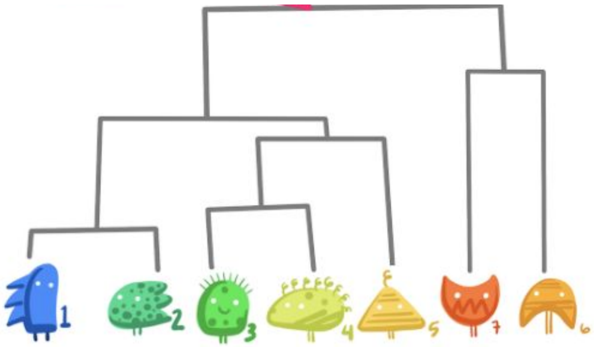
Hierarchical Clustering

Prof. Dr. Jan Kirenz
HdM Stuttgart

Hierarchical Clustering

- Builds **nested clusters** by merging or splitting them successively.
- This hierarchy of clusters is represented as a tree (or **dendrogram**)

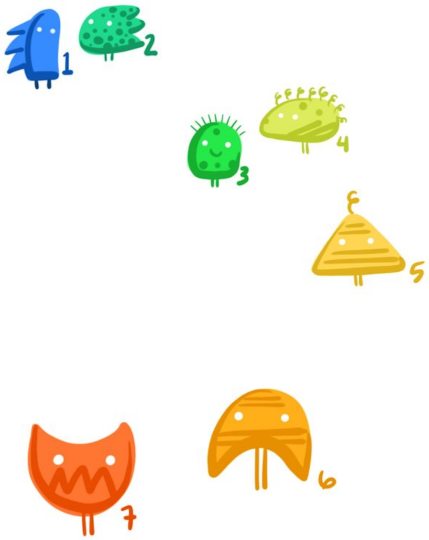


Simple example of Hierarchical clustering

hierarchical clustering: single linkage

(Step-by-step: combine clusters with the smallest distance between elements)

elements



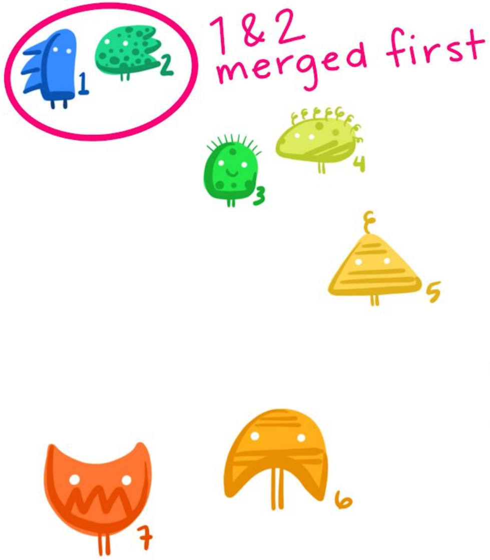
DISTANCE MATRIX

	1	2	3	4	5	6	7
1	0	10	30	40	60	85	82
2	10	0	24	38	55	87	90
3	30	24	0	16	26	50	63
4	40	38	16	0	21	52	67
5	60	55	26	21	0	41	58
6	85	87	50	52	41	0	32
7	82	90	63	67	58	32	0

Treat each element as a cluster

- Find smallest distance between elements in 2 clusters
- Those clusters get merged.

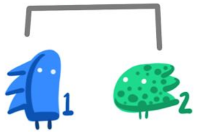
elements



DISTANCE MATRIX

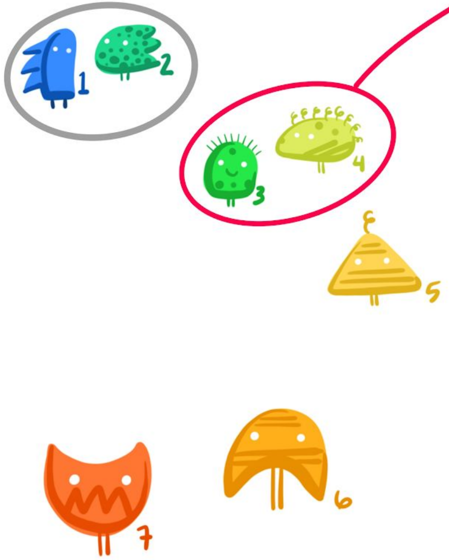
	1	2	3	4	5	6	7
1	0	10	30	40	60	85	82
2	10	0	24	38	55	87	90
3	30	24	0	16	26	50	63
4	40	38	16	0	21	52	67
5	60	55	26	21	0	41	58
6	85	87	50	52	41	0	32
7	82	90	63	67	58	32	0

build the DENDROGRAM



Now 1 & 2 are a single cluster.
Find the 2 clusters with smallest distance between elements, then merge them.

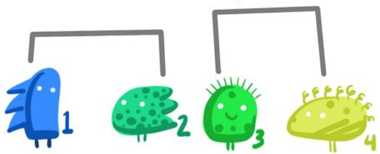
elements



DISTANCE MATRIX

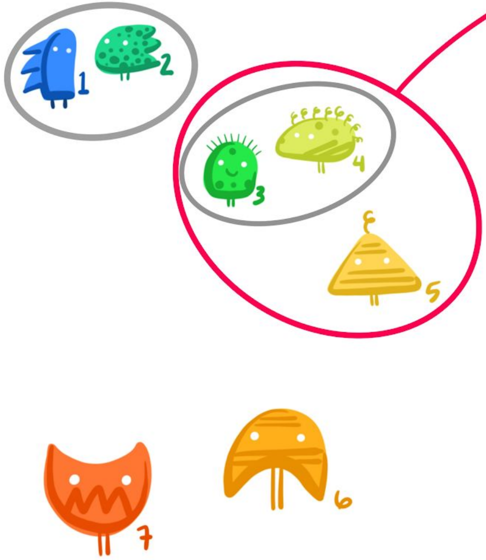
	1	2	3	4	5	6	7
1	0	~~10~~	30	40	60	85	82
2	~~10~~	0	24	38	55	87	90
3	30	24	0	16	26	50	63
4	40	38	16	0	21	52	67
5	60	55	26	21	0	41	58
6	85	87	50	52	41	0	32
7	82	90	63	67	58	32	0

build the DENDROGRAM



Repeat! Now the 2 clusters with the smallest distance between elements are the (3,4) and 5 clusters, so we merge them!

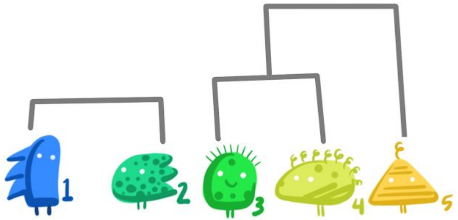
elements



DISTANCE MATRIX

	1	2	3	4	5	6	7
1	0	~~10~~	30	40	60	85	82
2	~~10~~	0	24	38	55	87	90
3	30	24	0	~~16~~	26	50	63
4	40	38	~~16~~	0	21	52	67
5	60	55	26	21	0	41	58
6	85	87	50	52	41	0	32
7	82	90	63	67	58	32	0

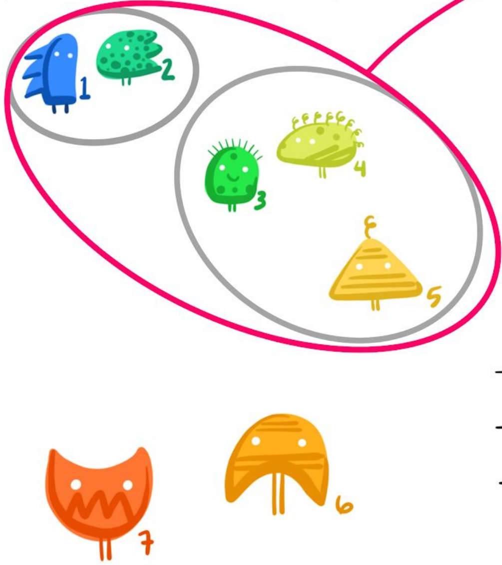
build the DENDROGRAM



Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

Yep, do it again! Now, the smallest distance between elements in two clusters is between 2 & 3, so we merge the clusters they're in!

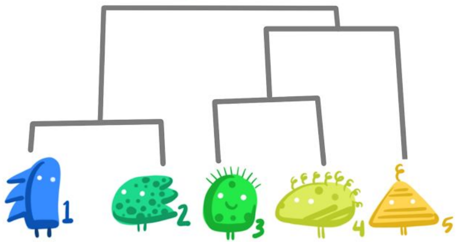
elements



DISTANCE MATRIX

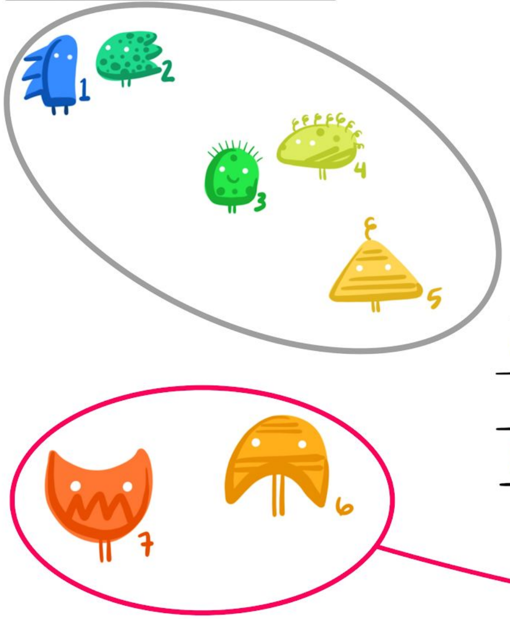
	1	2	3	4	5	6	7
1	0	~~10~~	30	40	60	85	82
2	~~10~~	0	24	38	55	87	90
3	30	24	0	~~16~~	26	50	63
4	40	38	~~16~~	0	~~21~~	52	67
5	60	55	26	~~21~~	0	41	58
6	85	87	50	52	41	0	32
7	82	90	63	67	58	32	0

build the DENDROGRAM



The next smallest distance between elements in separate clusters is between 6 & 7, so we merge them...

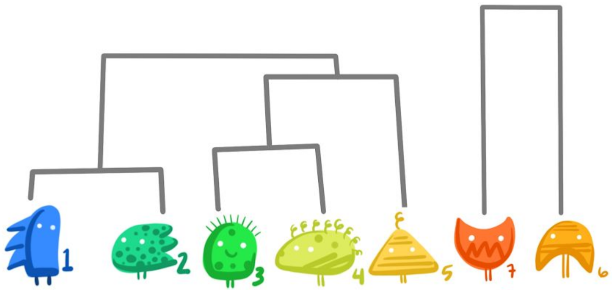
elements



DISTANCE MATRIX

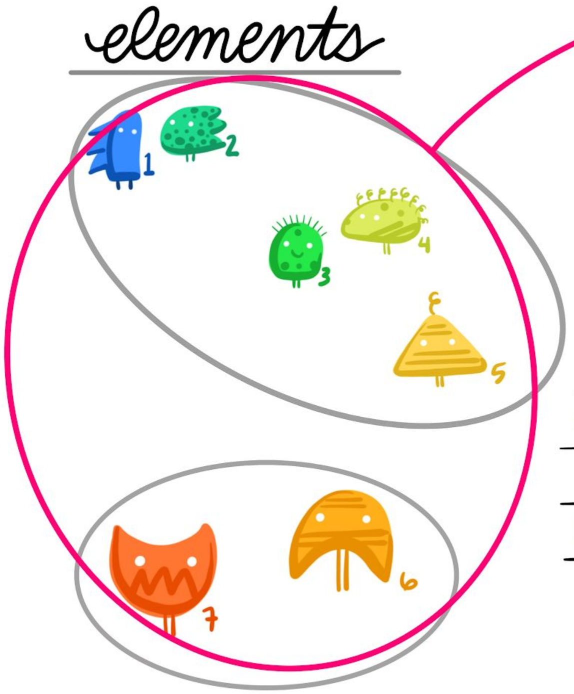
	1	2	3	4	5	6	7
1	0	~~10~~	~~30~~	~~40~~	~~60~~	85	82
2	~~10~~	0	~~24~~	~~38~~	~~55~~	87	90
3	~~30~~	~~24~~	0	~~16~~	~~26~~	50	63
4	~~40~~	38	~~16~~	0	~~21~~	52	67
5	~~60~~	~~55~~	~~26~~	~~21~~	0	41	58
6	85	87	50	52	41	0	32
7	82	90	63	67	58	32	0

build the DENDROGRAM



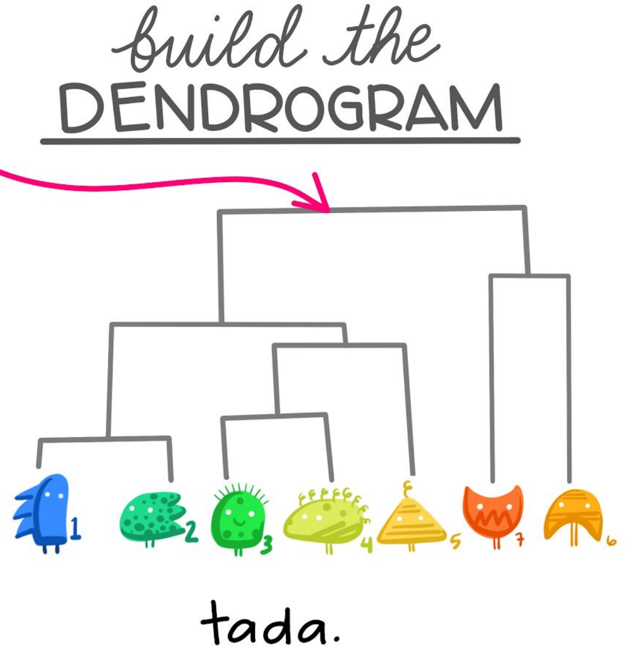
Source: Allison Horst https://github.com/allisonhorst/stats-illustrations/tree/master/other-stats-artwork

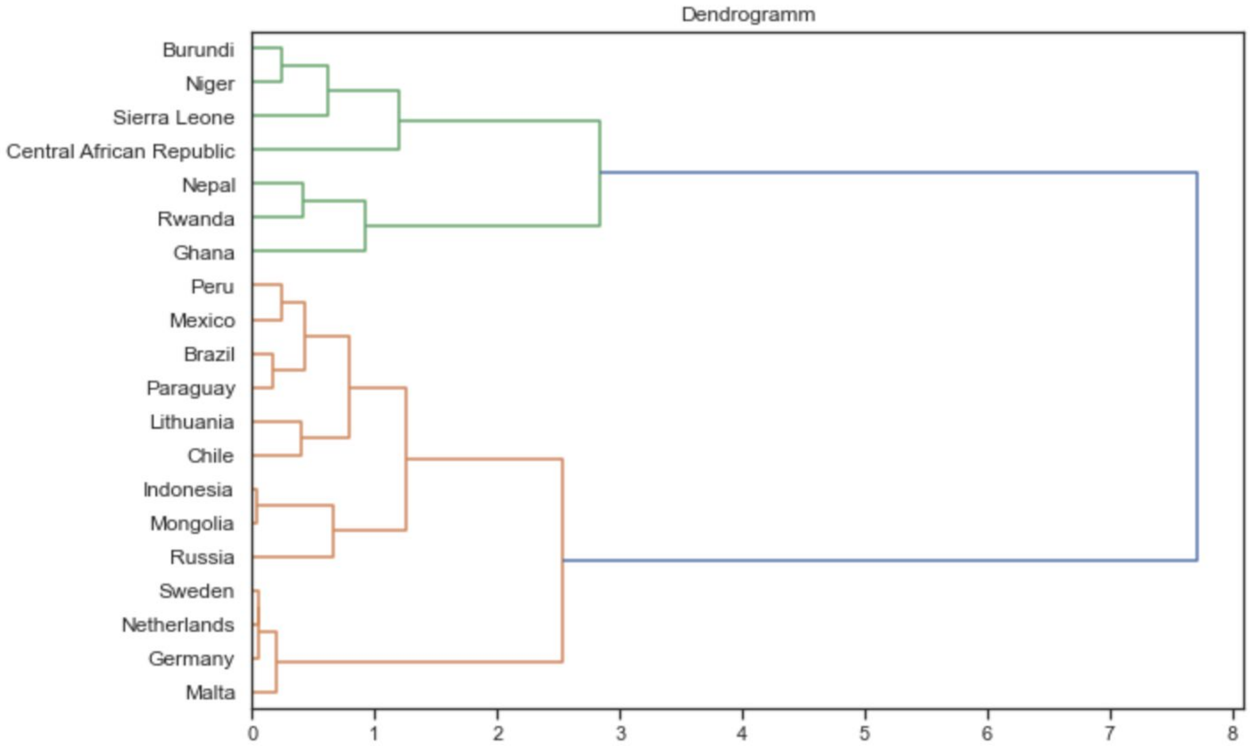
Now we only have two clusters, so they get merged!



DISTANCE MATRIX

	1	2	3	4	5	6	7
1	0	~~10~~	~~30~~	~~40~~	~~60~~	85	82
2	~~10~~	0	~~24~~	~~38~~	~~55~~	87	90
3	~~30~~	~~24~~	0	~~16~~	~~26~~	50	63
4	~~40~~	38	~~16~~	0	~~21~~	52	67
5	~~60~~	~~55~~	~~26~~	~~21~~	0	41	58
6	85	87	50	52	41	0	~~32~~
7	82	90	63	67	58	~~32~~	0




Dendrogramm
Burundi
Niger
Sierra Leone
Central African Republic
Nepal
Rwanda
Ghana
Peru
Mexico
Brazil
Paraguay
Lithuania
Chile
Indonesia
Mongolia
Russia
Sweden
Netherlands
Germany
Malta
0
1
2
3
4
5
6
7
8

Strategies for hierarchical clustering generally fall into two types

Agglomerative:

- This is a "bottom-up" approach
- Each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- This is the standard procedure

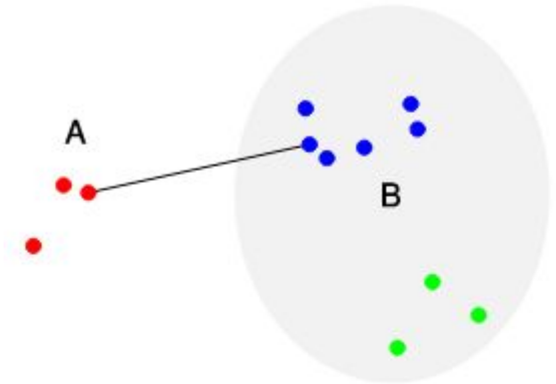
Divisive:

- This is a "top-down" approach
- All observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

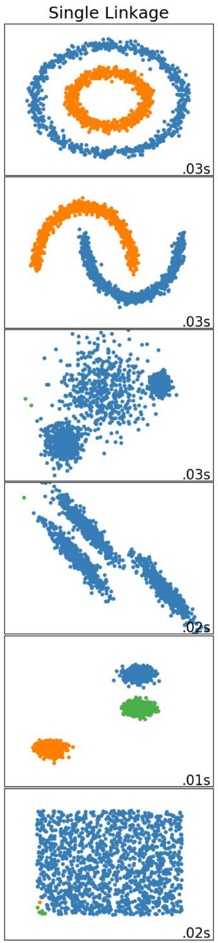
Different linkage types

Single linkage

- **Minimizes** the distance between the **closest observations** of pairs of clusters.
- Is very fast.
- Can perform well on non-globular data
- Performs poorly in the presence of noise.



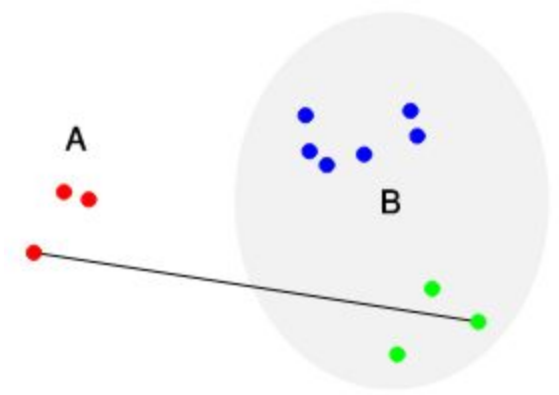
Source: Sigbert, 2011



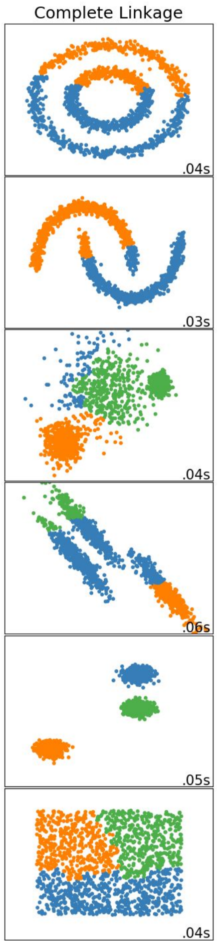
Source: Scikit-learn documentation https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html

Complete linkage (maximum linkage)

- **Minimizes** the **maximum distance** between observations of pairs of clusters.
- Performs well on cleanly separated globular clusters but has mixed results otherwise.

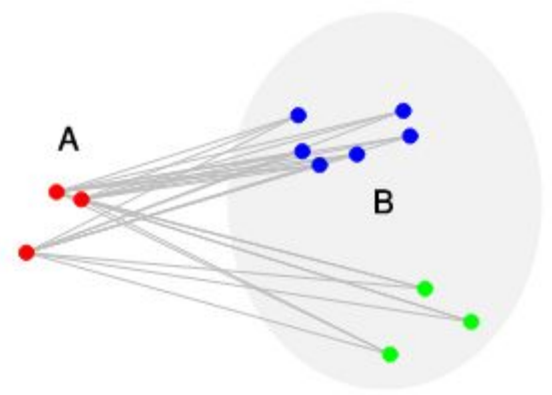


Source: Sigbert, 2011

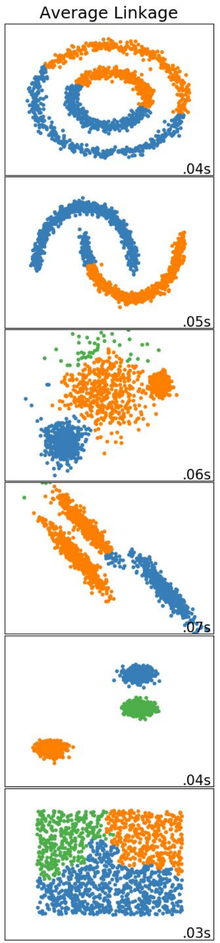


Average linkage

- **Minimizes** the **average** of the **distances** between all observations of pairs of clusters.
- Performs well on cleanly separated globular clusters but has mixed results otherwise.



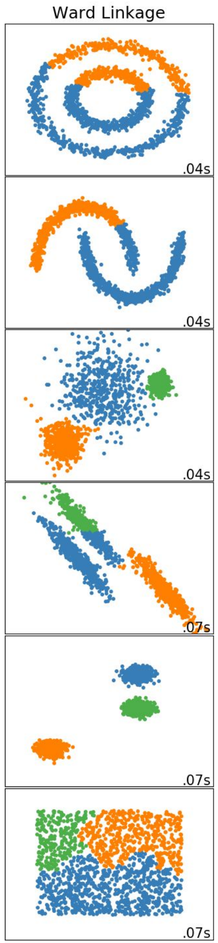
Source: Sigbert, 2011



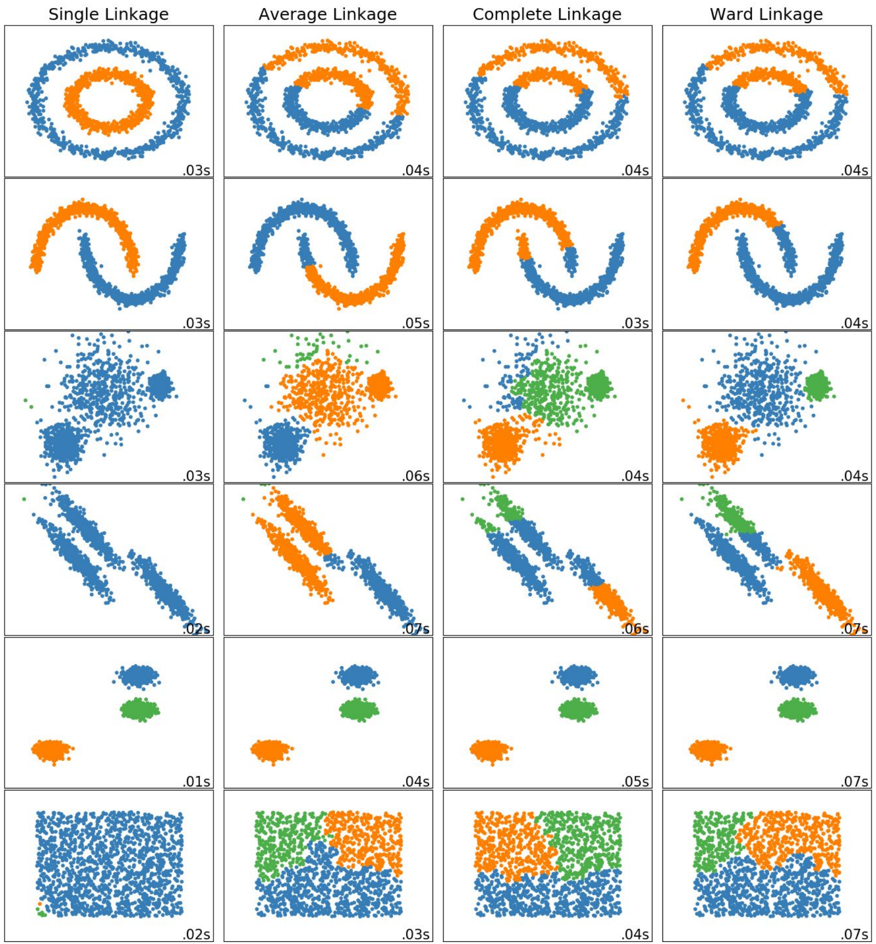
Source: Scikit-learn documentation https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html

Ward method linkage

- **Minimizes** the **sum** of **squared differences** within all clusters.
- Ward is the most effective method for noisy data.
- It is a variance-minimizing approach and in this sense is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.



Source: Scikit-learn documentation https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html



Source: https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html