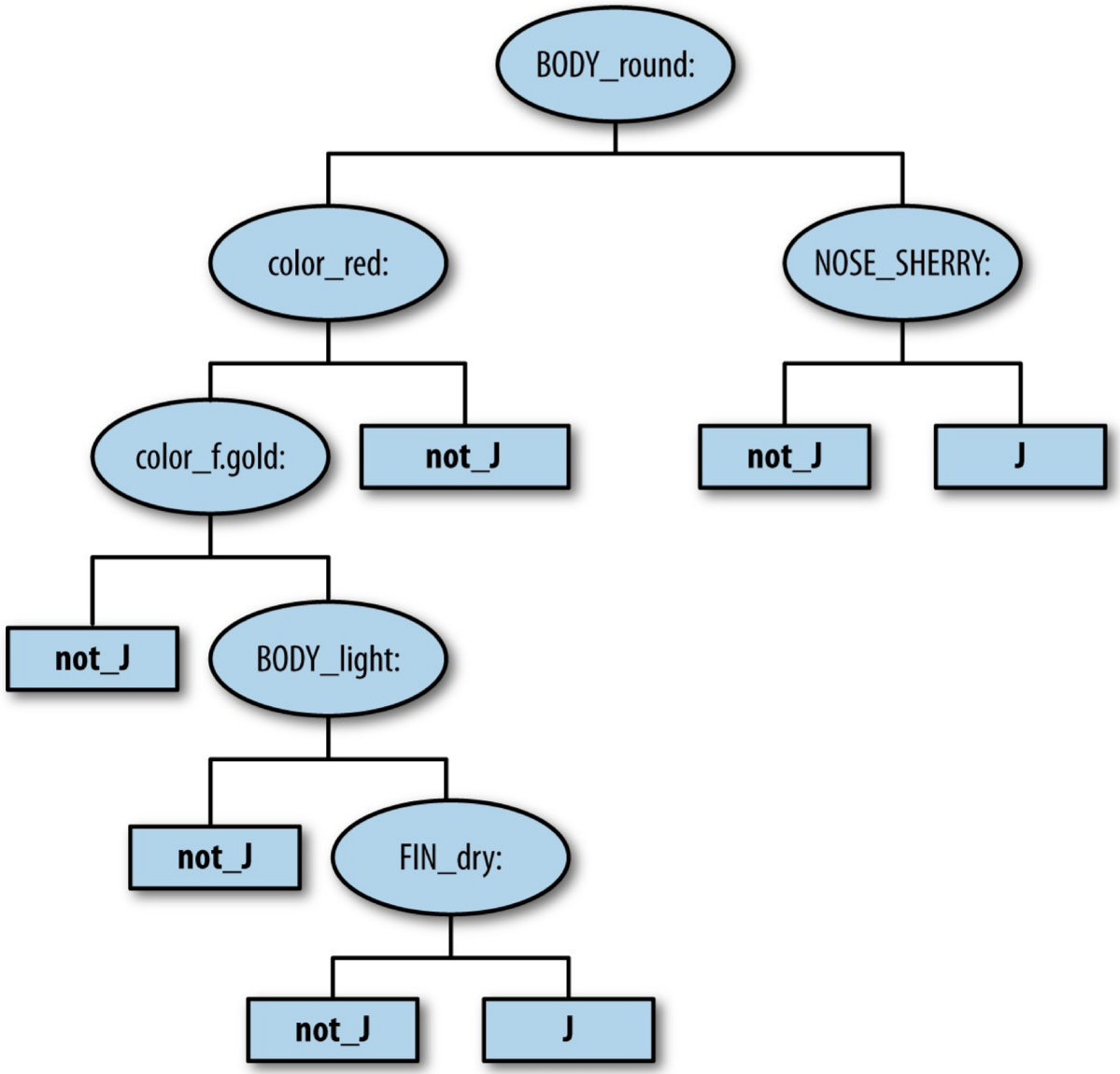
We adopt the 2nd approach to whiskey example: to classify cluster J from the others:

- Group J

  * Scotches: Glen Albyn, Glengoyne, Glen Grant, Glenlossie, Linkwood, North Port, Saint Magdalene, Tamdhu
  * The best of its class: Linkwood (Speyside), 12 years, 83
  * Average characteristics: full gold; dry, peaty, sherry; light to medium, round; sweet; dry

An excerpt of the dataset looks like this:

```
0,0,0,...,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,J % Glen Grant
0,0,0,...,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,not_J % Glen Keith
0,0,0,...,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,not_J % Glen Mhor
```



A decision tree for Cluster J on Scotches data.

'`No`' for all branches on the left,

'`Yes`' for all branches on the right

Only two leaves labelled J

Two leaves labelled J:

```
1. (ROUND_BODY = 1) AND (NOSE_SHERRY = 1)

2. (ROUND_BODY = 0) AND (color_red = 0) AND (color_f.gold = 1) AND
(BODY_light = 1) AND (FIN_dry = 1)
```

Therefore, we may conclude: J cluster is distinguished by Scotches having either:

1. A round body and a sherry nose, or

2. A full gold (but not red) color with a light (but not round) body and a dry finish.

Two sets of descriptions for Cluster J:

- *Characteristic Description*: represented by the cluster centroid, describing typical characteristics of the cluster, ignoring whether the other clusters may share some of those characteristics

  Intragroup Commonalities

- *Differential Description*: derived from a decision tree, describing what differentiates this cluster from the others, ignoring some commonalities of the members within the cluster

  Intergroup Differences

Neither is inherently better — it depends on what you are using it for.

One more example: Credit line optimization

Cluster the existing customers based on similarity in their use of their cards, payment of their bill, and profitability of the company, leading to 5 clusters representing very different consumer behaviour (e.g. spending a lot but paying off the cards in full each month, spending a lot but keeping their balance near their credit limit).

Those different clusters can tolerate different credit lines.

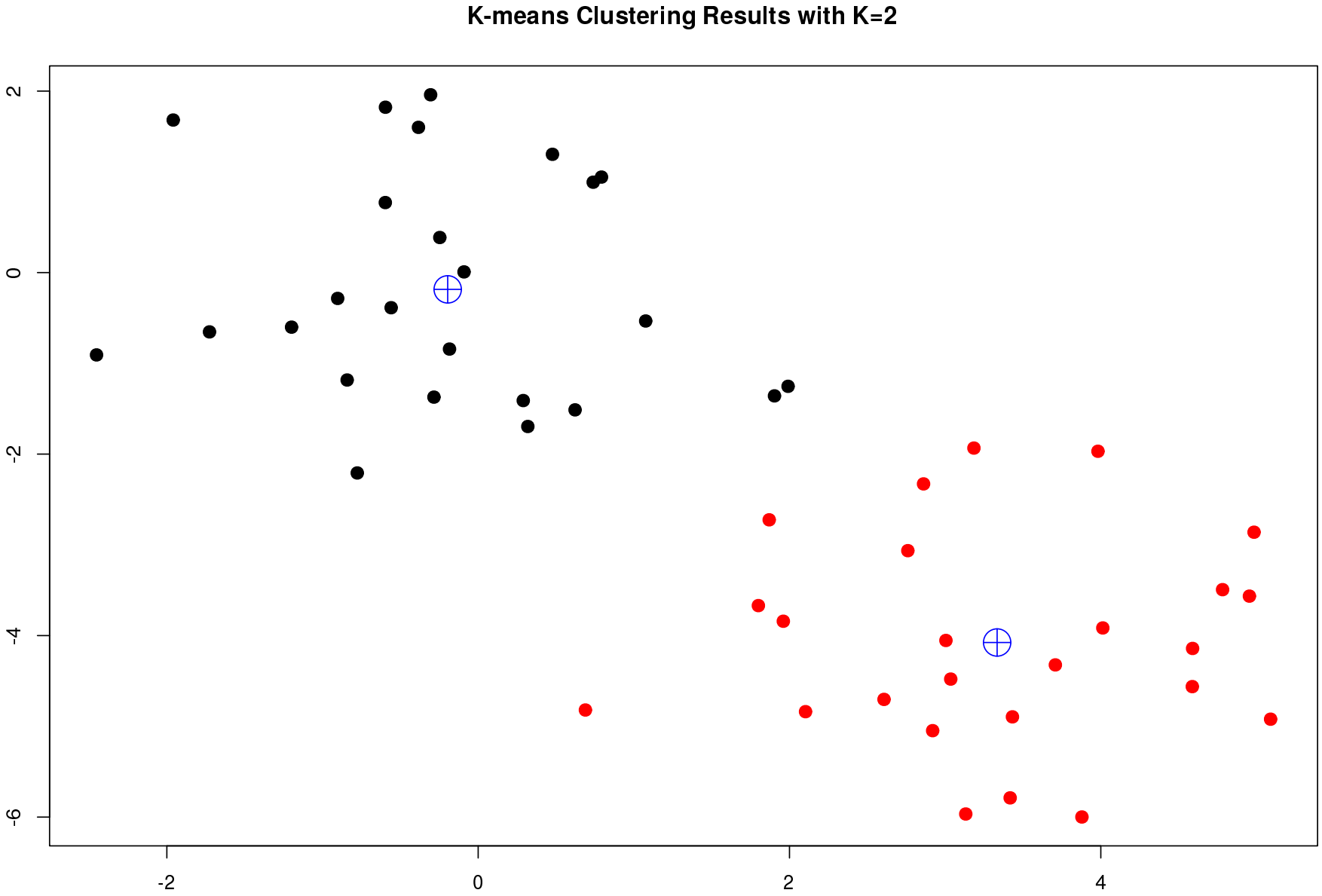
However the data used for clustering are not available for new customers to whom credit lines need to be assigned.

Use the data available at the time of credit approval to build a supervised learning model to classify customers into the 5 different clusters. This model then can be used to improve initial credit line decision.

Reference: Haimowitz, I., & Schwartz, H. (1997). Clustering and prediction for credit line optimization. In Fawcett, Haimowitz, Provost,& Stolfo (Eds.), AI Approaches to Fraud Detection and Risk Management, pp. 29-33. AAAI Press. Available as Technical Report WS-97-07.

In R function `kmeans` perform K-means clustering analysis. We start with a simple simulation example.

```
> x=matrix(rnorm(100), ncol=2) # 50x2 matrix containing indep N(0, 1) r.v.s.
> x[1:25,1]=x[1:25,1]+3
> x[1:25,2]=x[1:25,2]-4  # change the 1st 25 points to mean (3,-4)
> km.out=kmeans(x,2,nstart=20) # perform K-means with K=2, and 20 initial values
> km.out$cluster
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1
 [31] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> km.out$centers
        [,1]       [,2]
1 -0.1956978 -0.1848774
2  3.3339737 -4.0761910
> plot(x, col=(km.out$cluster), main="K-means Clustering Results with K=2", xlab="",
   ylab="", pch=20, cex=2)
> points(km.out$centers, col='blue', pch=10, cex=3) # adding two centroid points
```


K-means Clustering Results with K=2
2
0
-2
-4
-6
-2
0
2
4

What will happen if we set $K = 3$:

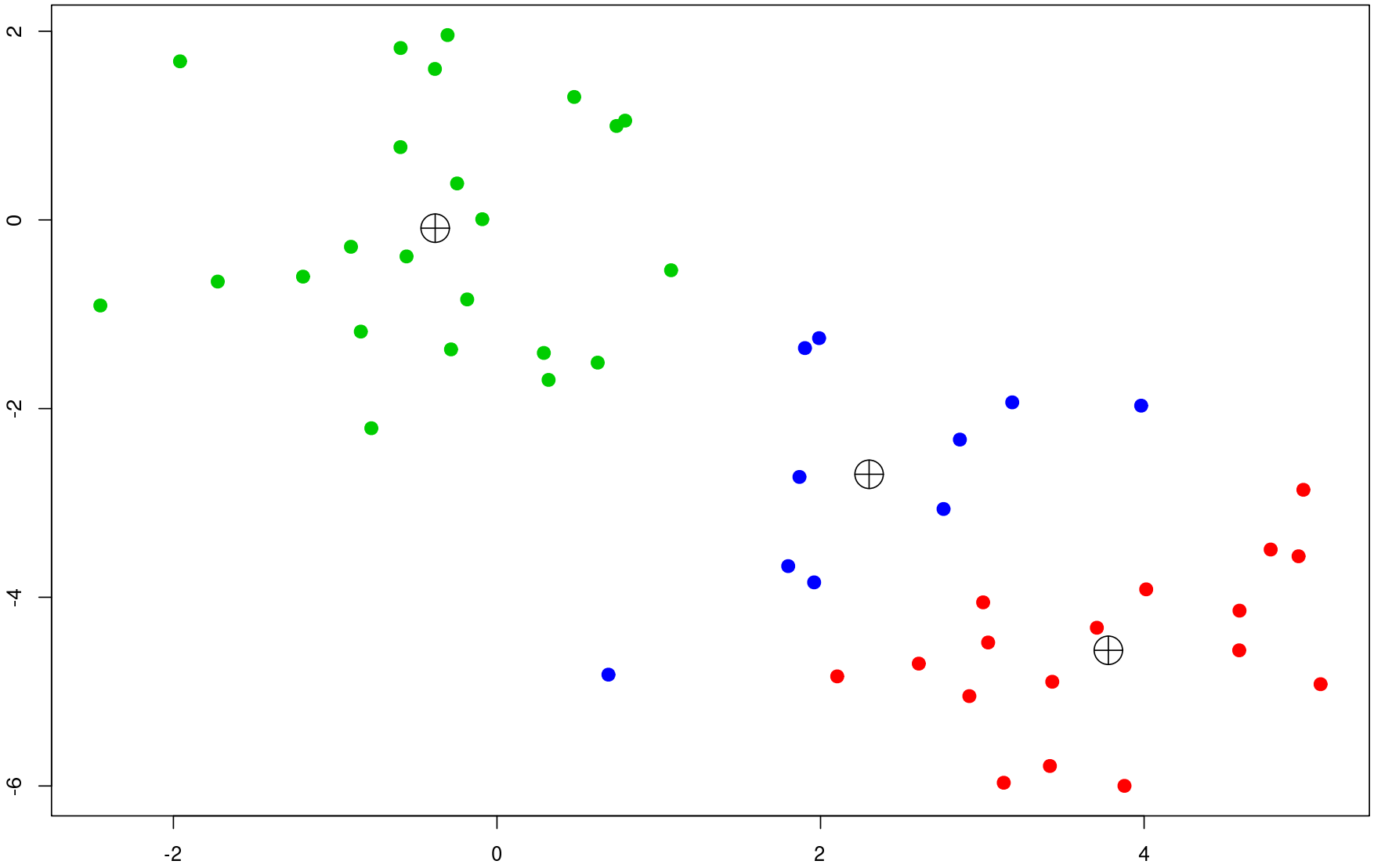
```
> km.out2=kmeans(x, 3, nstart = 20)
> km.out2
K-means clustering with 3 clusters of sizes 17, 23, 10
Cluster means:
        [,1]        [,2]
1  3.7789567 -4.56200798
2 -0.3820397 -0.08740753
3  2.3001545 -2.69622023

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2
 [31] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.74089 52.67700 19.56137
 (between_SS / total_SS =  79.3 %)

> plot(x, col=(km.out2$cluster+1), main="K-Means Clustering Results with K=3",
     xlab="", ylab="", pch=20, cex=2)
> points(km.out2$centers, pch=10, cex=3)
```

Note. It is always a good idea to use multiple initial values, i.e. set `nstart`> 1.


K-Means Clutering Results with K=3
2
0
-2
-4
-6
-2
0
2
4

We continue with dataset `customers` analysed in Chapter 6. It contains 8 attributes of 440 clients of a wholesale distributor: the first two attributes are `Channel` (i.e. Horeca or Retail) and `Region`. The other 6 variables are annual spendings on 6 categories of products. Now we use only the last 6 variables for clustering the 440 customers into, say, 5 clusters.

```
> > customers=read.csv("wholesaleCustomers.csv")
> dim(customers)
[1] 440   8
> customer6=customers[,3:8]
> km.custermers=kmeans(customer6, 5, nstart=20)
> table(Channel, km.customer$cluster, deparse.level = 2)
       km.customer$cluster
Channel   1   2   3   4   5
      1 195  13  83   0   7
      2  38   0  23   7  74
```

This is interesting, as Clusters 1, 2, 3 are dominated by Horeca customers while Clusters 4 and 5 are dominated by Retail customers. The percentages of Horeca customers in each clusters are:

```
> tab=table(Channel, km.customer$cluster, deparse.level = 2)
> for(i in 1:5) print(tab[1,i]/sum(tab[,i]))
[1] 0.8369099
```

```
[1] 1
[1] 0.7830189
[1] 0
[1] 0.08641975
```

This can be done more efficiently (i.e. avoiding the for-loop) as follow.

```
> tab.csum=apply(tab, 2, sum) # calculate sum for all columns of tab
> tab[1,]/tab.csum
        1         2         3         4         5
0.83690987 1.00000000 0.78301887 0.00000000 0.08641975
```

The wholesales distributor may provide different services to the 5 different clusters, according to their different spending profiles.

Also note those clusters tell little about customers' regions

```
> table(Region, km.customer$cluster, deparse.level = 2)
      km.customer$cluster
Region   1   2   3   4   5
     1  43   3  16   1  14
     2  23   1  11   1  11
     3 167   9  79   5  56
```

```
> table(Region)
Region
  1   2   3
 77  47 316
```

There are 316 customers in Region 3, and Clusters 1, 3, 5 have more customers from each of the three regions than the two small clusters.

Note. One can run K-means method based on other distance measures, check out `?kmeans`.

One also can use Mahalanobis distance in K-means clustering: this requires to standard the data first. For the above example, we use `customer6N` instead of `customer6`. See Chapter 6 for how the normalised data `customer6N` is defined.

Hierarchical clustering can be performed using function `hclust` together with function `cutree`. The input should be a distance matrix such as an output of function `dist`.

In the illustration below, we use rescaled 7 variables to cluster 440 customers.

```
> customersU=scale(customers)
    # make each columns of customers have variance 1
> hcC.customers=hclust(dist(customersU[,-2]), method="complete")
    # exclude Region in analysis. One may change complete to use
    # other linkage methods, check ?hclust
> hcC.index20=cutree(hcC.customers, 20) # cut tree with 20 terminal nodes
            # can also cut tree at certain height, check ?cutree
> table(Channel, hcC.index20, deparse.level = 2)
       hcC.index5
```

```
Channel   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
      1   0 208  43   0   0  15   0   0   0  20   2   0   0   1   2
      2  89   0  10   1   6   0  23   2   5   3   0   1   1   0   0
Channel  16  17  18  19  20
      1   1   1   4   1   0
      2   0   0   0   0   1
> hcC.index10=cutree(hcC.customers, 10)
> table(Channel, hcC.index10, deparse.level = 2)
       hcC.index10
Channel   1   2   3   4   5   6   7   8   9  10
      1 275   2  17   0   0   0   2   1   1   0
      2 108   1   0  28   3   1   0   0   0   1
```

One can plot the tree via `plot(hcC.customers)`. However with 440 individuals, the tree is too big to be plotted on a piece of paper.