Clustering

Finding internal structure in the data

Learning goals

1. Learn the idea and applicability of clustering.
2. Understand the idea of selected clustering algorithms:
   - K-means
   - Agglomerative hierarchical clustering
3. Learn to conduct cluster analysis in Python.

Cluster analysis

”People who love running and frequently read political thrillers form a distinct customer group”

- Clustering (or cluster analysis) is one of the most straighforward unsupervised machine learning methods.
- It is applicable to numeric variables at interval scale.
  - i.e. it must be possible to measure distances between the values.
- It finds internal structure in the data that is often unobservable by naked eye.

The idea of cluster analysis

- Cluster analysis, or, clustering means automatical grouping of observations based on values of multiple variables.
- Similar observations end up in the same cluster.
- The figure shows the formation of three clusters in two-dimensional space.
- In the figure, there are two variables corresponding to the *x* and *y* coordinates.
- If there are *n* variables, clusters are formed in *n*-dimensional space.
  - Impossible to visualize by humans if n > 3.

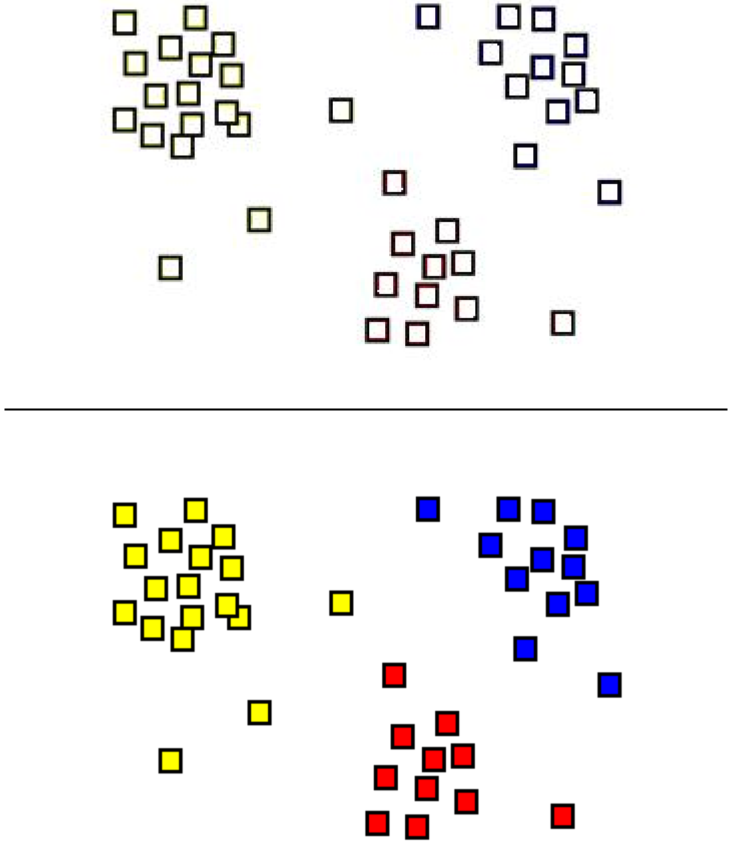


Image: Wikimedia Commons. Public domain.

The goal of cluster analysis

- The goal is to find a representative set of typical, yet different, groups of observations.
- Suitable for generating stereotypes and profiling.
- In the next few slides, we focus on two clustering techniques
  - k-means
  - agglomerative hierarchical clustering

Requirements for the data

- Cluster analysis is based on measuring distances.
  - Usually ”regular” Euclidean distance.
- Distance can be measured only if the variables are of at least interval scale.
  - E.g. income in euros, human height and weight.
  - Some ordinal scale variables can be usable, e.g. school grades.
    - The variables need to be ”equidistant enough”.
  - Also, binary nominal variables can be used.
    - Can be recoded as 0 and 1, and, subsequently, standardized.
- The mean and variance of the variable values should ideally be uniform.
  - Otherwise, the variables with large variance dominate cluster generation.
  - To achieve this, the variables usually need to be standardized as a preprocessing step.

Standard score

$$z(M_i) = \frac{M_i - \mu(M)}{\sigma(M)}$$

- Calculating the standard score – or z-score – is a common way of shifting and rescaling variable values.
- A standardized variable value is obtained by substracting the mean from each observation, after which the result is divided by standard deviation.
  - Standard deviation is the average of squared differences from the mean.
- The mean and standard deviation can easily be computed from the sample.
- The resulting variable has a mean of zero and a variance of one.
- As a consequence of standardization, the variables are ”treated equally” in the analysis.

Phases of k-means algorithm

1. Decide the number of clusters k.
2. Standardize the observations if necessary.
3. Select k cluster centerpoints – called as centroids.
   - either randomly, or
   - by selecting random observations.
4. Compute the distance between each observation and each centroid.l
5. Assign observation to the cluster whose centroid is closest to the observation.
   - This divides the observation space into so-called Voronoi cells (see next slide).
6. The location of each centroid is computed again from the observation assigned to that cluster.
   - Done by taking the mean of each coordinate.
7. Repeat from step 4 unless the centroids have stayed same.

Voronoi cells in the k-means algorithm

- In the Step 5 of the previous slide, each observation was assigned to the cluster whose centroid is closest to the observation.
- As a consequence, the observation space is split into Voronoi cells (see figure). The dots represent centroids.
- The areas of similar color belong to the same cluster.
- All observations within same cluster are closest to the same centroid.

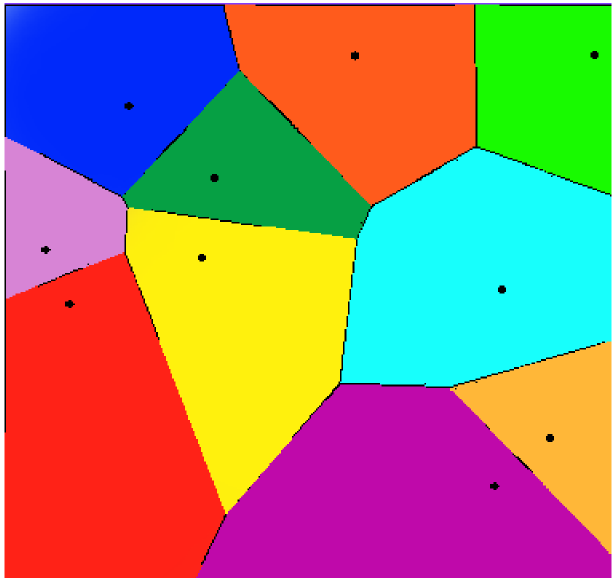
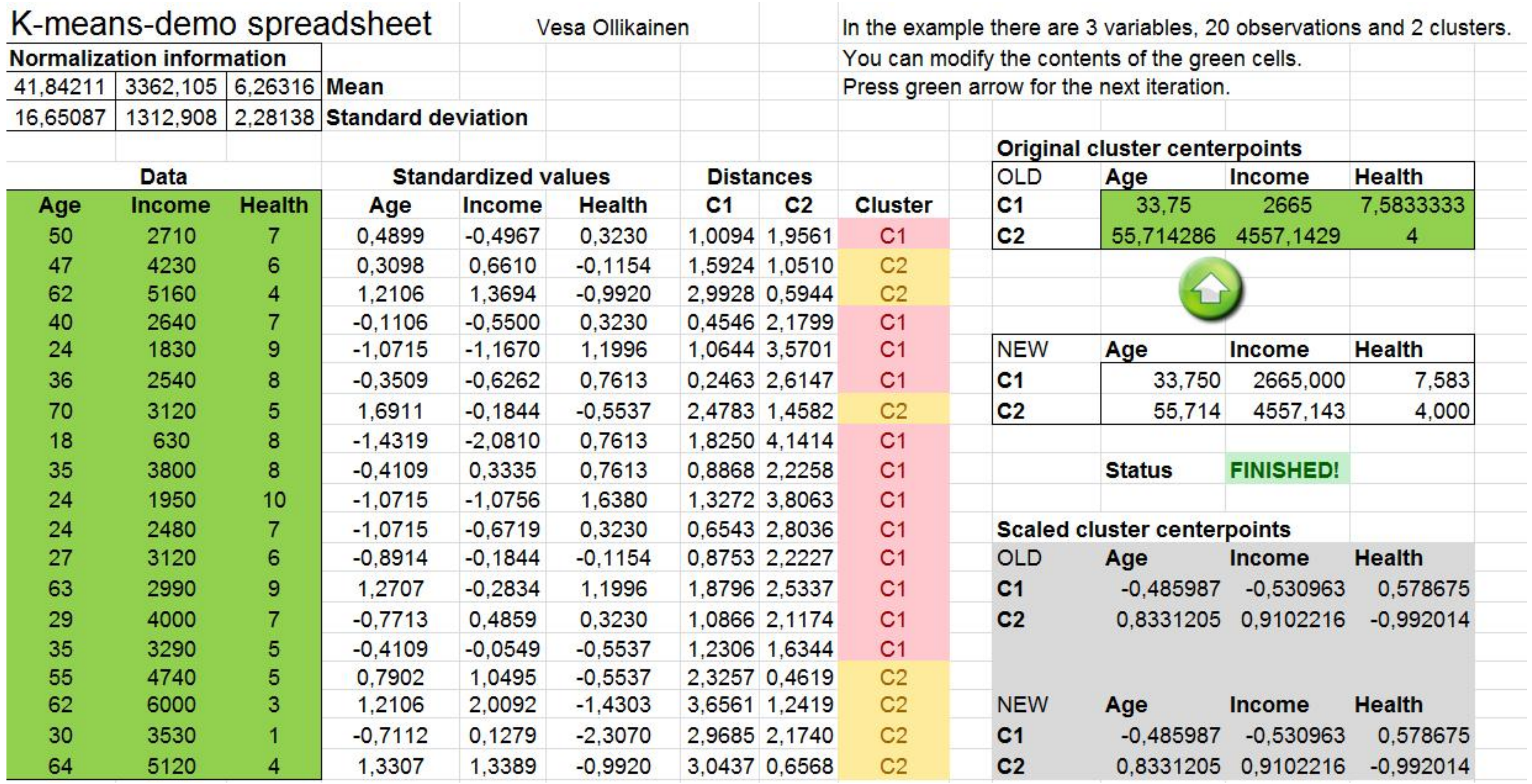


Image: Wikimedia Commons. Public domain.

Demo of k-means algorithm

- In the Oma workspace there is an demo spreadsheet **clustering_demo.xlsm** that simulates the k-means algorithm for k =2 (two clusters) in three-dimensional space (three variables).
  - First, set the cluster centerpoints (centroids).
  - You can also play with data values.
  - By pressing the green arrow, you proceed to the next iteration.
  - You have to allow macros to be able to run the demo.

Demo of k-means algorithm



K-means-demo spreadsheet

Vesa Ollikainen

In the example there are 3 variables, 20 observations and 2 clusters.
You can modify the contents of the green cells.
Press green arrow for the next iteration.

Normalization information

41,84211	3362,105	6,26316	**Mean**
16,65087	1312,908	2,28138	**Standard deviation**

Data			**Standardized values**			**Distances**		
Age	**Income**	**Health**	**Age**	**Income**	**Health**	**C1**	**C2**	**Cluster**
50	2710	7	0,4899	-0,4967	0,3230	1,0094	1,9561	C1
47	4230	6	0,3098	0,6610	-0,1154	1,5924	1,0510	C2
62	5160	4	1,2106	1,3694	-0,9920	2,9928	0,5944	C2
40	2640	7	-0,1106	-0,5500	0,3230	0,4546	2,1799	C1
24	1830	9	-1,0715	-1,1670	1,1996	1,0644	3,5701	C1
36	2540	8	-0,3509	-0,6262	0,7613	0,2463	2,6147	C1
70	3120	5	1,6911	-0,1844	-0,5537	2,4783	1,4582	C2
18	630	8	-1,4319	-2,0810	0,7613	1,8250	4,1414	C1
35	3800	8	-0,4109	0,3335	0,7613	0,8868	2,2258	C1
24	1950	10	-1,0715	-1,0756	1,6380	1,3272	3,8063	C1
24	2480	7	-1,0715	-0,6719	0,3230	0,6543	2,8036	C1
27	3120	6	-0,8914	-0,1844	-0,1154	0,8753	2,2227	C1
63	2990	9	1,2707	-0,2834	1,1996	1,8796	2,5337	C1
29	4000	7	-0,7713	0,4859	0,3230	1,0866	2,1174	C1
35	3290	5	-0,4109	-0,0549	-0,5537	1,2306	1,6344	C1
55	4740	5	0,7902	1,0495	-0,5537	2,3257	0,4619	C2
62	6000	3	1,2106	2,0092	-1,4303	3,6561	1,2419	C2
30	3530	1	-0,7112	0,1279	-2,3070	2,9685	2,1740	C2
64	5120	4	1,3307	1,3389	-0,9920	3,0437	0,6568	C2

Original cluster centerpoints

OLD	**Age**	**Income**	**Health**
C1	33,75	2665	7,5833333
C2	55,714286	4557,1429	4

⇧

NEW	**Age**	**Income**	**Health**
C1	33,750	2665,000	7,583
C2	55,714	4557,143	4,000

Status FINISHED!

Scaled cluster centerpoints

OLD	**Age**	**Income**	**Health**
C1	-0,485987	-0,530963	0,578675
C2	0,8331205	0,9102216	-0,992014

NEW	**Age**	**Income**	**Health**
C1	-0,485987	-0,530963	0,578675
C2	0,8331205	0,9102216	-0,992014

On k-means algorithm

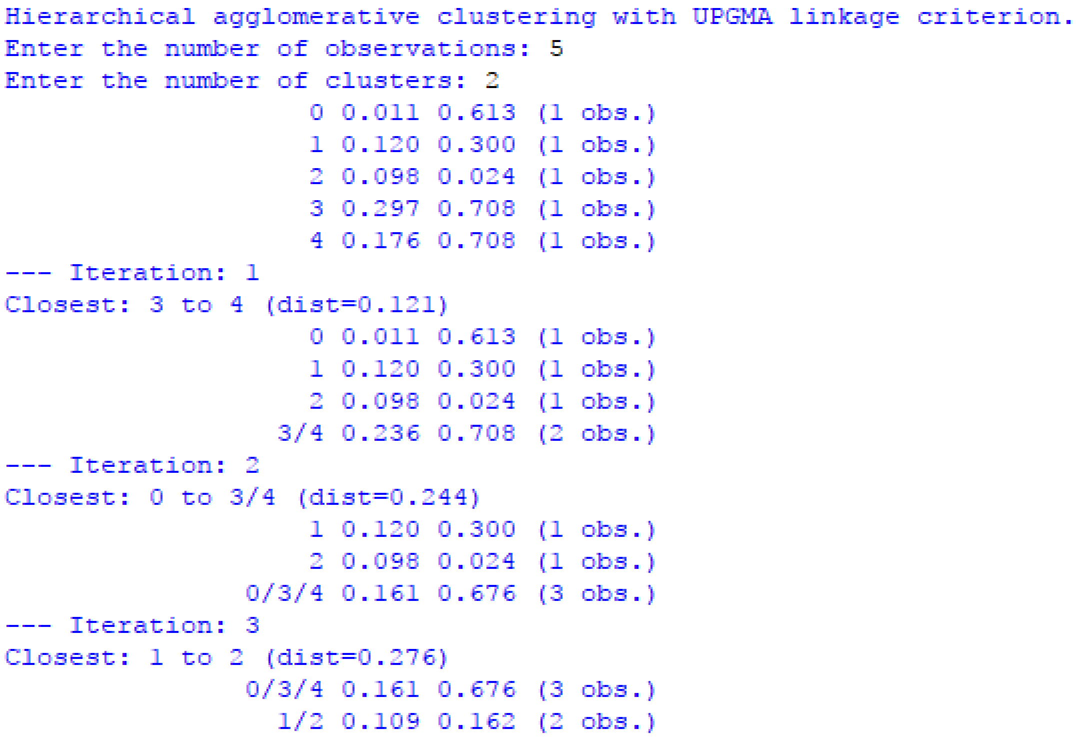
- The algorithm is iterative.
  - The cluster division gets ’better’ in each iteration.
  - Finally, the algorithm converges into a result that won’t change.
- The result is not necessarily an optimal cluster division.
  - The cluster formation problem is NP complete. There is no general solution that is guarenteed to find the best division (unless computation time is allowed to grow very fast).

Hierarchichal clustering

- In hierarchical clustering, the number of clusters changes dynamically as the algorithm proceeds.
    1. Agglomerative clustering
        - All data points are in the separate clusters. The clusters with smallest distance are repeatedly merged.
    2. Divisive clustering
        - First, all observations are in the same cluster. In each step, a non-hierarchical clustering method is applied *within clusters*. As a consequence, the number of clusters grows.

Agglomerative hierarchical clustering example

- In the example, there at 5 random data point.
- In each step, two closest clusters are combined.
- The new cluster centerpoint is the average of all points in the merged clusters.



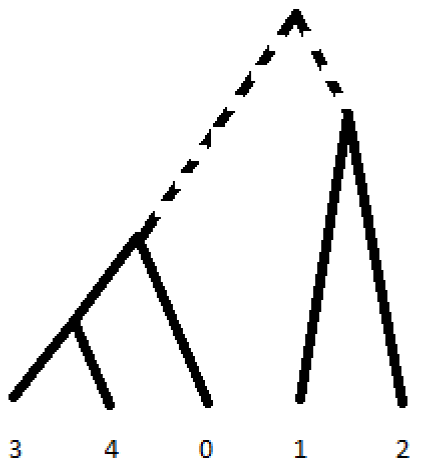
```
Hierarchical agglomerative clustering with UPGMA linkage criterion.
Enter the number of observations: 5
Enter the number of clusters: 2
                   0 0.011 0.613 (1 obs.)
                   1 0.120 0.300 (1 obs.)
                   2 0.098 0.024 (1 obs.)
                   3 0.297 0.708 (1 obs.)
                   4 0.176 0.708 (1 obs.)
--- Iteration: 1
Closest: 3 to 4 (dist=0.121)
                   0 0.011 0.613 (1 obs.)
                   1 0.120 0.300 (1 obs.)
                   2 0.098 0.024 (1 obs.)
                 3/4 0.236 0.708 (2 obs.)
--- Iteration: 2
Closest: 0 to 3/4 (dist=0.244)
                   1 0.120 0.300 (1 obs.)
                   2 0.098 0.024 (1 obs.)
               0/3/4 0.161 0.676 (3 obs.)
--- Iteration: 3
Closest: 1 to 2 (dist=0.276)
               0/3/4 0.161 0.676 (3 obs.)
                 1/2 0.109 0.162 (2 obs.)
```

Computation of inter-cluster distances

- The linkage criterion dictates how the distances between the clusters are computed:
  - Minimum: distance between the closest observations.
  - Maximum: distance between the most distant observations
  - Average (UPGMA): average distance between observations
  - Centroid (UPGMC): distance between cluster centerpoints

Dendrogram

- A dendrogram can be drawn from the process of agglomerative clustering.
- It visualizes the distances between pairs of observations.



Other clustering approaches

- Distribution-based clustering
  - In the beginning, form one cluster. In each step, the distance of each observation from the existing clusters is computed. If it is below a given threshold, assign the observation to that cluster. Otherwise, form a new cluster.
- Density-based clustering
  - Define clusters as dense areas of observations surrounded by sparse regions of observations.

Clustering in Python

- **scikit-learn** package contains implementations of clustering algorithms.
- For an example of a k-means clustering task, see the contents of **Appreciation (demo)** notebook.