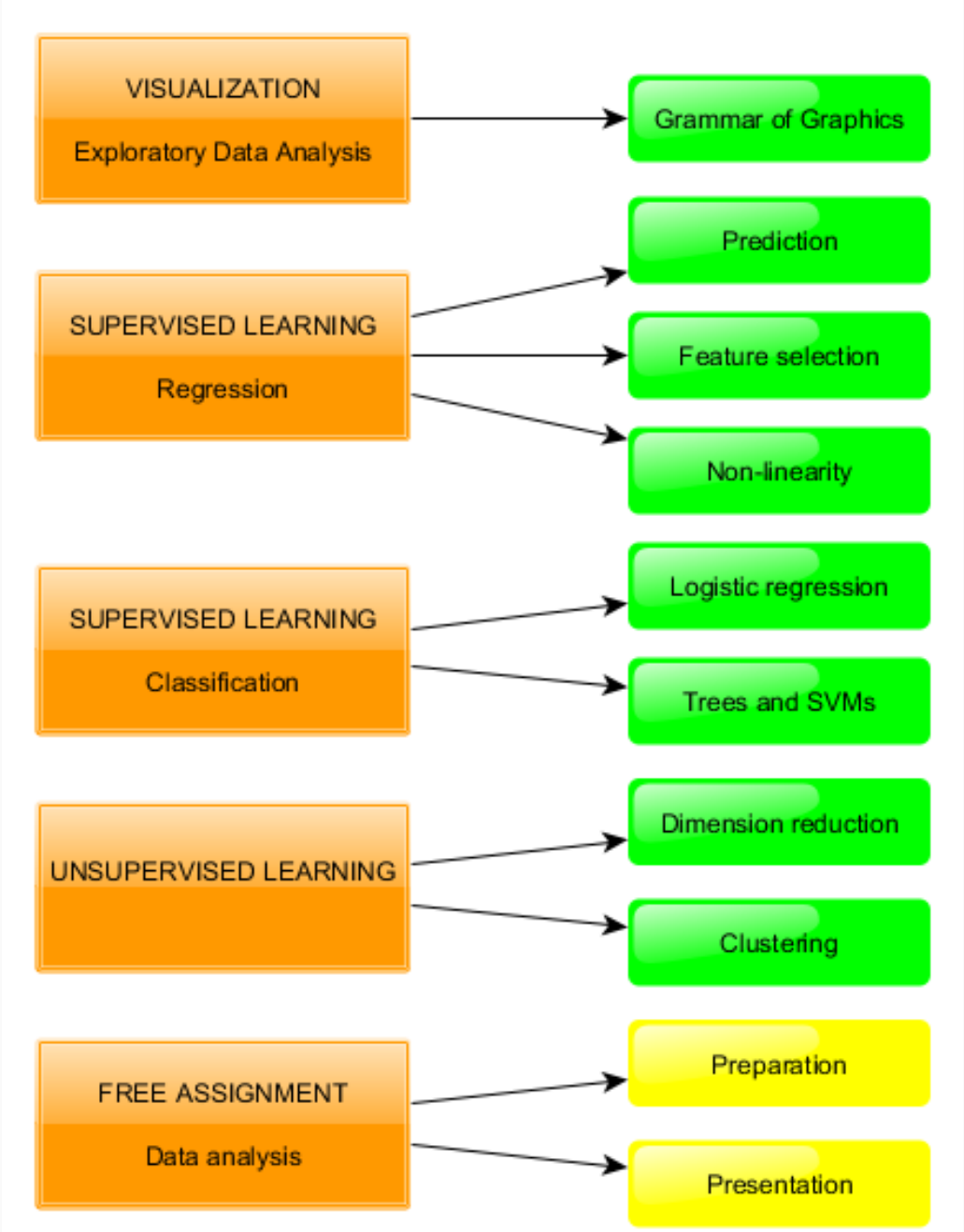
Unsupervised Learning

Clustering

Maarten Cruyff

Program

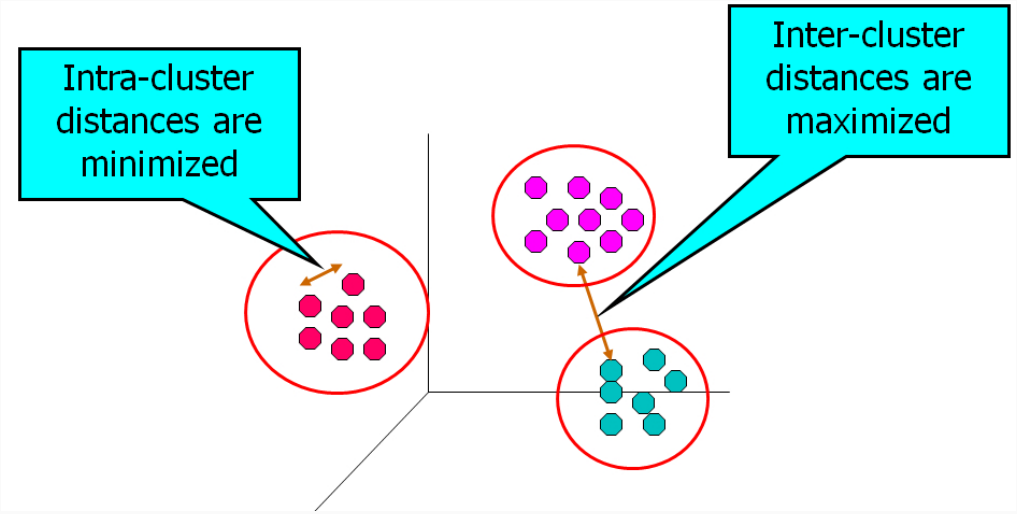

VISUALIZATION
Exploratory Data Analysis
Grammar of Graphics
SUPERVISED LEARNING
Regression
Prediction
Feature selection
Non-linearity
SUPERVISED LEARNING
Classification
Logistic regression
Trees and SVMs
UNSUPERVISED LEARNING
Dimension reduction
Clustering
FREE ASSIGNMENT
Data analysis
Preparation
Presentation

Content

1. K-means clustering
2. Hierarchical clustering

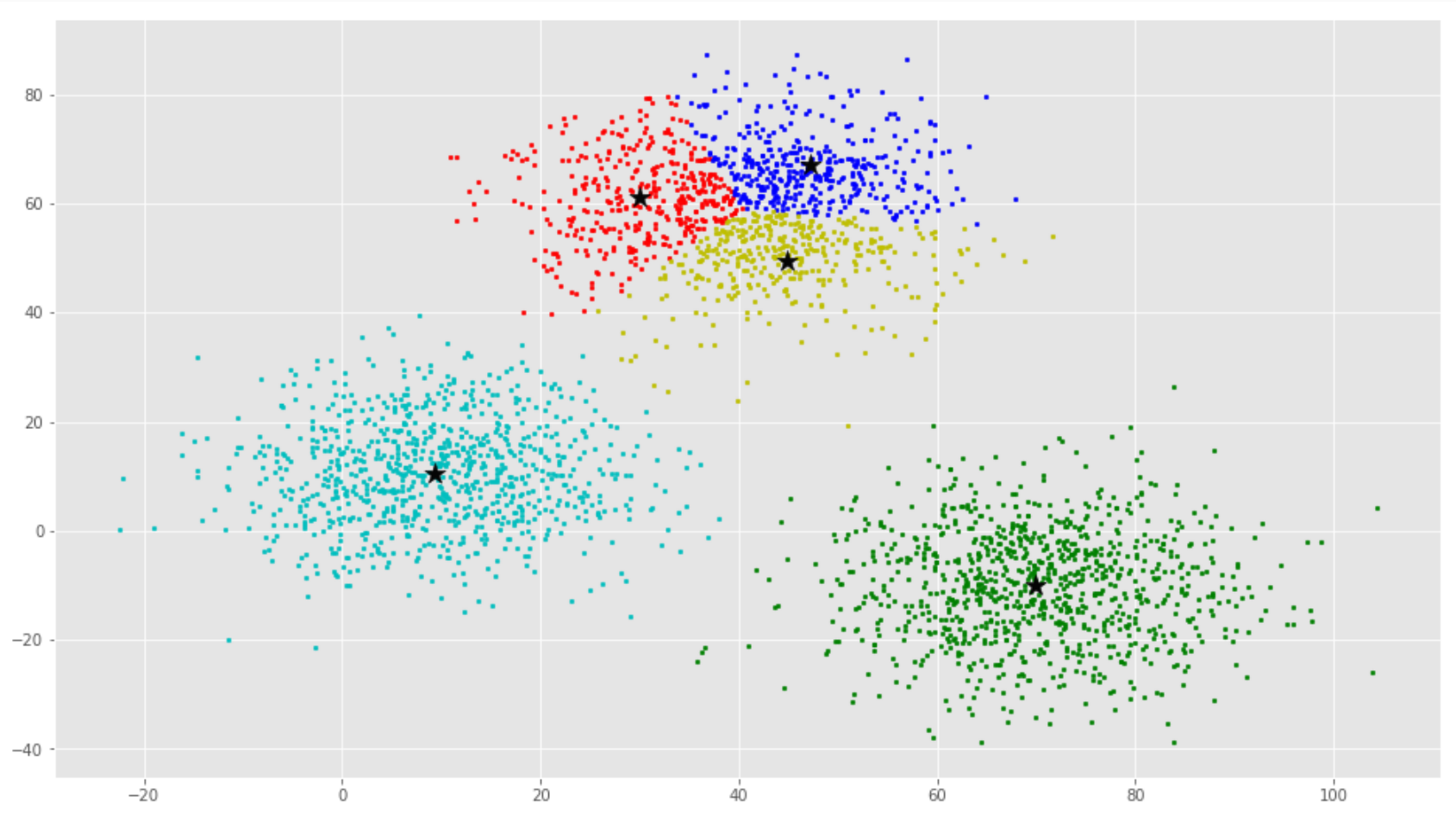
K-means algorithm

1. Specify the total number of clusters K
2. Randomly assign each observation to a cluster
3. Compute cluster centroids
4. Reassign observations to cluster with closest centroid
5. Repeat 3 and 4 untill convergence



Solution with 5 centroids

Would a solution with 3 clusters been better?



Issues

Determination optimal number of clusters

- elbow criterion for within-cluster SS

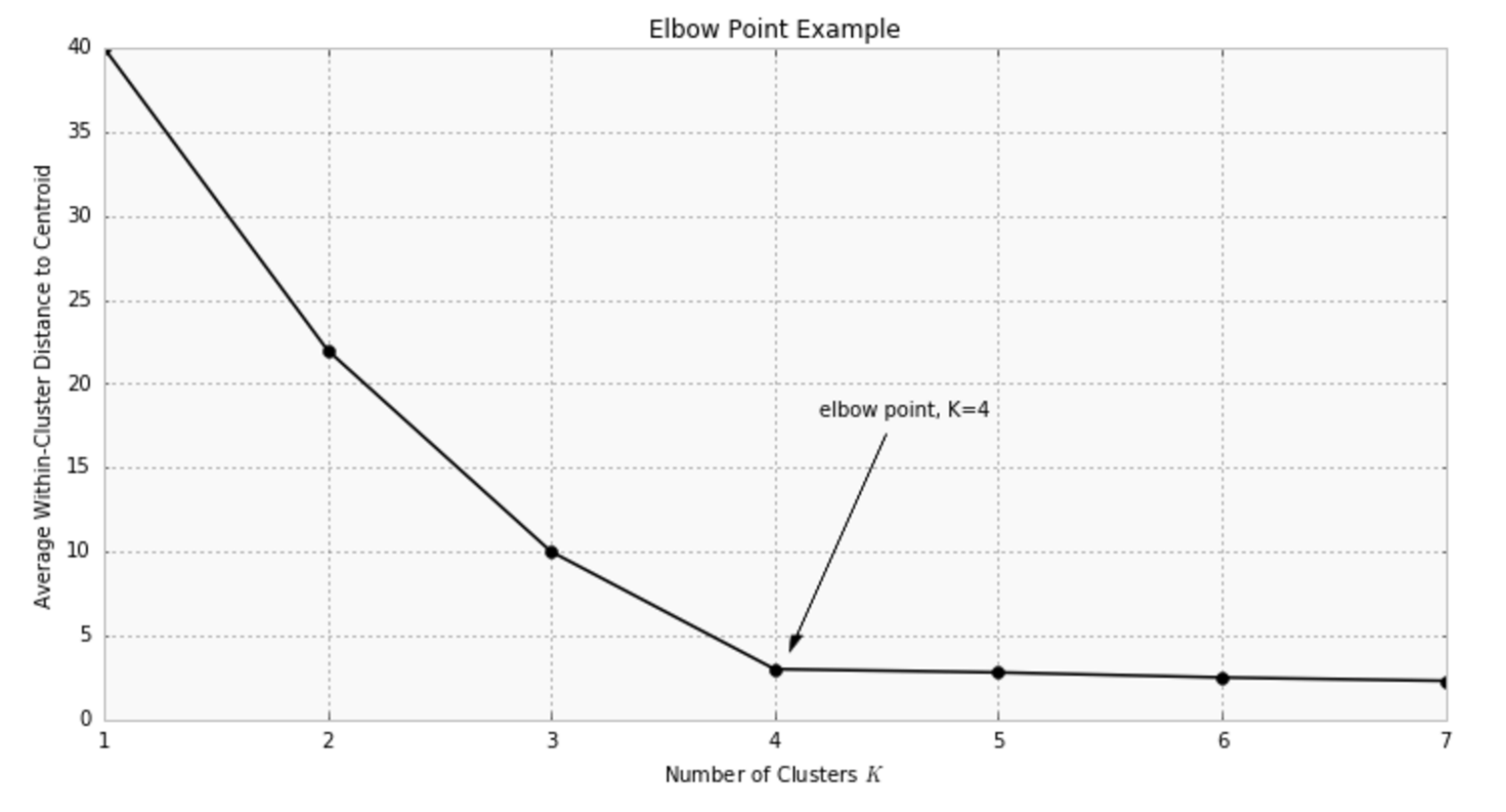
Solution is local minimum of within-cluster SS

- try out multiple starting values, e.g.

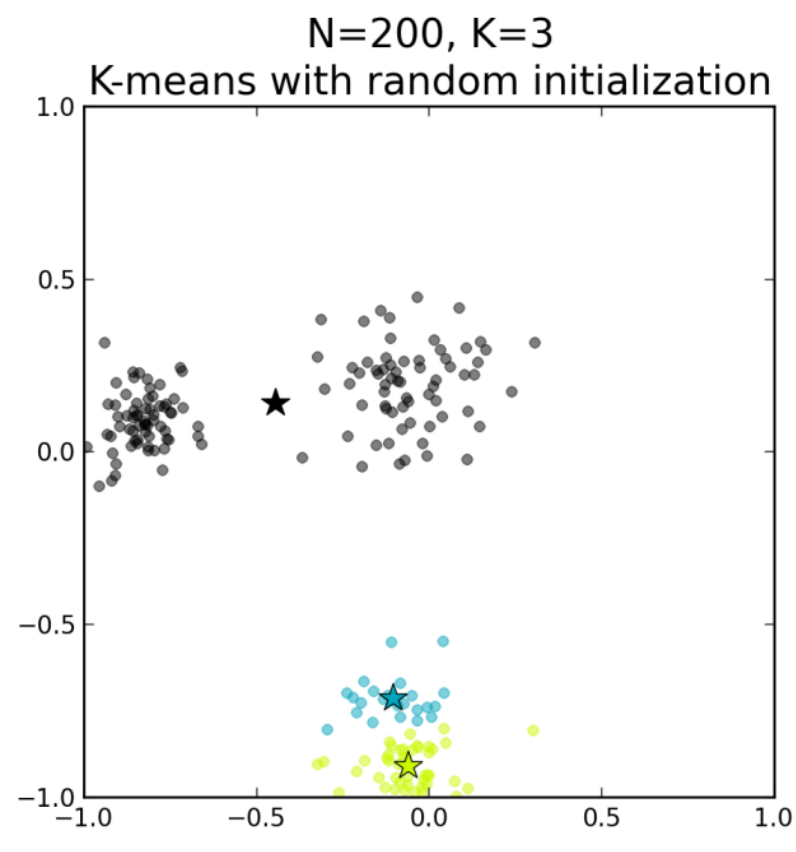
Scaling

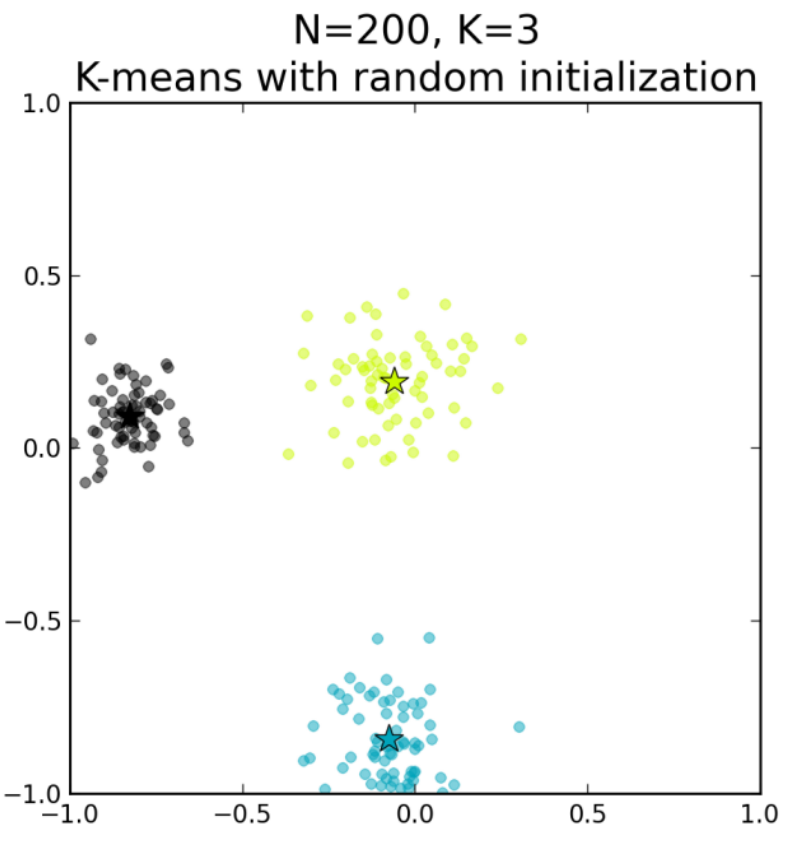
- standardization of features (same scale)
- principal components scores (noise reduction)

Elbow criterion

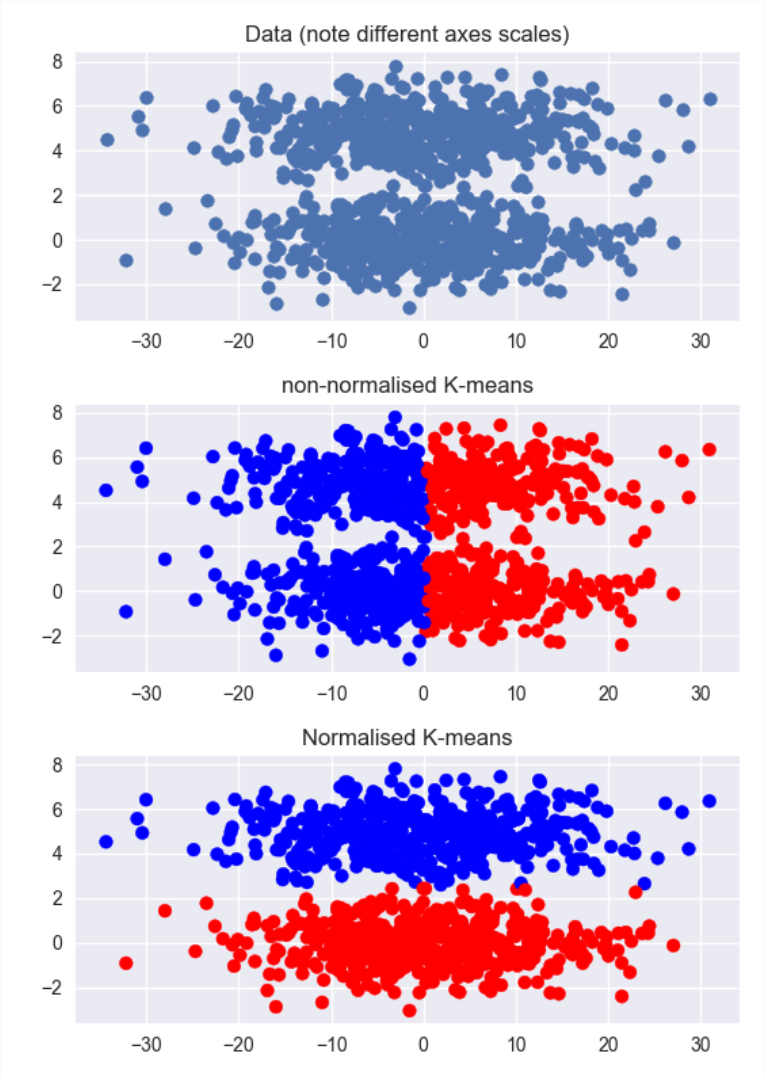


Local minimum





Scaling


Data (note different axes scales)
8
6
4
2
0
−2
−30
−20
−10
0
10
20
30
non-normalised K-means
8
6
4
2
0
−2
−30
−20
−10
0
10
20
30
Normalised K-means
8
6
4
2
0
−2
−30
−20
−10
0
10
20
30

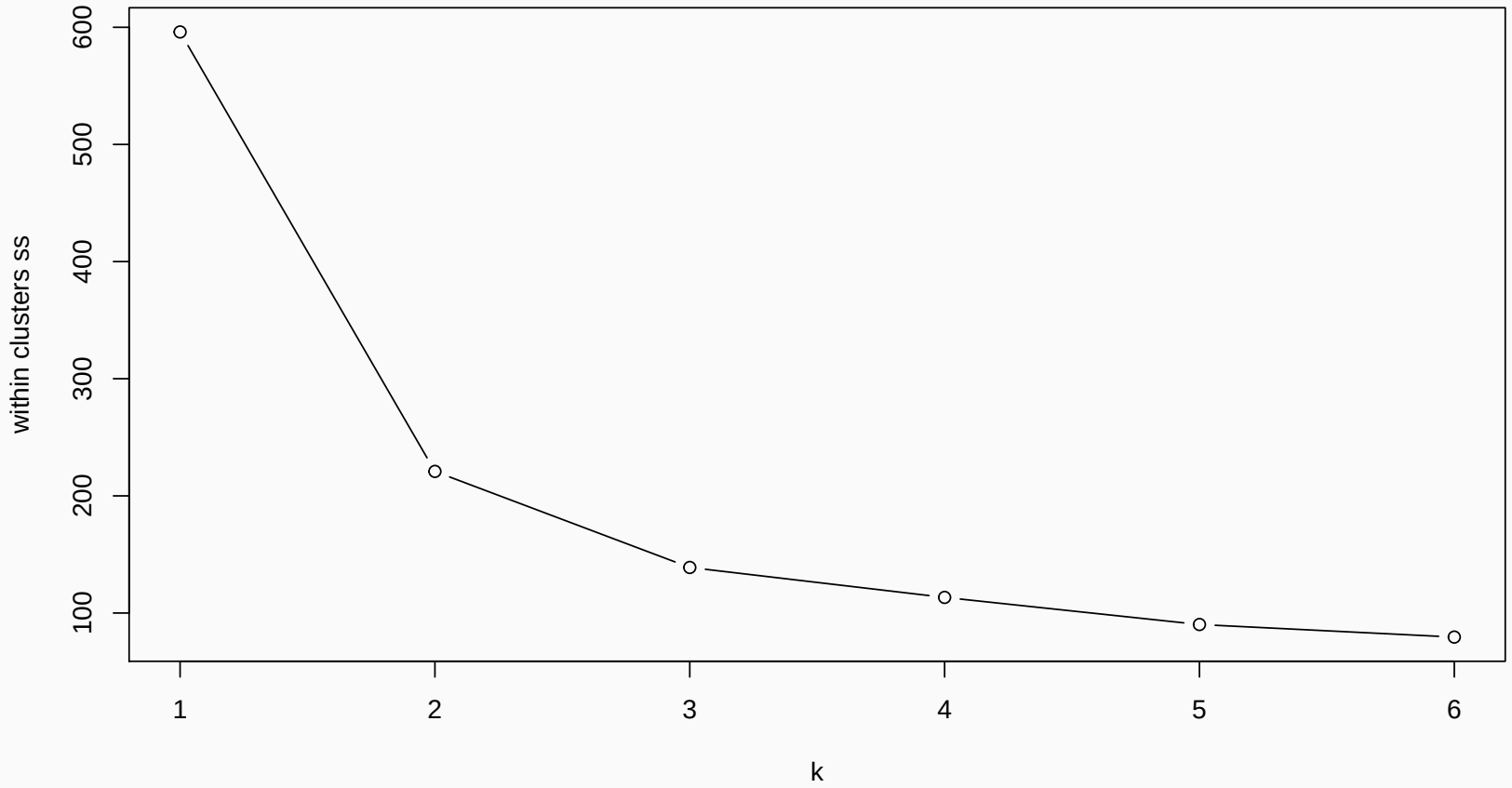
K-means in R

```
k_fit <- kmeans(x, centers, nstart = 1)  # scale(x) for standardization

print(k_fit)   # print a summary

fitted(k_fit)  # centroids for each case

k_fit$withinsss # total within-cluster sum of squares
```

Let's look at examplekmeansoutput for theiris' data

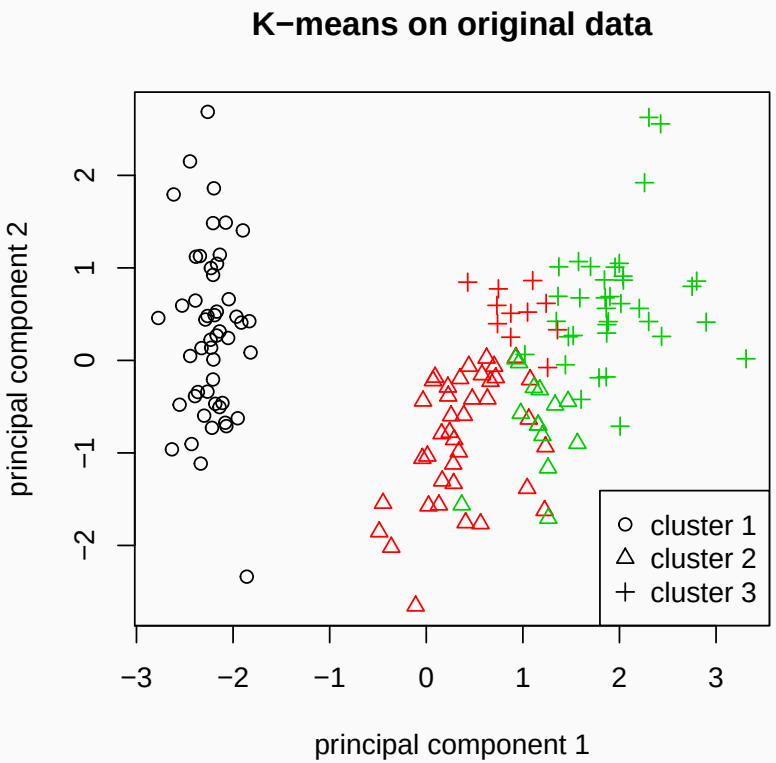
Screeplot `iris` data

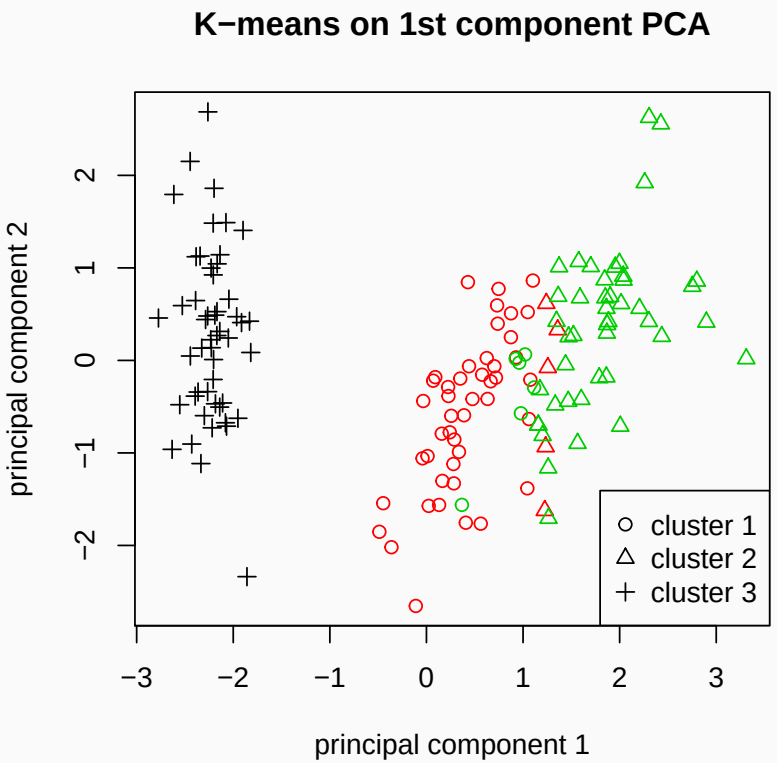
- 6 clusters
- 10 random starts



```
K-means clustering with 3 clusters of sizes 50, 53, 47

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -1.01119138  0.85041372   -1.3006301  -1.2507035
2  -0.05005221 -0.88042696    0.3465767   0.2805873
3   1.13217737  0.08812645    0.9928284   1.0141287

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 2 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 2
 [71] 3 2 2 2 2 3 3 3 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3
[106] 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 3 3 3 3 3 3 2 2 3 3 3 2 3
[141] 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 47.35062 44.08754 47.45019
 (between_SS / total_SS =  76.7 %)

Available components:

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

Example `iris` data





Confusion matrices

K-means on scaled `iris` data

	1	2	3
setosa	50	0	0
versicolor	0	39	11
virginica	0	14	36

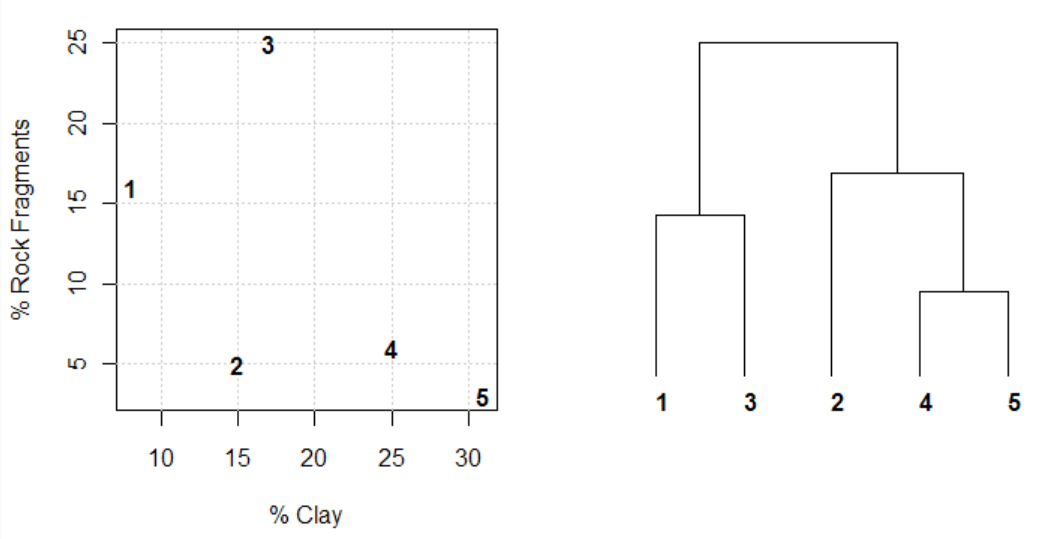
K-means on 1st principal component

	3	1	2
setosa	50	0	0
versicolor	0	45	5
virginica	0	6	44

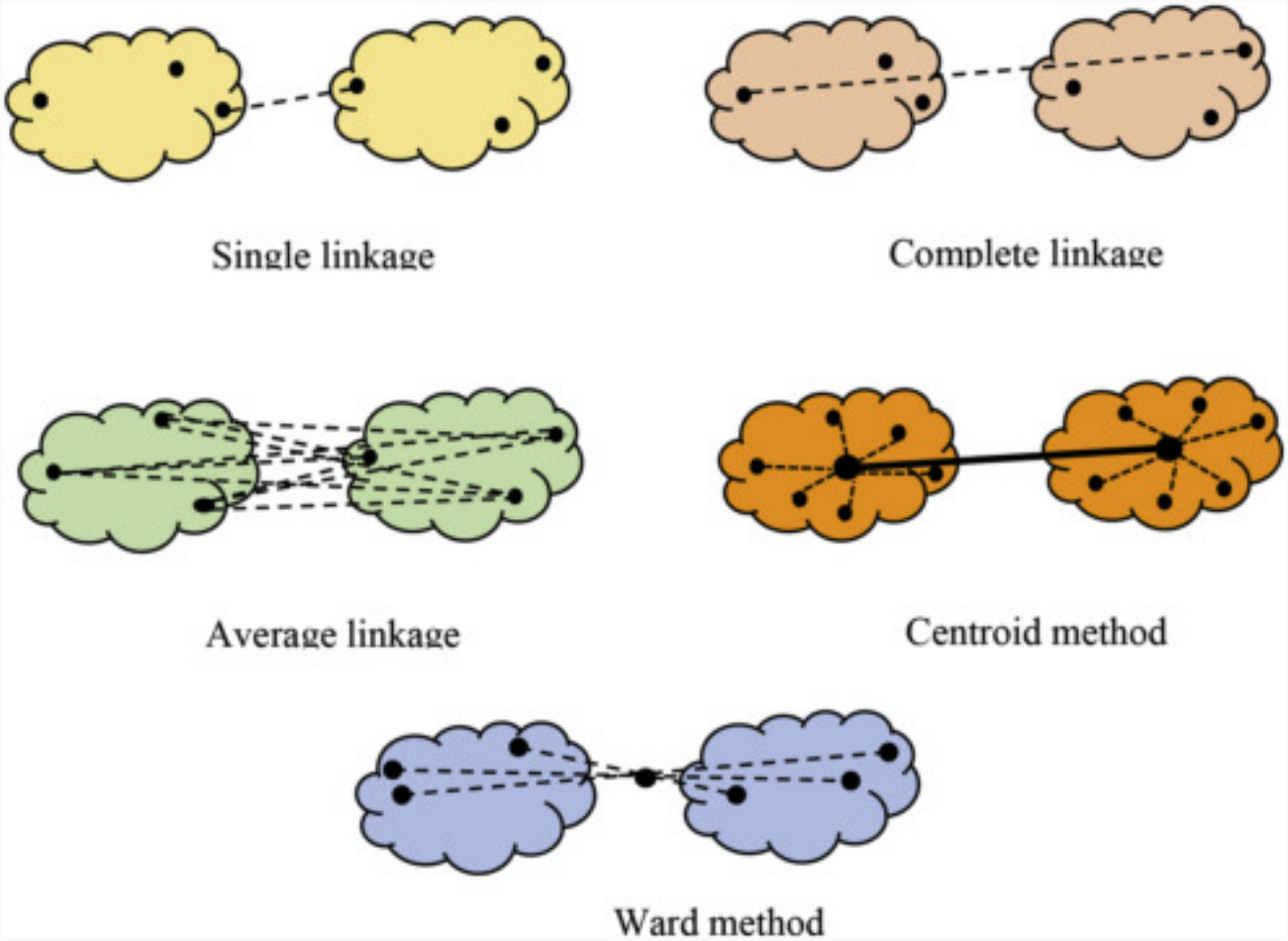
Hierarchical clustering

Algorithm (bottom-up and greedy)

1. Treat each observation as a cluster
2. Compute *distances* between all $\binom{n}{2}$ cluster pairs
3. Link pair with smallest *distance* in new cluster
4. Repeat 2-3 until 2 clusters left
5. Plot dendogram

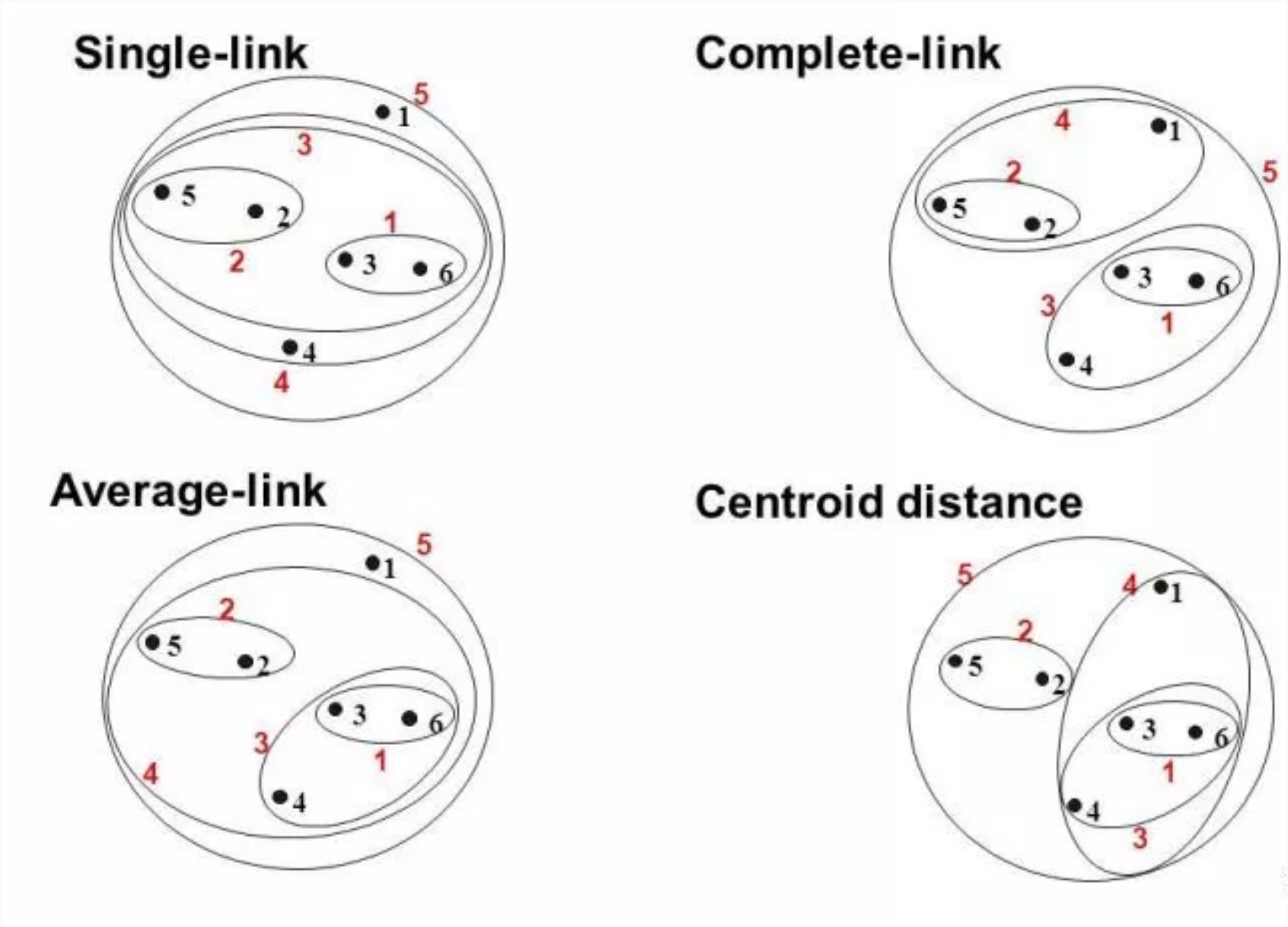


Linkage methods



and others!

Effect linkage



Distance measures

Euclidean

- distance between points

$$\sum_i \sqrt{(a_i - b_i)^2}$$

Mahalanobis

- distance between profiles

$$\sqrt{(a-b)' \boldsymbol{S}^{-1} (a-b)}$$

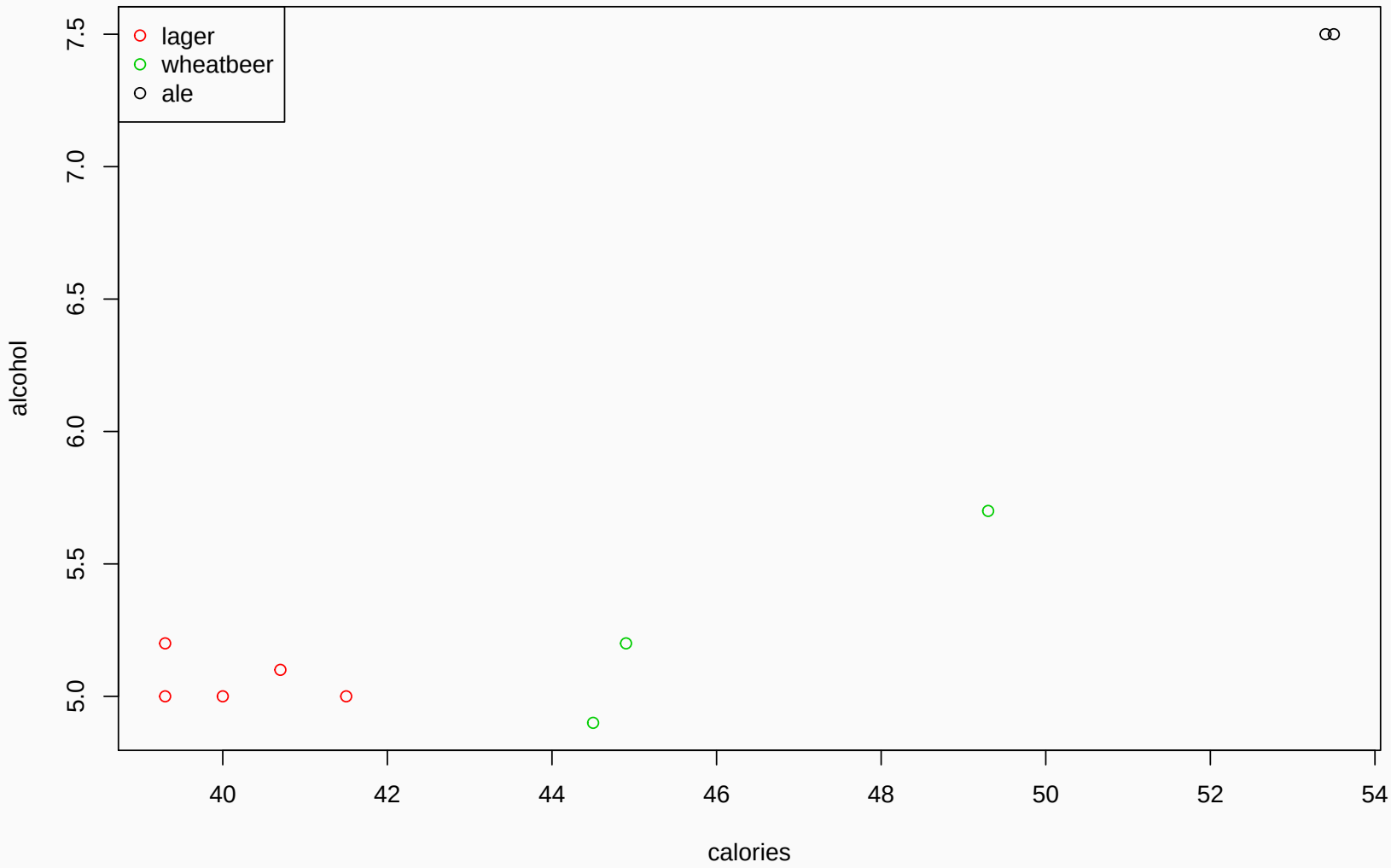
and others!

Hierachical clustering in R

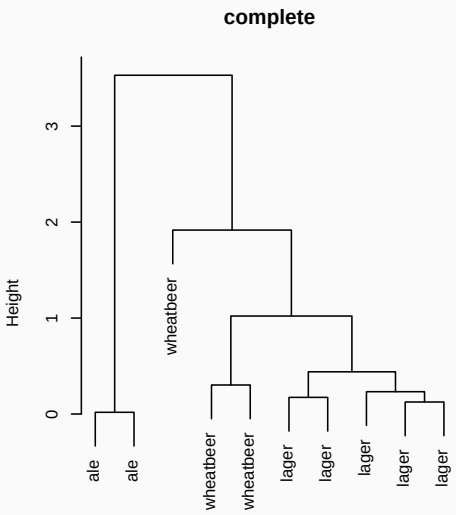
```
hclust(x = dist(scale(<data>), method = "euclidean"),
       method = c("complete",
                  "ward.D2",
                  "single",
                  "average",
                  "centroid")

plot(<hclust object>,
     labels = <names observations>) # default is row names
```

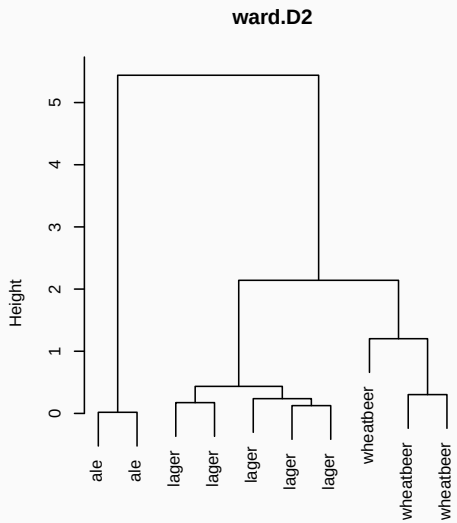
Beer example

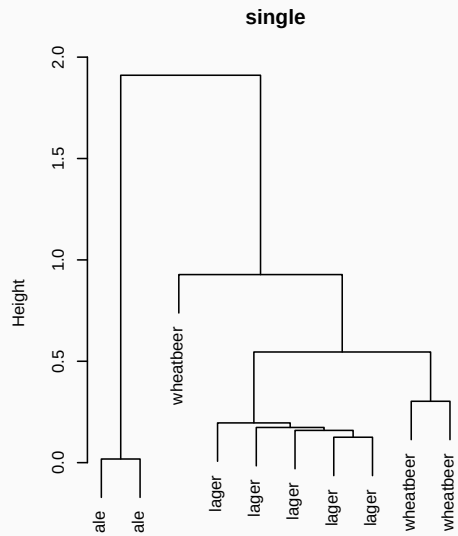
Cluster 10 beers (3 types) on calories and alcohol

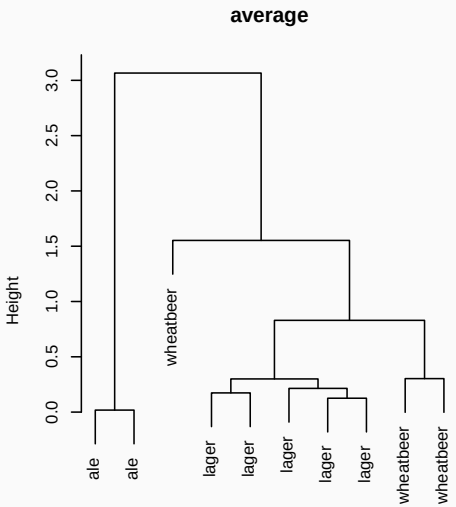


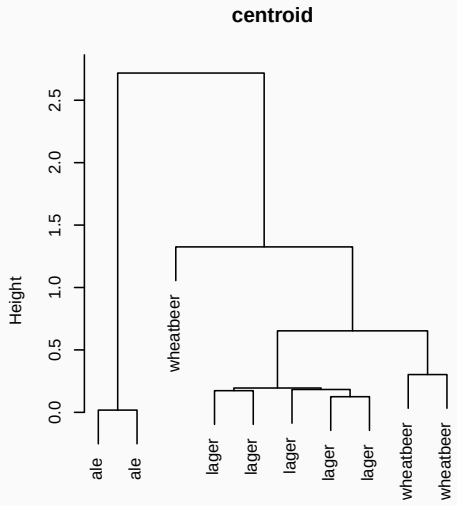
Comparisons











How many clusters?

- large distances between and small distance within