Clustering

- Divide the data \((x_1, \ldots, x_n)\) into groups
- Groups consist of similar objects (observations)
- Contrast to classification (discrimination)
  1. Classification with predetermined classes (supervised, label \(Y\) available)
  2. Clustering to determine classes (unsupervised)
- “Unsupervised learning”: data segmentation, class discovery—examples include
  1. Marketers use demographics and consumer profiles to segment the marketplace into small, homogeneous groups
  2. Physicians use medical records to cluster patients for personalized treatment
  3. Pandora and Netflix use viewing history to group viewers/listeners to recommend next songs and movies
Example: Old Faithful Geyser

- Section 12.1.2 Izenman
- 107 bivariate observation for duration of eruption ($X_1$) and the waiting time until the next eruption ($X_2$).
- Can this dataset be divided into two or three sub-groups?
Old Faithful Geyser

- Human perception is excellent??
Old Faithful Geyser

- Same data, same clustering, but with different axis.
- Is human perception really excellent?? (Depend on vis.)
Example: mRNA expression profiling

- Bhattacharjee et al (2001) PNAS
- Preprocessed gene expressions with $d = 2530$ genes and $n = 56$ subjects with lung cancer.
- Subgroup for different types of lung cancers?
mRNA expression profiling

- Clustering by human eyes requires a good way to visualize the data: Use PCA (scatterplot matrix for PC scores 1–3)
• Again, is human perception excellent?
• Yes, for this data, compare to true subgroups
Ingredients for clustering

- Need a distance measure for different observations similarity or dissimilarity measure
- Quality of clusters, number of clusters?
Clustering algorithms

1. Combinatorial algorithm
2. Partitioning methods: $K$-means, $K$-medoids, Partitioning around medoids, Fuzzy Analysis
3. Hierarchical clustering
   agglomerative, divisive
4. Model-based clustering
   e.g. Normal mixture

All methods (with exception of a few) allow to use only dissimilarity measures. For now, assume data are quantitative, i.e., $x_1, \ldots, x_n \in \mathbb{R}^d$
Dissimilarity and within-cluster scatter

Given data \( \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d \),

- Dissimilarity \( d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j) \), for example:
  1. the usual 2-norm \( d(\mathbf{x}, \mathbf{y}) = \| \mathbf{x} - \mathbf{y} \|_2 \),
  2. 1-norm \( d(\mathbf{x}, \mathbf{y}) = \| \mathbf{x} - \mathbf{y} \|_1 \) (taxi driver’s distance)
  3. \( p \)-norm \( d(\mathbf{x}, \mathbf{y}) = \| \mathbf{x} - \mathbf{y} \|_p = \left( \sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}, \quad p > 0 \).

- Clustering is an assignment function
  \( c(i) : \{1, \ldots, n\} \rightarrow \{1, \ldots, K\} \), where \( K \) is the number of points.

- Within-cluster scatter:

  \[ W(c) = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \cdot c(i) = k} \sum_{j \cdot c(j) = k} d(i, j), \]

  where \( n_k = \#\{i : c(i) = k\} \) number of points in cluster \( k \).

- Small \( W \) is better.
Combinatorial algorithm

- One needs to minimize $W$ over all possible assignments of $n$ points to $K$ clusters.
- The number of distinct assignments is

$$A(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n \approx K^n.$$  

- Not so easy for large $K$ and $n$

  $n = 25$ observations, $K = 4$ clusters: $A(n, K) \geq 10^{13}$

- It calls for more efficient algorithm: may not be optimal but reasonably good sub-optimal solutions.
K-means algorithm: motivation

- In need of an efficient algorithm to (approximately) minimize $W$ among all clustering $c$
- Another look at $W$, with $d(i, j) = \|x_i - x_j\|_2^2$ (squared Euclidean distance):

$$\frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i: c(i) = k} \sum_{j: c(j) = k} \|x_i - x_j\|_2^2 = \sum_{k=1}^{K} \sum_{i: c(i) = k} \|x_i - \bar{x}(k)\|_2^2,$$

$$= \min_{m_1, \ldots, m_K} \sum_{k=1}^{K} \sum_{i: c(i) = k} \|x_i - m_k\|_2^2,$$

where $\bar{x}(k) = \frac{1}{n_k} \sum_{i: c(i) = k} x_i$ (average of all points in cluster $k$)

- Idea is to minimize the enlarged criterion $W(c, m_1, \ldots, m_K) = \sum_{k=1}^{K} \sum_{i: c(i) = k} \|x_i - m_k\|_2^2$ by alternately minimizing over $c$ and over $m_1, \ldots, m_K$:
K-means algorithm

K-means algorithm

The algorithmic iteration begins with an initial guess for \( K \) cluster centers \((m_1, \ldots, m_K)\).

1. Minimize over \( c \): For each \( x_i \) \((i = 1, \ldots, n)\), find the cluster center \( m_k \) closest to \( x_i \), then update \( c(i) = k \).
2. Minimize over \( m_1, \ldots, m_K \): For each cluster, update \( m_k \) by the new average of points in cluster \( k \).
3. Iterate Steps 1 and 2 until \( W(c) \) does not change.

Variation on K-means algorithm: When \( \rho(x_i, x_j) \) is used other than the squared Euclidean distance.

- in Step 1, closest cluster center is found by \( \rho \)
- in Step 2, average is appropriately defined by \( \rho \)
**$K$-means example**

- First two iterations of the algorithm
**K-means example**

- First two iterations of the algorithm (*with different initial*)
**K-means example**

- First two iterations of the algorithm (*another different initial*)
K-means algorithm

- Each iteration always minimizes $W(c, m_1, \ldots, m_K)$
- Iteration always finishes (converges)
- Different initial values may lead to different solutions.
- $K$-means is typically run multiple times, with random initial values for each run. Final solution is chosen among from the collection of centers based on which one gives the smallest within-cluster scatters
- Still sub-optimal than the combinatorial method
- Works well for quantitative variables
When K-means is not preferred

- In K-means, each cluster is represented by the centroid 
  \( m_k = \) the average of all points in \( k \)th cluster 
  In the geyser example, each centroid is a good representative
- In some applications
  1. we want each cluster represented by one of the points in the cluster
  2. we only have pairwise dissimilarities \( d_{ij} \) but do not have actual points (thus no averaging)
- This is where \textit{K-medoids} comes in (two slides later)
- Country Dissimilarities example: Kaufman and Rousseeuw (1990)
  A study in which political science students were asked to provide pairwise dissimilarity measures for 12 countries: Belgium, Brazil, Chile, Cuba, Egypt, France, India, Israel, United States, Union of Soviet Socialist Republics, Yugoslavia and Zaire (Data next slide)
Country Dissimilarities

- The average dissimilarity scores are given.
- K-means clustering could not be applied because we have only distances rather than raw observations.

**TABLE 14.3.** Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.

<table>
<thead>
<tr>
<th></th>
<th>BEL</th>
<th>BRA</th>
<th>CHI</th>
<th>CUB</th>
<th>EGY</th>
<th>FRA</th>
<th>IND</th>
<th>ISR</th>
<th>USA</th>
<th>USS</th>
<th>YUG</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRA</td>
<td>5.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHI</td>
<td>7.00</td>
<td>6.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUB</td>
<td>7.08</td>
<td>7.00</td>
<td>3.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGY</td>
<td>4.83</td>
<td>5.08</td>
<td>8.17</td>
<td>5.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRA</td>
<td>2.17</td>
<td>5.75</td>
<td>6.67</td>
<td>6.92</td>
<td>4.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>6.42</td>
<td>5.00</td>
<td>5.58</td>
<td>6.00</td>
<td>4.67</td>
<td>6.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISR</td>
<td>3.42</td>
<td>5.50</td>
<td>6.42</td>
<td>6.42</td>
<td>5.00</td>
<td>3.92</td>
<td>6.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>2.50</td>
<td>4.92</td>
<td>6.25</td>
<td>7.33</td>
<td>4.50</td>
<td>2.25</td>
<td>6.33</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USS</td>
<td>6.08</td>
<td>6.67</td>
<td>4.25</td>
<td>2.67</td>
<td>6.00</td>
<td>6.17</td>
<td>6.17</td>
<td>6.92</td>
<td>6.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YUG</td>
<td>5.25</td>
<td>6.83</td>
<td>4.50</td>
<td>3.75</td>
<td>5.75</td>
<td>5.42</td>
<td>6.08</td>
<td>5.83</td>
<td>6.67</td>
<td>3.67</td>
<td></td>
</tr>
<tr>
<td>ZAI</td>
<td>4.75</td>
<td>3.00</td>
<td>6.08</td>
<td>6.67</td>
<td>5.00</td>
<td>5.58</td>
<td>4.83</td>
<td>6.17</td>
<td>5.67</td>
<td>6.50</td>
<td>6.92</td>
</tr>
</tbody>
</table>
K-medoids algorithm

K-medoids is similar to K-means, but searches for $K$ representative objects (medoids)

**K-medoids**

The algorithmic iteration begins with an initial guess for $K$ cluster medoids ($m_i \in \{x_1, \ldots, x_n\}$),

1. Minimize over $c$: For each $x_i$ ($i = 1, \ldots, n$), find the cluster medoids $m_k$ closest to $x_i$, then update $c(i) = k$.

2. Minimize over $m_1, \ldots, m_K$: Locate the medoid for each cluster. The medoid of the $k$th cluster is defined as that item in the $k$th cluster that minimizes the total dissimilarity to all other items within that cluster.

3. Iterate Steps 1 and 2 until $W(c)$ does not change.
**$K$-medoids example (Geyser)**

- First two iterations of the algorithm, using sq. Euc. distance as dissimilarity
- New set of medoids consists of original observations
**K-medoids example (Country)**

Survey of country dissimilarities. (Left panel:) dissimilarities reordered and blocked according to 3-medoid clustering. Heat map is coded from most similar (dark red) to least similar (bright red). (Right panel:) two-dimensional multidimensional scaling plot, with 3-medoid clusters indicated by different colors.
More partitioning methods

- Izenman discusses two other methods pam and fanny
- pam (partitioning around medoids) is a variation of K-medoids, by allowing swapping of medoids
- fanny (fuzzy clustering): instead of assigning clusters by the clustering function $c(i)$, probabilities $u_{ik}$ (of the $i$th point belonging to $k$th cluster) are assigned. The solution for probabilities are minimizing

$$
\sum_{k=1}^{K} \frac{\sum_i \sum_j u_{ik}^2 u_{jk}^2 d_{ij}}{2 \sum_{\ell} u_{\ell k}^2}
$$
Hierarchical clustering

Partitioning methods (K-means, K-medoids):

- fit $K$ clusters, for pre-determined number $K$ of clusters.
- Results of clustering depend on the choice of initial cluster centers
- No relation between clusterings from 2-means and those from 3-means.

Hierarchical clustering:

- does not depend on initial values – one and unique solution,
- gives clustering assignments for all $K = 1, \ldots, n$.
- has clear relationship between $(K - 1)$-cluster clusterings and $K$-cluster clustering
Agglomerative vs divisive

Two types of hierarchical clustering algorithms

**Agglomerative (bottom-up)**

- Start with all points in their own group
- Until there is only one cluster, repeatedly: merge the two groups that have the smallest dissimilarity

**Divisive (top-down)**

- Start with all points in one cluster
- Until all points are in their own cluster, repeatedly: split the group into two resulting in the biggest dissimilarity
Example: hierarchical clustering

Given airline distances in miles between seven major cities (n = 7, dissimilarity is the airline distance), a hierarchical clustering gives a clustering sequence:
Example: hierarchical clustering

The sequence of clustering assignments is visually represented by a *dendrogram*:

Note that cutting the dendrogram horizontally partitions the data points into clusters.
Dendrogram: Graphical representation of hierarchical sequence of clustering assignments.

- **Vertical axis**: distance between clusters
- **Horizontal axis**: observations
- Dendrogram is a binary tree where
  - each node represents a cluster
  - Each leaf node is the observation
  - Root node is a cluster with all observations
distance between clusters

Agglomerative (bottom-up) hierarchical clustering needs a measure of distance between two clusters.

- We have dissimilarities $d_{ij}$ between any pair of observations $i$ and $j$.
- Clusters $G_1 = \{1, 2, 4, 6\}$ and $G_2 = \{3, 5\}$ (an example)
- **Linkage**: function $d(G_1, G_2)$ that takes two groups $G_1, G_2$ and returns a dissimilarity score between them
  - **[Single linkage (nearest-neighbor linkage)]**
    
    $$d(G_1, G_2) = \min_{i \in G_1, j \in G_2} d_{ij}$$
  
  - **[Complete linkage (furthest-neighbor linkage)]**
    
    $$d(G_1, G_2) = \max_{i \in G_1, j \in G_2} d_{ij}$$
  
  - **[Average linkage]**
    
    $$d(G_1, G_2) = \text{Average}_{i \in G_1, j \in G_2} d_{ij} = \frac{\sum_{i \in G_1, j \in G_2} d_{ij}}{|G_1| \cdot |G_2|}$$
Agglomerative hierarchical clustering algorithm

Input $D = (d_{ij})$, the $n \times n$ (symmetric) matrix of dissimilarities $d_{ij} = d(x_i, x_j)$ between the $n$ clusters, given a linkage $d(G, H)$.

1. Merge two clusters $G$ and $H$ such that $d(G, H)$ is the smallest.
2. With the new cluster $GH$ and remaining clusters, repeat Step 1 until there is only one cluster.
A worked example: Airline distances

\( n = 7 \). Consider using Single Linkage:

D(7) =

\[
\begin{array}{cccccccc}
\text{Fr} & \text{HK} & \text{Lnd} & \text{Mnt} & \text{Mos} & \text{NY} & \text{Tk} \\
\text{Fr} & 0 & 8277 & 400 & 3640 & 1253 & 3851 & 9776 \\
\text{HK} & 8277 & 0 & 8252 & 10345 & 6063 & 10279 & 1788 \\
\text{Lnd} & 400 & 8252 & 0 & 3251 & 1557 & 3456 & 9536 \\
\text{Mnt} & 3640 & 10345 & 3251 & 0 & 5259 & 330 & 8199 \\
\text{Mos} & 1253 & 6063 & 1557 & 5259 & 0 & 5620 & 4667 \\
\text{NY} & 3851 & 10279 & 3456 & 330 & 5620 & 0 & 8133 \\
\text{Tk} & 9776 & 1788 & 9536 & 8199 & 4667 & 8133 & 0 \\
\end{array}
\]

Merge two clusters Mnt and NY as \( d(\text{Mnt}, \text{NY}) \) smallest

Compute new \((n - 1) \times (n - 1)\) dissimilarity matrix
A worked example: Airline distances

Compute new $6 \times 6$ dissimilarity matrix with $d(\text{MntNY}, \cdot)$ being the single linkage

$$D(6) =$$

<table>
<thead>
<tr>
<th></th>
<th>MntNY</th>
<th>Fr</th>
<th>HK</th>
<th>Lnd</th>
<th>Mos</th>
<th>Tk</th>
</tr>
</thead>
<tbody>
<tr>
<td>MntNY</td>
<td>0</td>
<td>3640</td>
<td>10279</td>
<td>3251</td>
<td>5259</td>
<td>8133</td>
</tr>
<tr>
<td>Fr</td>
<td>3640</td>
<td>0</td>
<td>8277</td>
<td>400</td>
<td>1253</td>
<td>9776</td>
</tr>
<tr>
<td>HK</td>
<td>10279</td>
<td>8277</td>
<td>0</td>
<td>8252</td>
<td>6063</td>
<td>1788</td>
</tr>
<tr>
<td>Lnd</td>
<td>3251</td>
<td>400</td>
<td>8252</td>
<td>0</td>
<td>1557</td>
<td>9536</td>
</tr>
<tr>
<td>Mos</td>
<td>5259</td>
<td>1253</td>
<td>6063</td>
<td>1557</td>
<td>0</td>
<td>4667</td>
</tr>
<tr>
<td>Tk</td>
<td>8133</td>
<td>9776</td>
<td>1788</td>
<td>9536</td>
<td>4667</td>
<td>0</td>
</tr>
</tbody>
</table>

Merge two clusters Fr and Lnd as $d(\text{Fr}, \text{Lnd})$ smallest
A worked example: Airline distances

Compute new $5 \times 5$ dissimilarity matrix

$$D(6) = \begin{array}{cccccc}
& \text{FrLnd} & \text{MntNY} & \text{HK} & \text{Mos} & \text{Tk} \\
\text{FrLnd} & 0 & 3251 & 8252 & 1253 & 9536 \\
\text{MntNY} & 3251 & 0 & 10279 & 5259 & 8133 \\
\text{HK} & 8252 & 10279 & 0 & 6063 & 1788 \\
\text{Mos} & 1253 & 5259 & 6063 & 0 & 4667 \\
\text{Tk} & 9536 & 8133 & 1788 & 4667 & 0
\end{array}$$

Merge two clusters FrLnd and Mos as $d(\text{FrLnd}, \text{Mos})$ smallest
A worked example: Airline distances

Compute new $4 \times 4$ dissimilarity matrix

$$D(4) = \begin{array}{cccc}
& \text{FrLndMo} & \text{MntNY} & \text{HK} & \text{Tk} \\
\text{FrLndMo} & 0 & 3251 & 6063 & 4667 \\
\text{MntNY} & 3251 & 0 & 10279 & 8133 \\
\text{HK} & 6063 & 10279 & 0 & 1788 \\
\text{Tk} & 4667 & 8133 & 1788 & 0 \\
\end{array}$$

Repeat until there is only one cluster.
Another example

- Randomly generated data
Another example: Dendrograms

- Three different linkage—single, average and complete
- Compare cluster assignments with three clusters
Cluster assignments by Single Linkage

- Tends to leave single points as clusters
- Suffers from *chaining* (clusters spread out, not compact)
Cluster assignments by Complete Linkage

- Can have a disjoint cluster
- Suffers from *crowding* (a point can be closer to points in other clusters than to points in its own cluster)
Average Linkage

- A good balance – relatively compact, relatively far apart
More Dissimilarities

For quantitative variables. For $\mathbf{x}, \mathbf{y} \in \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$,

- $p$-norm $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p = \left(\sum_{i=1}^{d} \|x_i - y_i\|^p\right)^{1/p}$, $p > 0$.

- Standardized distance $d^2(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} \frac{\|x_i - y_i\|^2}{s_i^2} = (\mathbf{x} - \mathbf{y})' \mathbf{D}^{-1} (\mathbf{x} - \mathbf{y})$, where $s_i$ is the standard deviation of $i$th measurements and $\mathbf{D}$ is the diagonal matrix consisting of diagonal elements of (sample) covariance matrix $\mathbf{S}$.

- Mahalanobis distance $d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})' \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})$.

- Many others...

Different distances lead different clustering, as seen in the next few slides.
Choice of Dissimilarities (Geyser)

Using squared 2-norm:

---

**k-means**

- Interval to next (min) vs. Duration of eruption (min)

**Hier.--single**

- Interval to next (min) vs. Duration of eruption (min)

**Hier.--complete**

- Interval to next (min) vs. Duration of eruption (min)

**Hier.--average**

- Interval to next (min) vs. Duration of eruption (min)
Choice of Dissimilarities (Geyser)

Using squared 2-norm (squared Euclidean distance)
Choice of Dissimilarities (Geyser)

Using Standardized distance:

- k-means
- Hier.--single
- Hier.--complete
- Hier.--average
Choice of Dissimilarities (Geyser)

Using Standardized distance – equivalent to using squared Euclidean distance for standardized variables
Choice of Dissimilarities (Geyser)

Using Mahalanobis distance:

- k-means
- Hier.--single
- Hier.--complete
- Hier.--average
Choice of Dissimilarities (Geyser)

Using Mahalanobis distance – equivalent to using squared Euclidean distance for sphered variables

![Graphs showing different clustering methods](image)
How many clusters?

What is the value of $K$?

Using K-means, K-medoids, or hierarchical clustering, attempts at formulating formal criteria to decide on the number of clusters have not been successful, by and large.

There are situations where the value of $K$ is pre-determined, e.g.,

- Theory suggests existence of $K$ clusters
- Segmenting a client database into $K$ clusters for $K$ salesman
Scatter decomposition

Focus on squared Euclidean distance.
Recall Within-cluster scatter

\[ W(c) = \sum_{k=1}^{K} \sum_{i: c(i)=k} \|x_i - \bar{x}(k)\|^2. \]

- smaller \( W \) is better (combinatorial method)
- WCS \( W \) keeps decreasing for larger \( K \) (No use in decision of right \( K \))

Consider ANOVA-like decomposition of total scatter

\[ \sum_i \|x_i - \bar{x}\|^2 = T = W(c) + B(c), \]

where \( B(c) = \sum_{k=1}^{K} n_k \| \bar{x}(k) - \bar{x} \|^2 \) is the between-cluster scatter.
Scatter decomposition

Between-cluster scatter

\[ B(c) = \sum_{k=1}^{K} n_k \| \bar{x}_{(k)} - \bar{x} \|^2. \]

- larger \( B \) is better (large gaps between clusters)
- BCS \( W \) keeps decreasing for larger \( K \) (No use in decision of right \( K \))
Cluster Index
Cluster index is the standardized with-cluster scatter

\[ CI(K, c) = \frac{W(K, c)}{T}. \]

- Unit free \( \in (0, 1) \)
- Still increasing for large \( K \)

CH Index

\[ CH(K, c) = \frac{B(K, c)/(K - 1)}{W(K, c)/(n - K)}. \]

- A large \( CH \) \iff a small \( W \) and a large \( B \)
- Can choose a \( K \) with largest CH index
- (Problem: no way choosing \( K = 1 \))
Scatter decomposition

One may look for an elbow (as we did when deciding the number of principal components) of a scree plot of $W(K)$ or $CI(K)$.

$W_1 \gg W_2$ since natural groups are assigned to separate clusters.

Smaller decrease $W_k$ to $W_{k+1}$ ($k \geq 2$), as natural groups are partitioned.
Gap statistic

- Measures how much $W(K)$ drops compared to a null case.
- The observed WCS $W(K)$ is compared with the expected WCS when there is only one cluster:

$$Gap(K) = E^* \left[ \log W_0(K) \right] - \log W(K)$$

- $E^* [\log W_0(K)]$ is simulated by a uniform one cluster distribution.
- The standard deviation $s(K)$ of $\log W_0(K)$ is also computed.
- We then choose the smallest $K$ such that

$$Gap(K) \geq Gap(K + 1) - s(K + 1).$$
Gap statistic: Toy example

- Data with *true* $K = 3$.
- Results from *K*-means algorithm.

![Graphs showing results for different values of k (1 to 6), with corresponding W values.](image-url)
Gap statistic: Toy example

- What would be the value of $W(k)$ if there is only one cluster?
- Reference sampled from uniform, then $W$(reference) computed from $K$-means algorithm.
Gap statistic: Toy example

- Compare the amount of decrease of $W(k)$ in 'data' and 'reference' (Gap between $\log(W(\text{ref.}))$ and $\log(W(\text{data}))$)
- Remember that this is just one example of reference distribution.
Gap statistic: Toy example

- Take the mean and standard deviation of many $\log(W_{\text{ref.}})$ to obtain $E^* [\log W_0(K)]$ and $s(K)$
- The smallest $K$ with $\text{Gap}(K) \geq \text{Gap}(K + 1) - s(K + 1)$ is 1?
- In cases where there are smaller subclusters within larger well-separated clusters, Gap curve can exhibit non-monotone behavior; *Important to examine the entire gap curve*
Gap statistic– 2-cluster example

\( (K = 1:) \) \( \text{Gap}(1) < \text{Gap}(2) - s(2) \), move on to \( K = 2 \)

\( (K = 2:) \) \( \text{Gap}(2) > \text{Gap}(3) - s(3) \). \( \hat{K} = 2 \).
Gap statistic—1-cluster example

\( (K = 1:) \text{ Gap}(1) > \text{ Gap}(2) - s(2), \hat{K} = 1. \)
Model-based clustering

Consider a mixture model density in $\mathbb{R}^p$

$$g(x) = \sum_{k=1}^{K} \pi_k g_k(x),$$

where $g_k$ is the pdf of $N_p(\mu_k, \sigma^2 I_p)$, $\pi_k > 0$, $\sum_k \pi_k = 1$.

- For data $x_1, \ldots, x_n \sim g$, clustering the data $\sim$ estimation of parameters
- EM algorithm is a soft version of $K$-means algorithm
Clustering variables

So far we have focused on clustering subjects (or individuals). Variables (Measurements) can also be grouped into several clusters. We only need to have dissimilarity between variables. Common choices are:

- 1-correlation:
  \[ d(V_i, V_j) = 1 - \rho_{ij} \]
  where \( \rho_{ij} \) is the correlation coefficient between r.vs \( V_i \) and \( V_j \).

- 1-squared-correlation:
  \[ d(V_i, V_j) = 1 - \rho_{ij}^2. \]

One can then use the dissimilarity-based clustering algorithms with dissimilarity matrix \( D_{(p \times p)} = (d_{ij}), \; d_{ij} = d(V_i, V_j) \).
kmeans and agglomerative hierarchical algorithms are part of base R distribution. For analysis of gap statistics, use clusGap in cluster package

```r
k <- 3
kmeansobj<-kmeans(iris[1:4],k)

d = dist(iris[1:4])
tree.avg = hclust(d, method="average")
plot(tree.avg)
membership <- cutree(tree.avg, k = 3)

library(cluster)
gap <- clusGap(iris[1:4], FUN = kmeans, K.max = 8)
plot(gap)
```
kmeans and agglomerative hierarchical algorithms are part of Statistics Toolbox. For gap statistics and other methods of evaluating number of clusters, use the very recent version (R2013b).