

Grundpraktikum Bioinfo - Week 1

Biological Data Analysis

Carl Herrmann
IPMB - Universität Heidelberg



Institut für Pharmazie und
Molekulare Biotechnologie

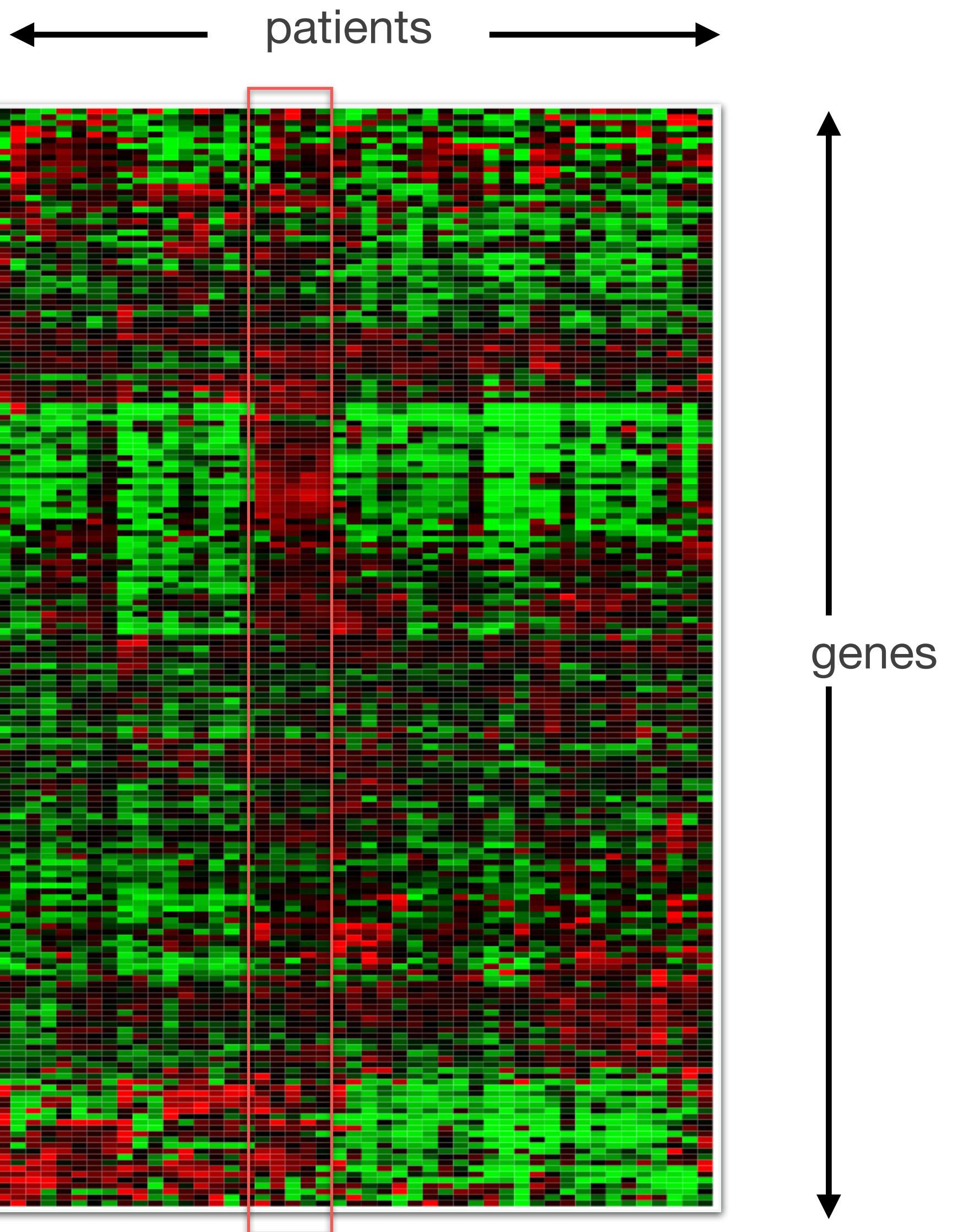


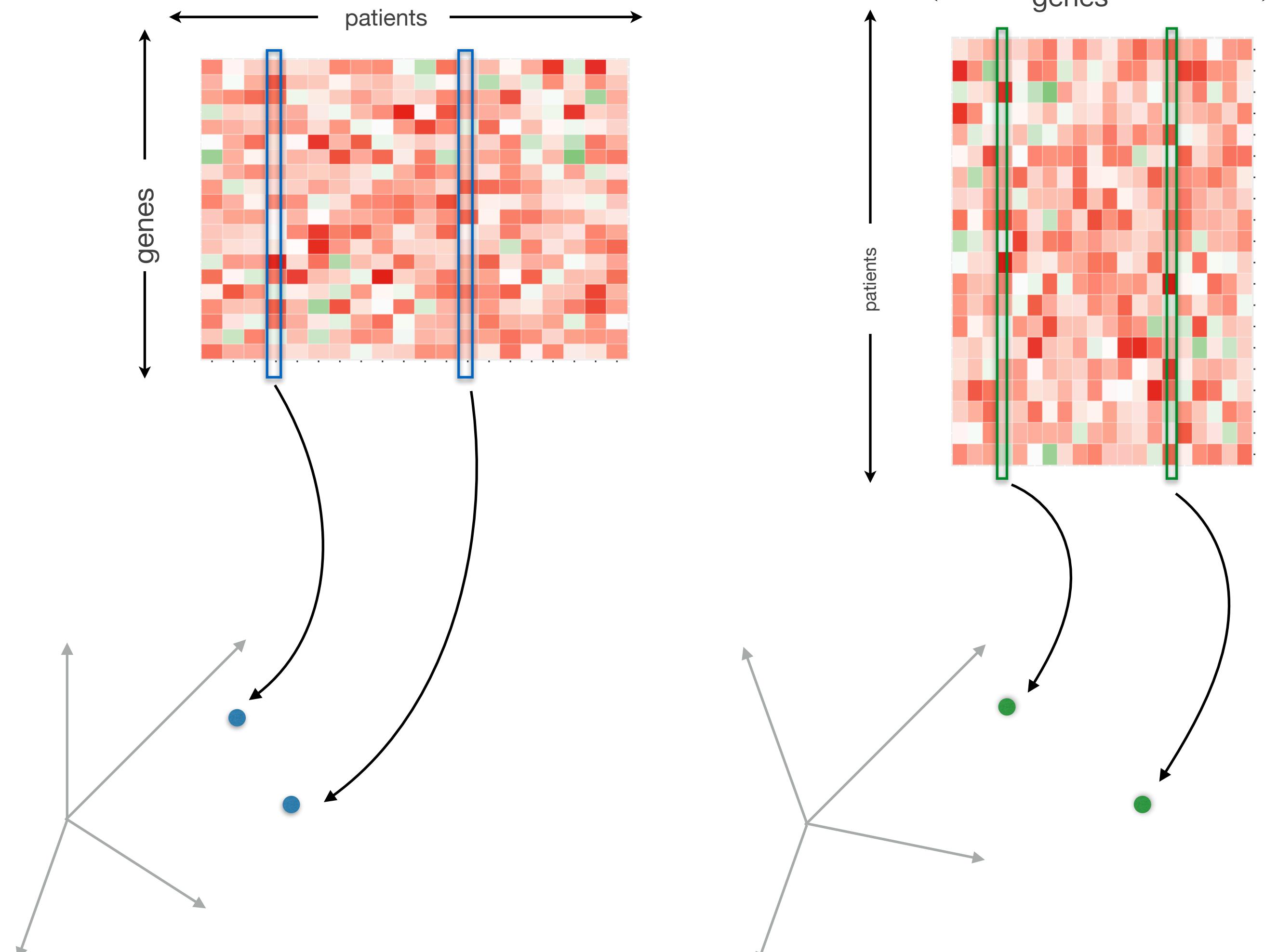
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

4. finding structure in the data introduction to clustering

Finding structure in the data

- **Goal:**
use a dataset to identify groups of samples which seem to be similar
- **Example**
 - gene expression data
 - clinical data
 - ...
- Important for molecular subtyping



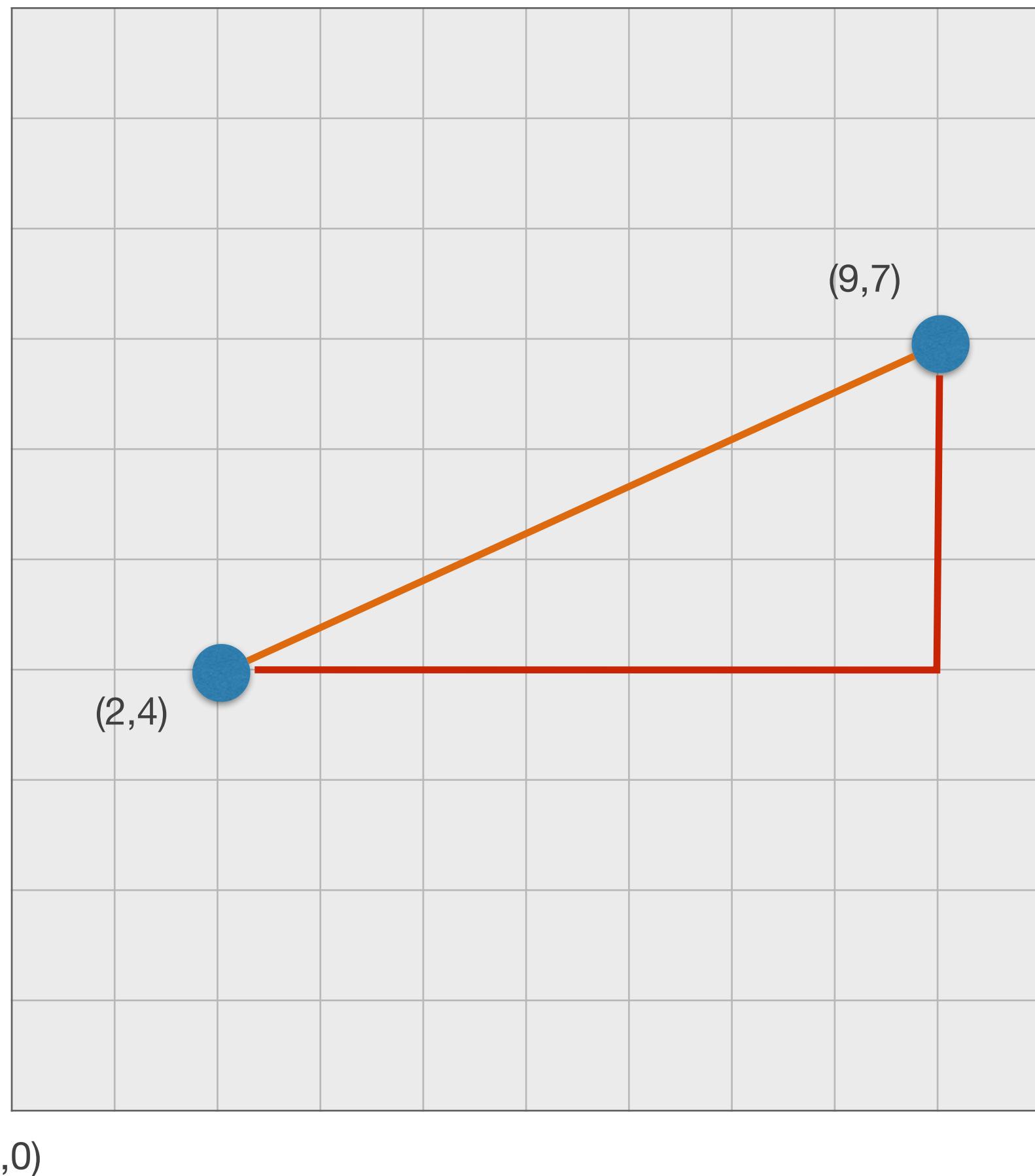


Points = **patients** in
20,000 dimensional
gene space

Points = **genes** in
500 dimensional
patient space

Distances / dissimilarities

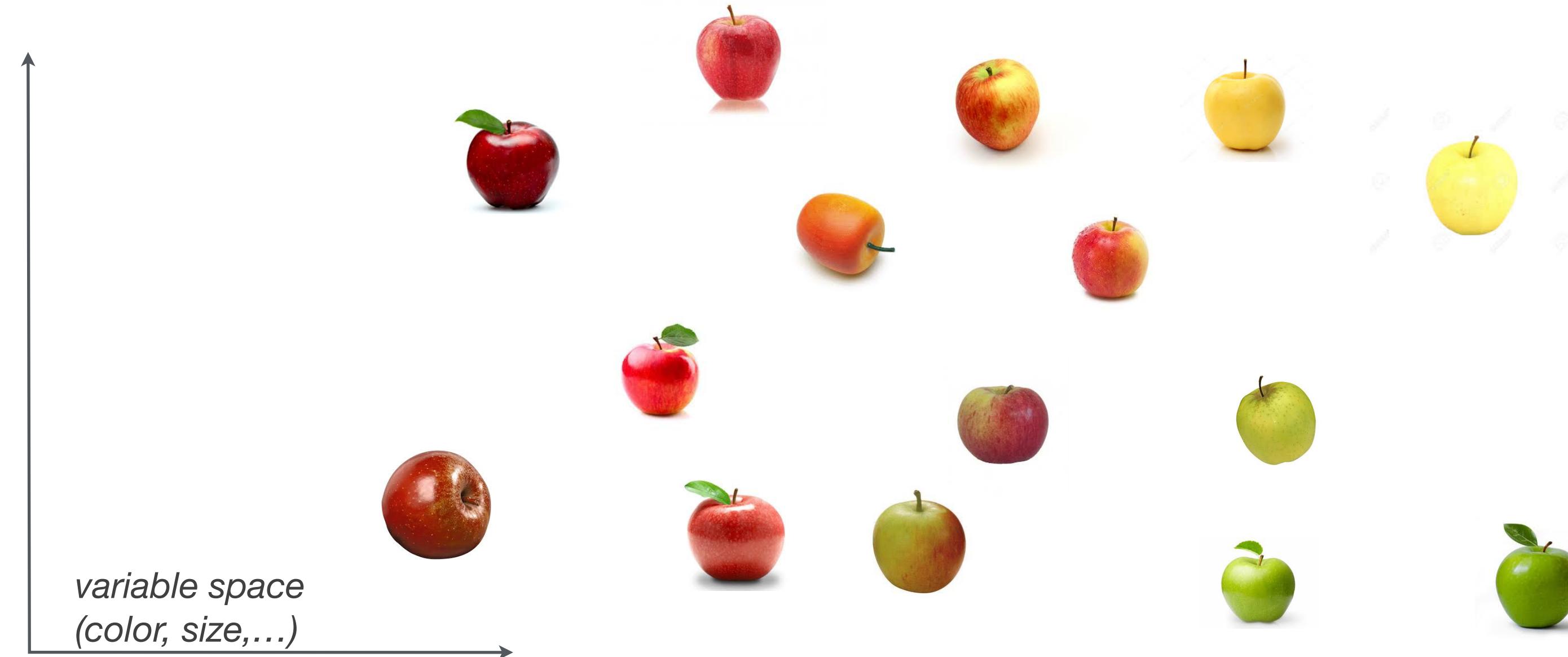
- Distances should be
 - positively defined : $d(a,b) \geq 0$
 - $d(a,a) = 0$
 - $d(a,c) \leq d(a,b) + d(b,c)$
- Several distances can be defined in n-dimensional space
 - **Euclidean distance**
 $d^2 = (9-2)^2 + (7-4)^2 = 58$
 - **Manhattan distance**
 $d = 7 + 3 = 10$



Can be generalized to any number of dimensions!

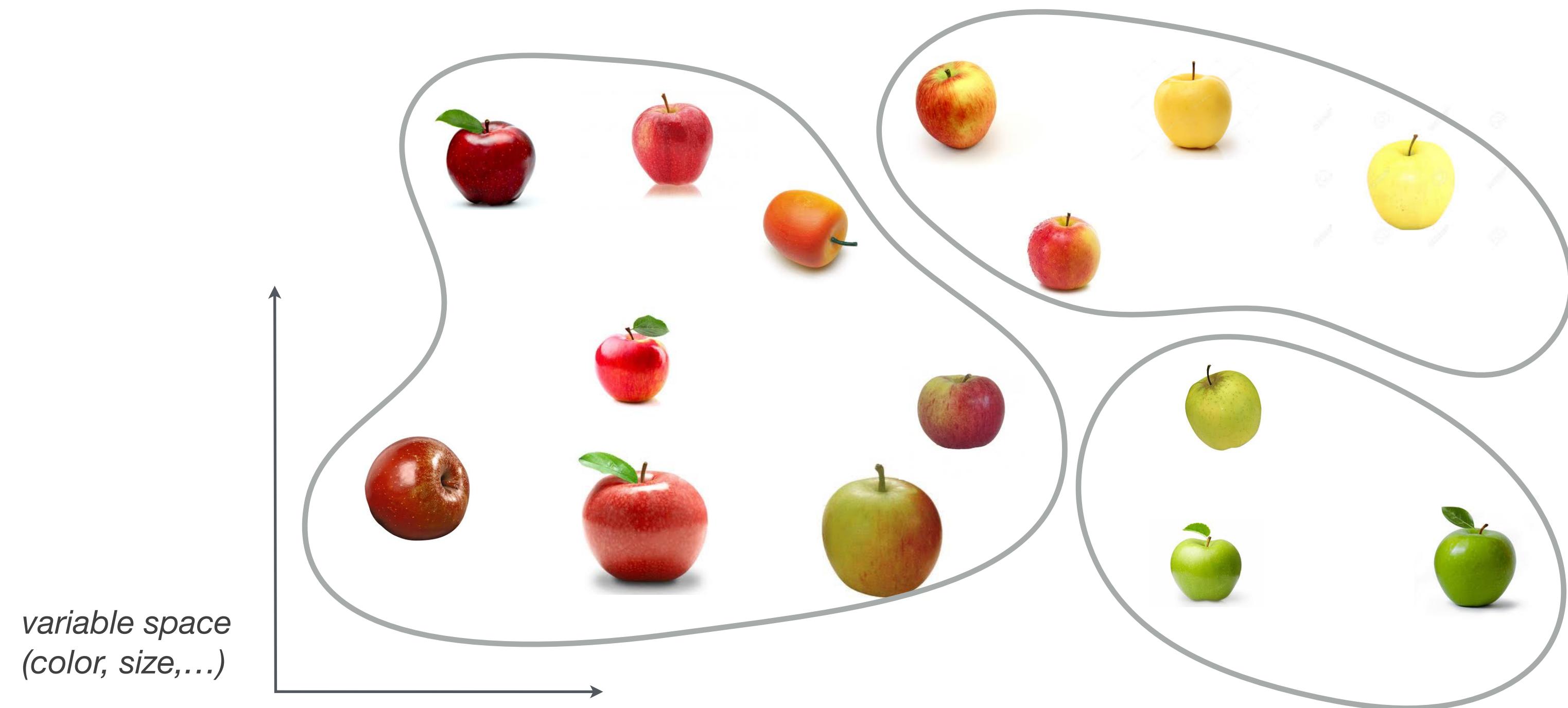
Clustering data

- Idea: group objects (= samples) which are **close to each other** in the variable space
- this implies the definition of a **distance between objects**
- **unsupervised approach:** we do **NOT** know to which group which sample belongs!



Clustering data

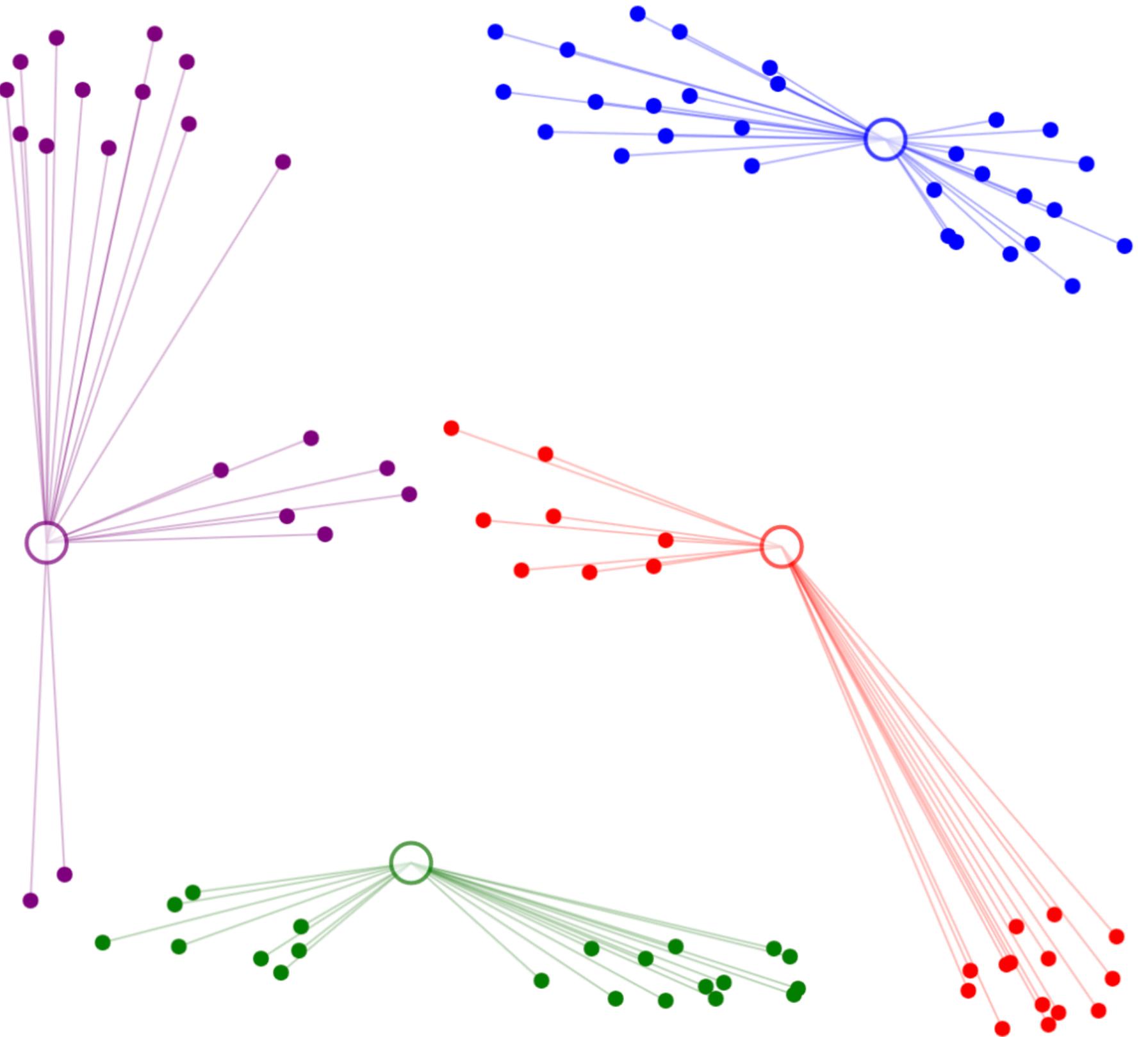
- Partition samples into k-groups
- Issues
 - how to define a **distance**?
 - optimal **number k of groups**?
 - how to deal with **ambiguous samples**?



4. finding structure in the data k-means clustering

k-means clustering

- **Objects** in a d-dimensional space
 - points in a 2d euclidean space
 - patients in a 20000d space of gene expression, ...
- Distance between object is the **euclidean metric**
- Define an **expected number k of clusters**
 - try out different numbers k
 - use external knowledge
- [Click here to see a demo]



k-means



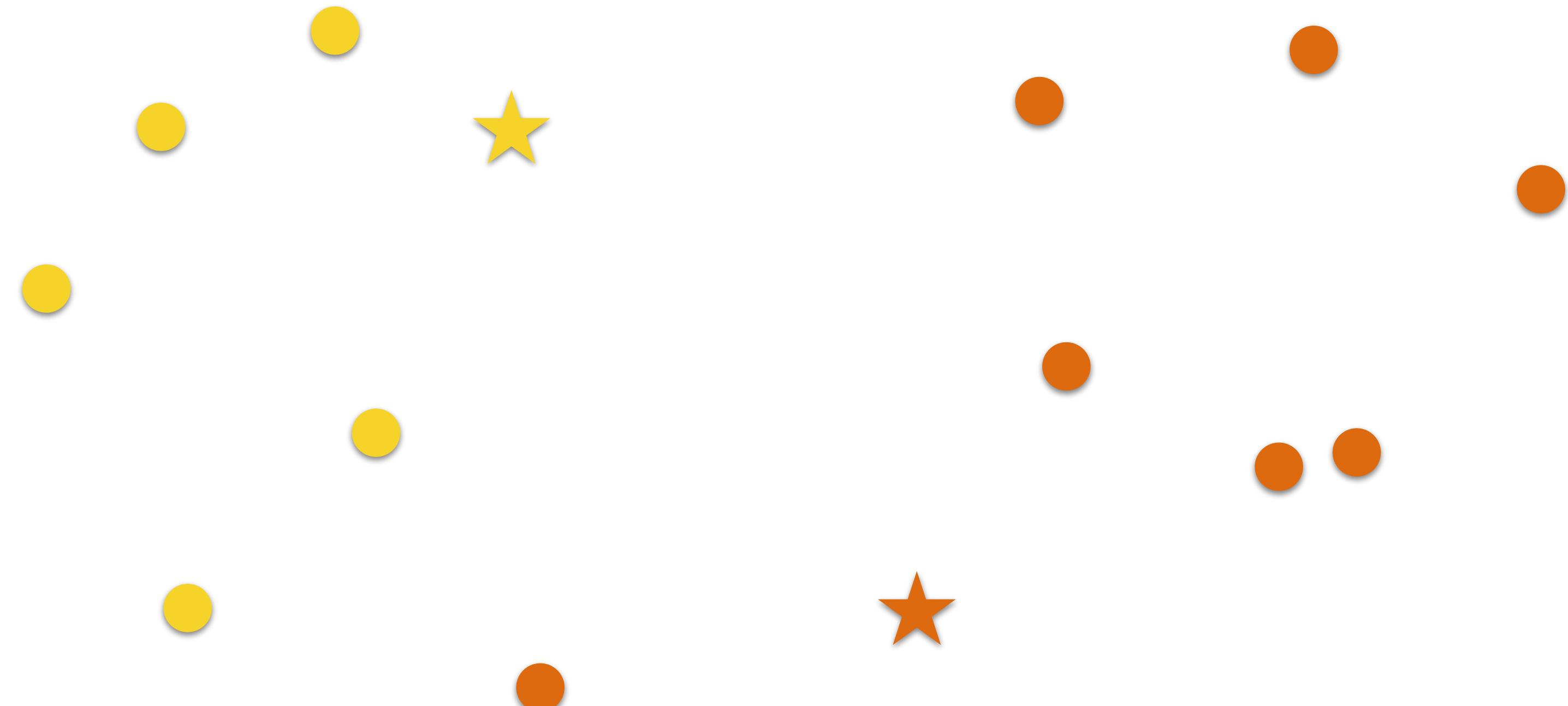
k-means

- **Step 1:** define k random centers (here: $k=2$)
- assign each point to the closest one of these centers



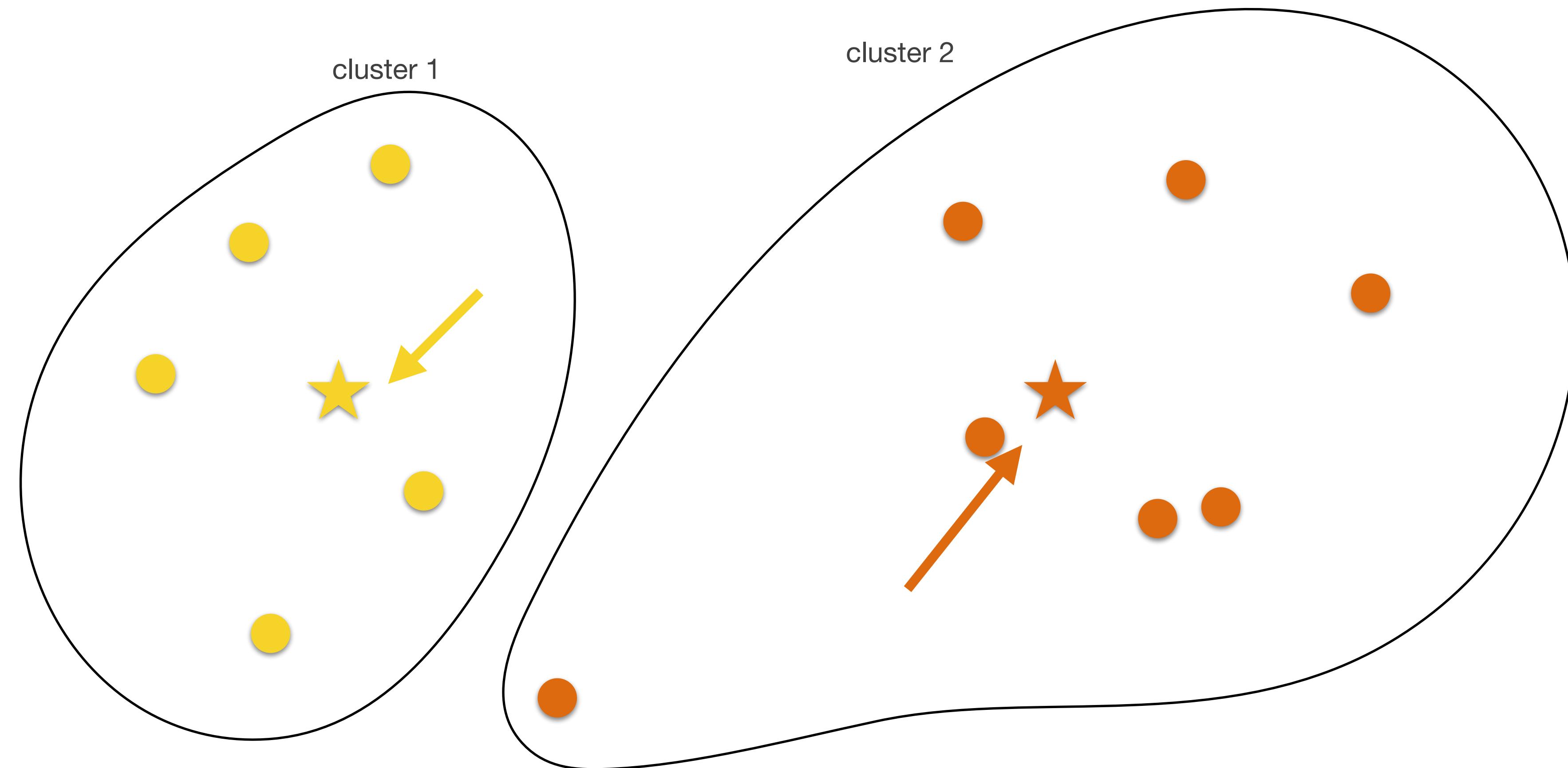
k-means

- **Step 1:** define k random centers (here: $k=2$)
- assign each point to the closest one of these centers



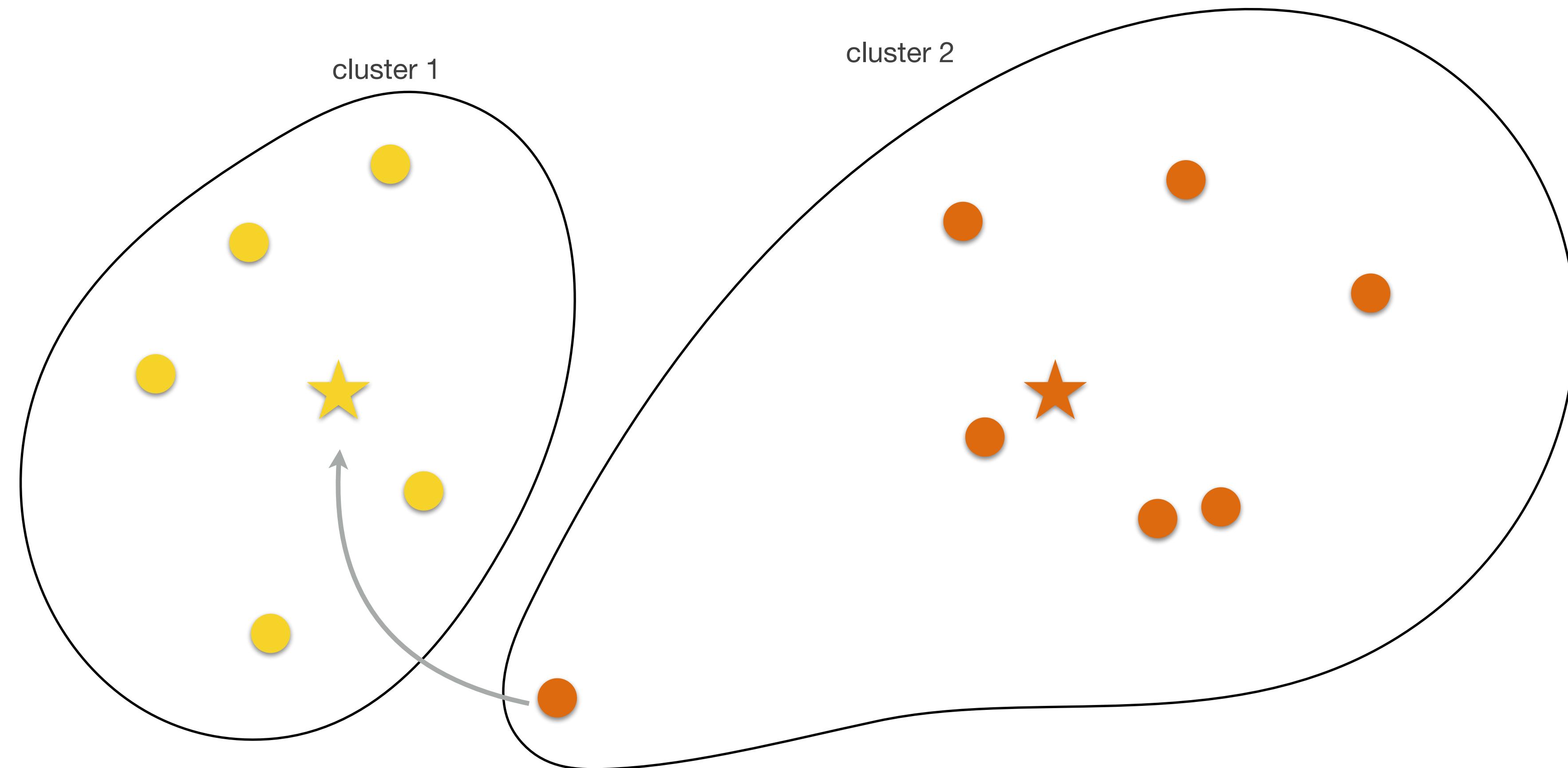
k-means

- **Step 2:** determine the **center of gravity** of these 2 groups of objects
- assign each point to the closest one of these centers



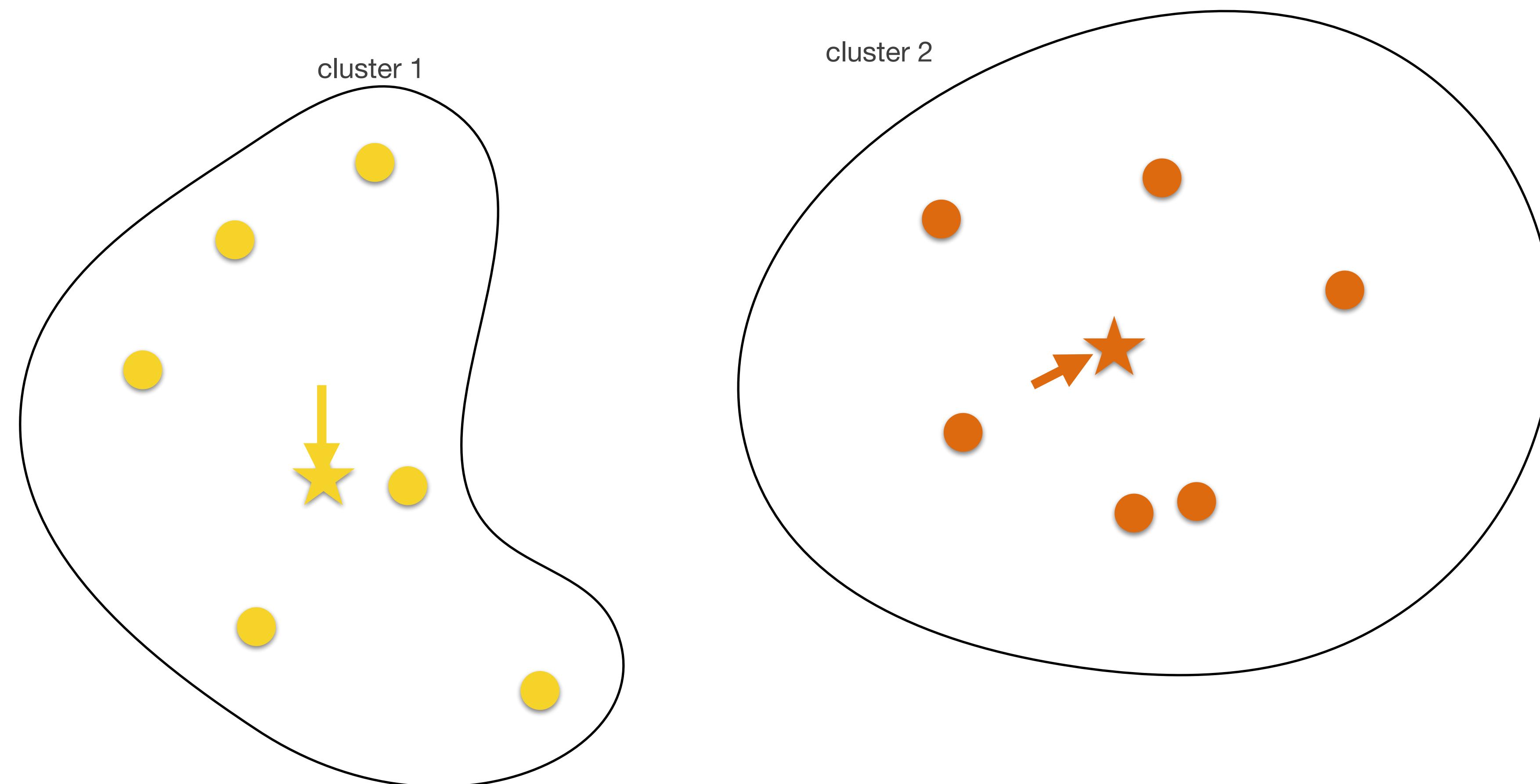
k-means

- **Step 3:** assign again each object to the updated centers
- assign each point to the closest one of these centers



k-means

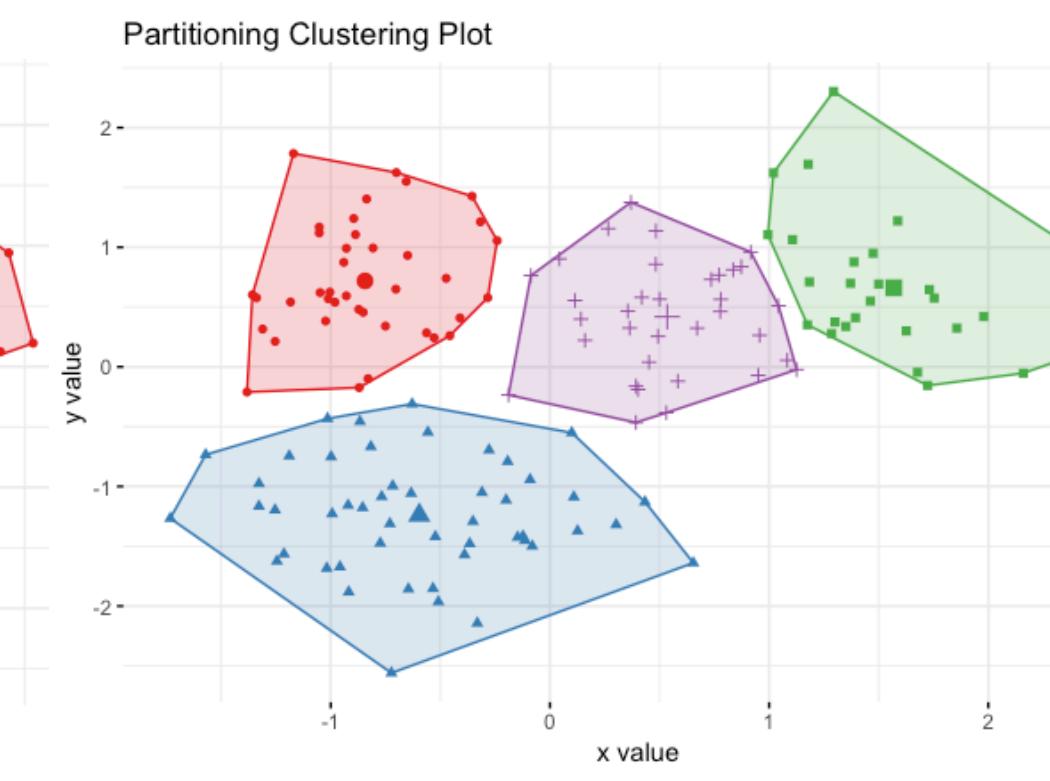
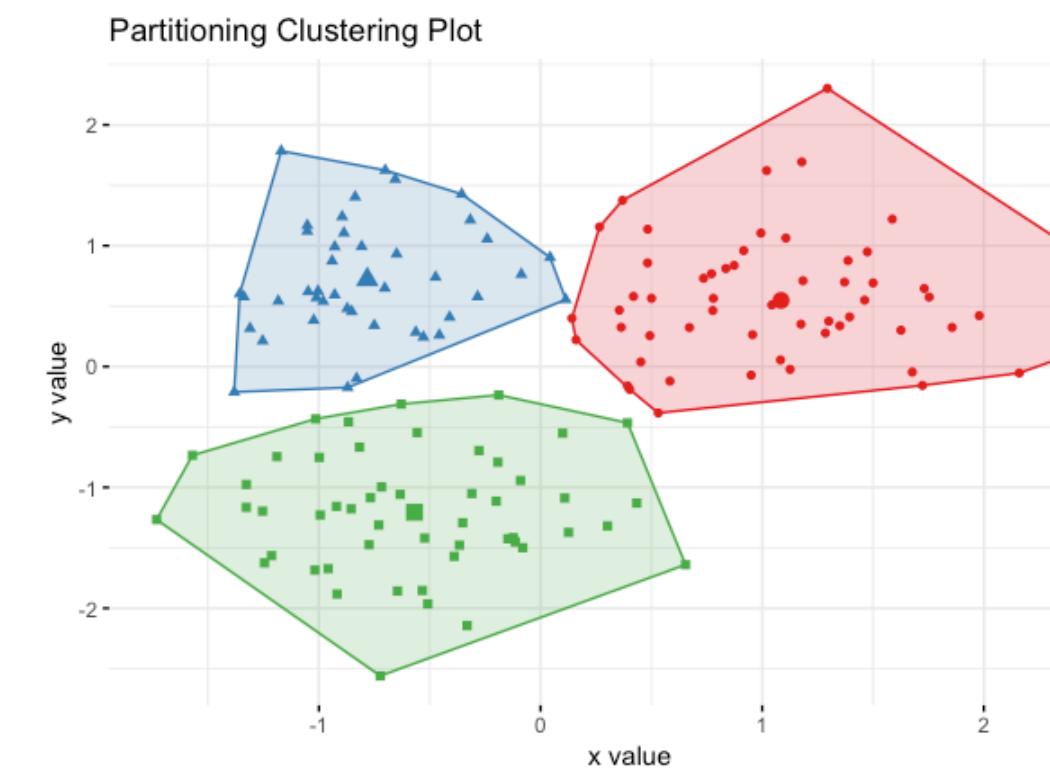
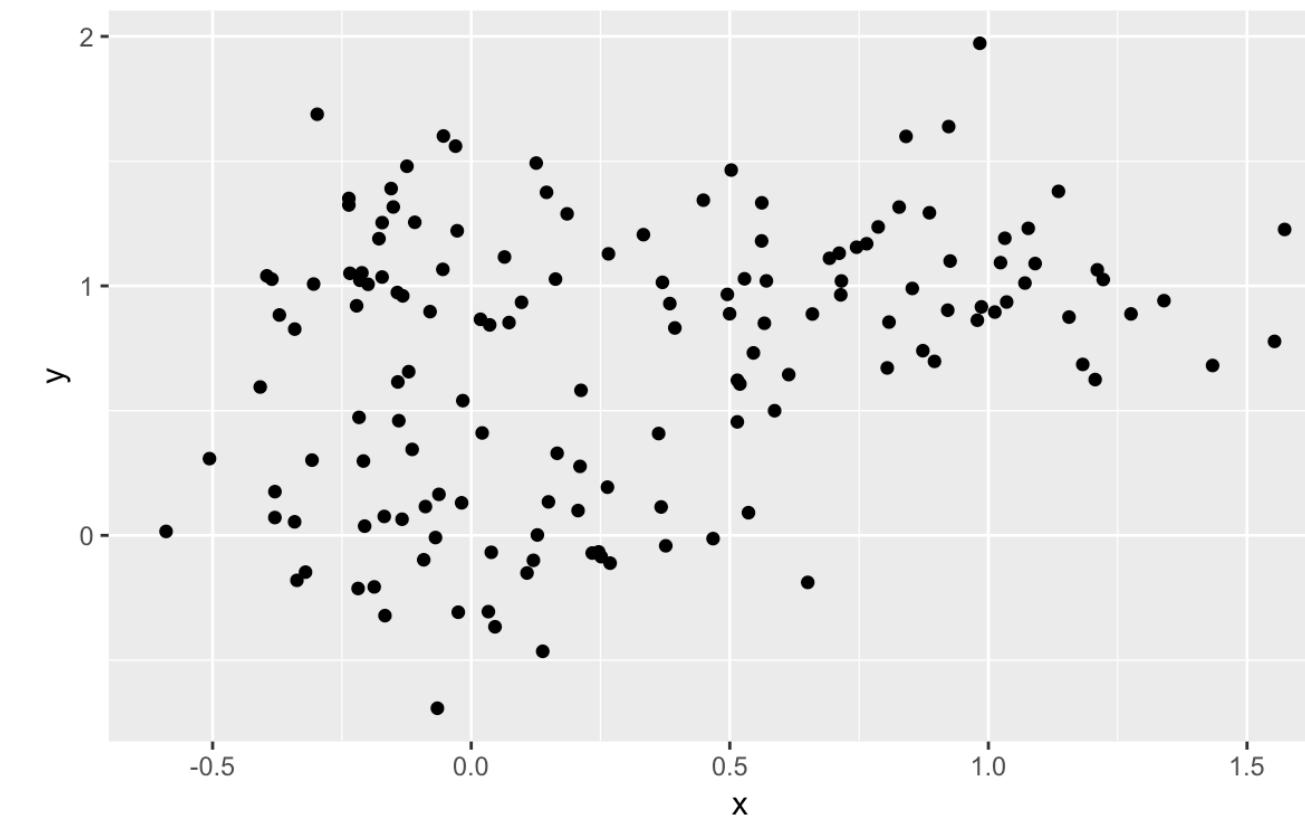
- **Step 4:** update the centers according to the new assignment, and start over



k-means

- When to stop:
 - when no object changes its cluster assignment anymore
 - or: when a certain number of max iterations are reached
- Process should be repeated over a number of random initial conditions!
- Verify the consistency of the clustering obtained using several random conditions
 - identical = very strong cluster structure
 - fluctuating = no clear cluster structure
- **Clusters will be identified, even if there is no group structure in the data!**

k-means: choosing k



Which of these clusterings is better and why?

k-means: choosing k

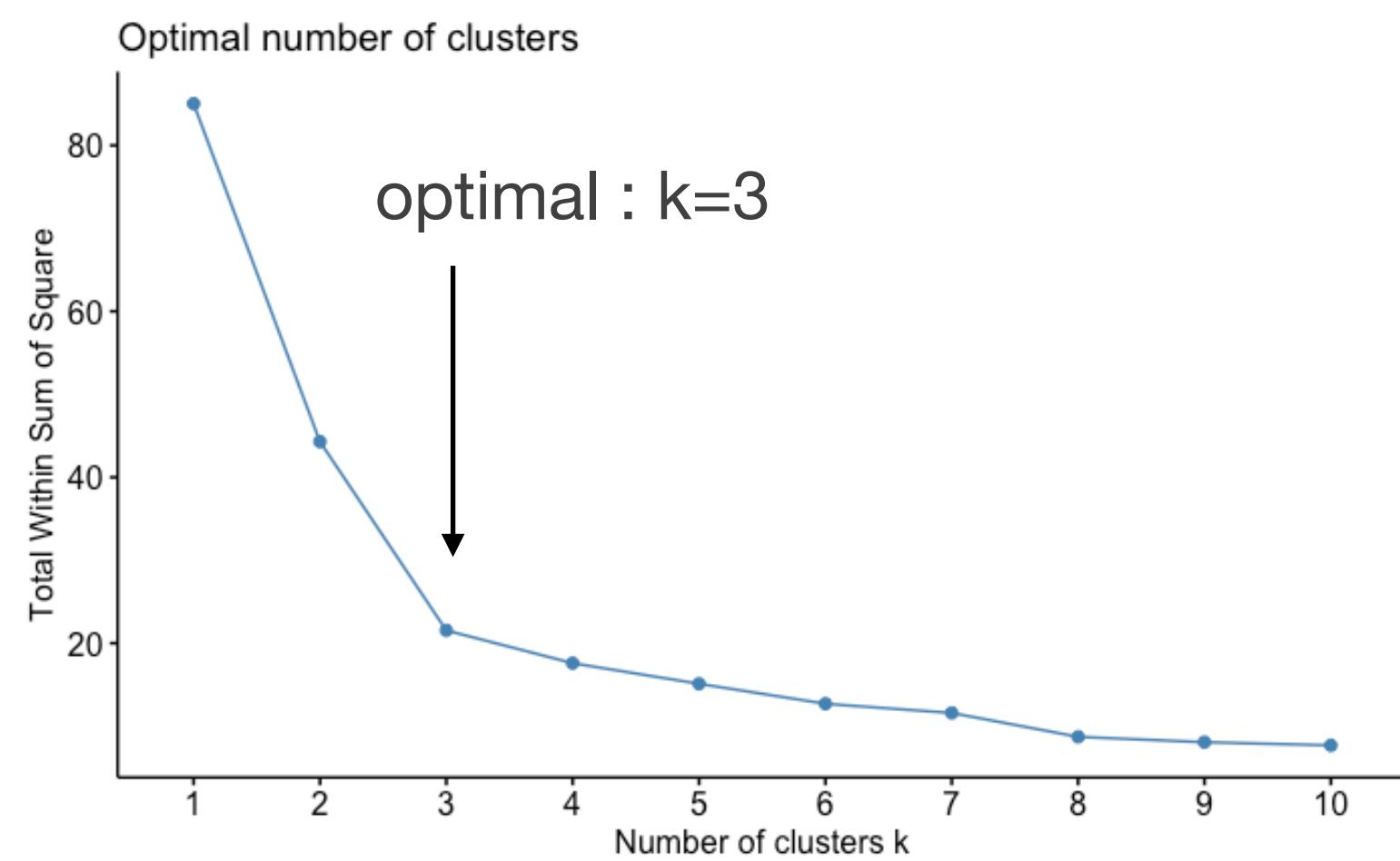
- **Fundamental idea**
in a good clustering, the **distance whithin clusters** should be much smaller than the **distance between clusters**
- Several methods implement this principle
 - **elbow** method:
 - **silhouette** method
 - (many other methods exist to define optimal cluster number...)

k-means: elbow method

- Sum all **pairwise distances squared** between members of the same cluster (within square distance = WSS)

$$WSS = \sum_{i,j \in \text{same cluster}} x_i - x_j^2$$

- If each point is its own cluster, then WSS = 0
- WSS decreases with increasing k
- when optimal number of clusters is reached, adding additional clusters does not improve the WSS that much
→ kink (or “elbow”) in the curve



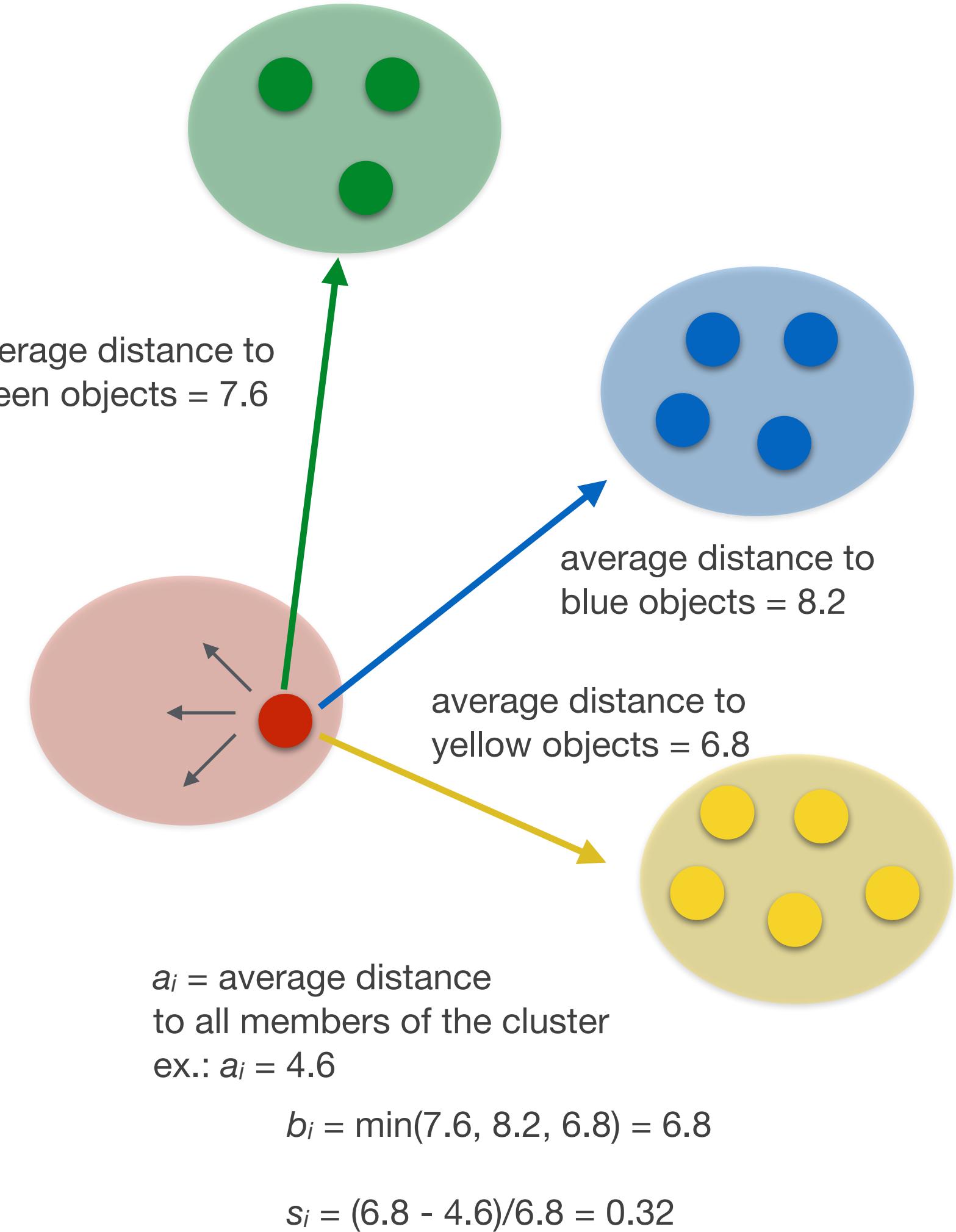
k-means: silhouette method

- for each element i, compute
 - mean distance a_i to all members of his cluster
 - smallest average distance b_i to members of all other clusters
- Define silhouette value as

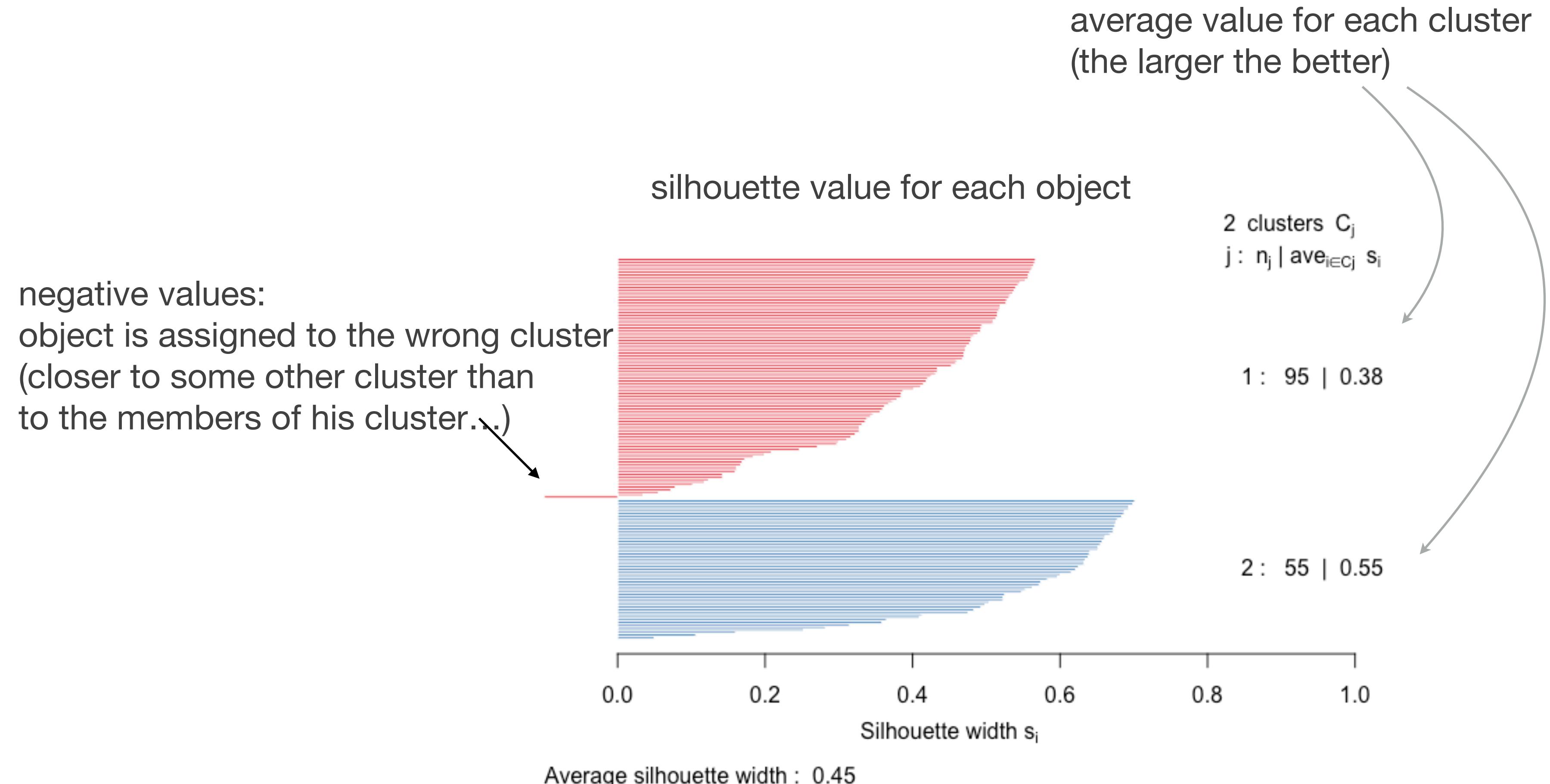
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

$s_i = 0$ if only one point in cluster

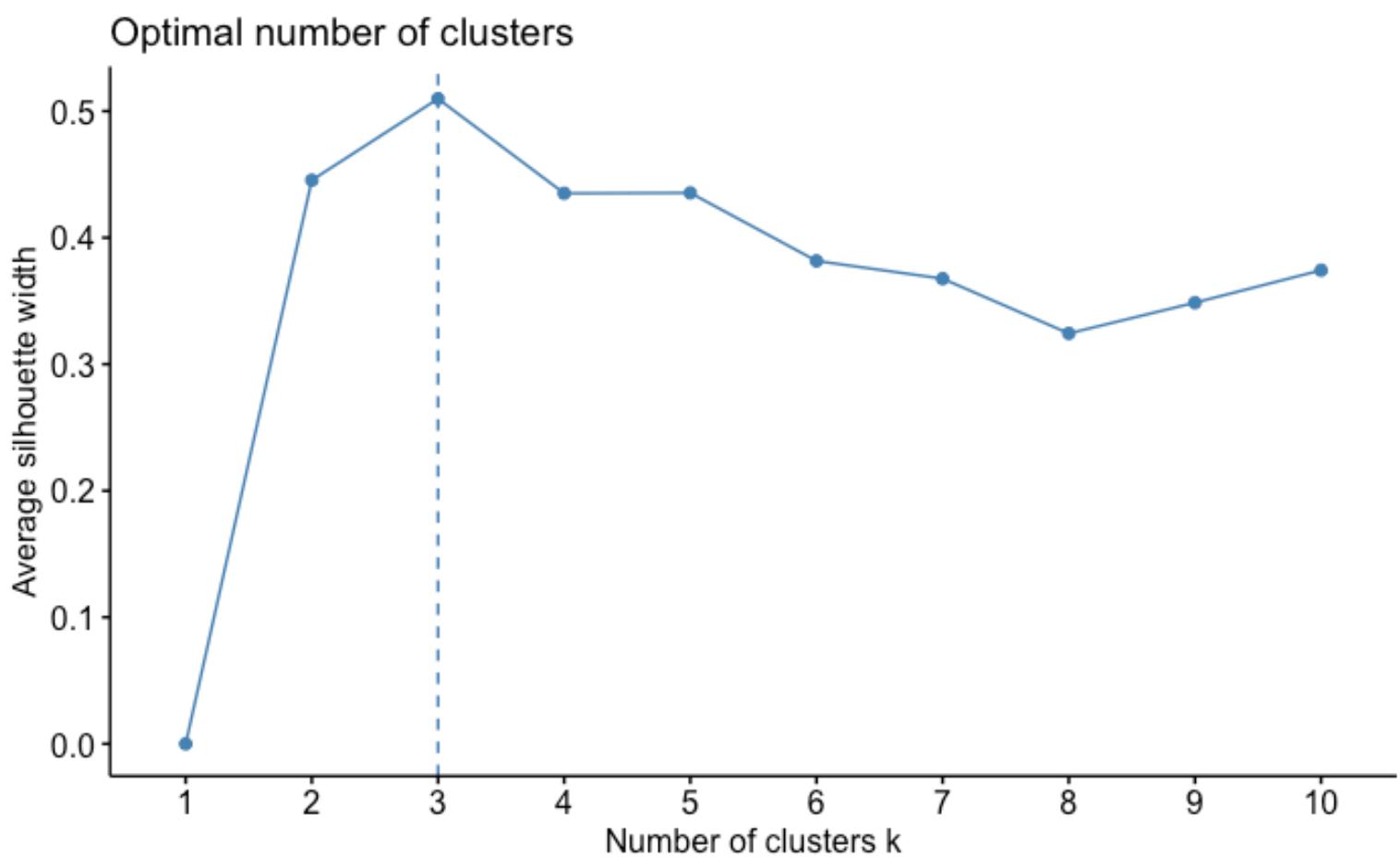
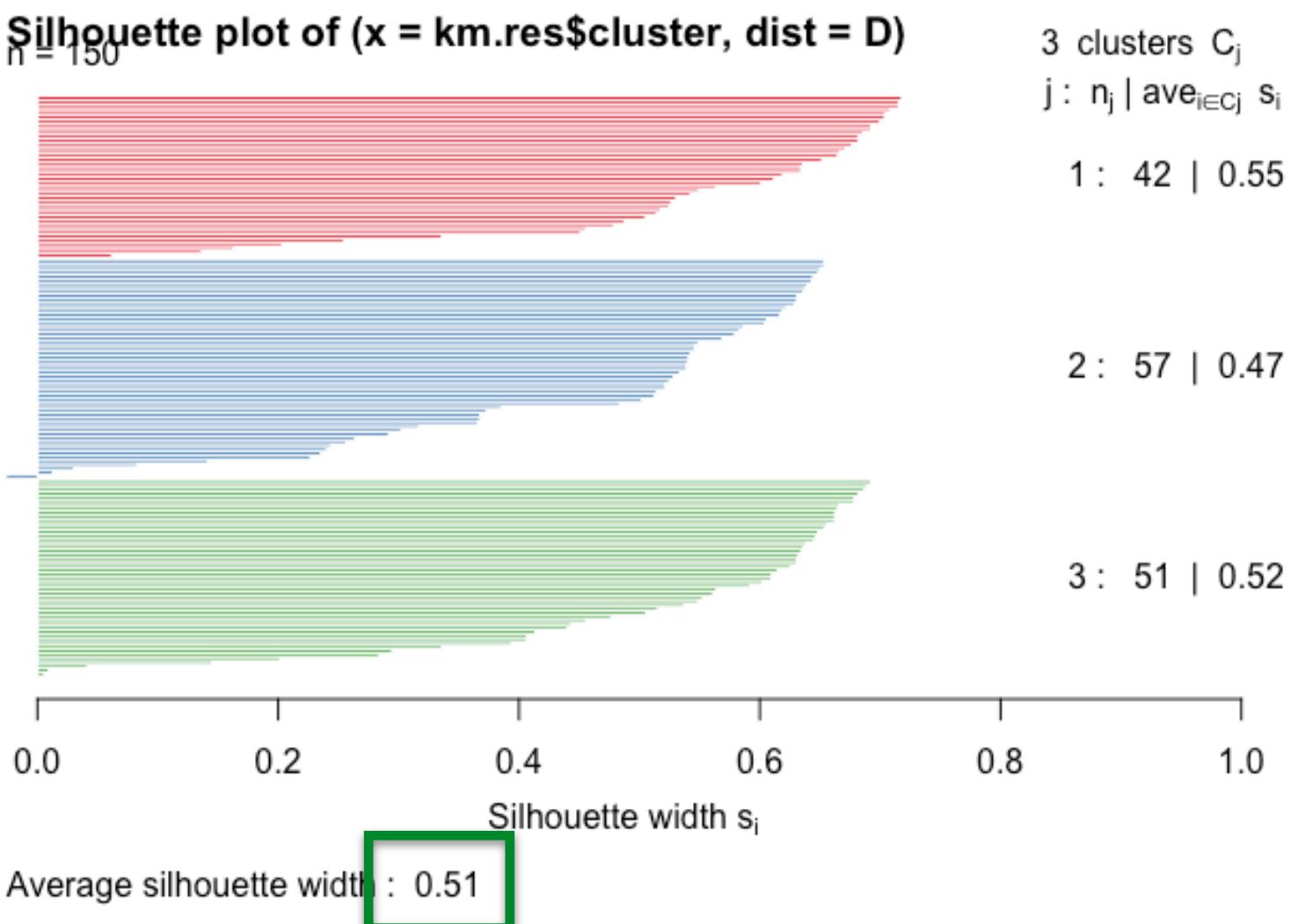
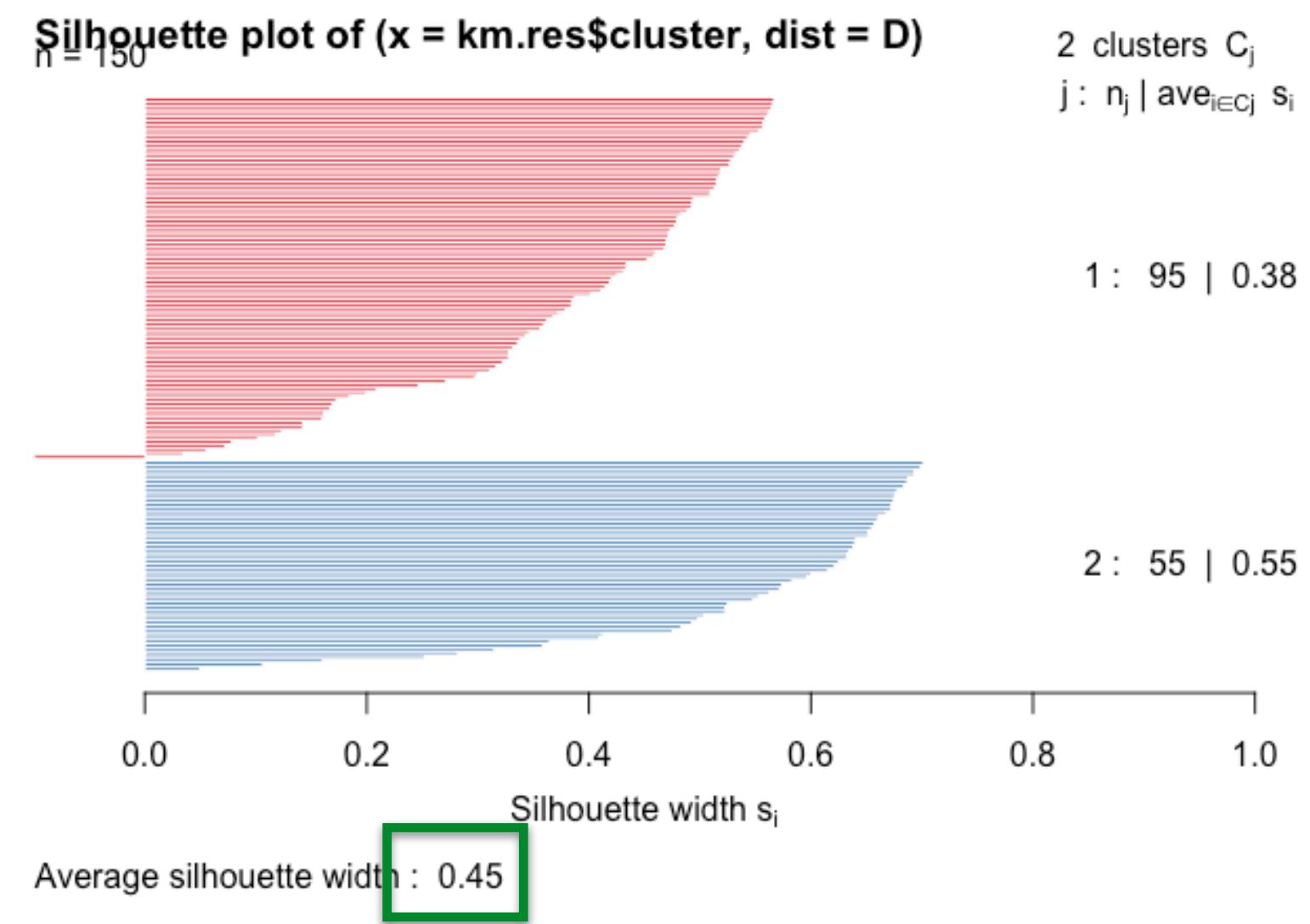
- Properties
 - $s_i \sim 1$: object very well clustered
 - $s_i \sim 0$: ambiguous object
 - $s_i < 0$: wrongly assigned
- Convention : $s_i = 0$ if only one element in the cluster!



k-means: silhouette method



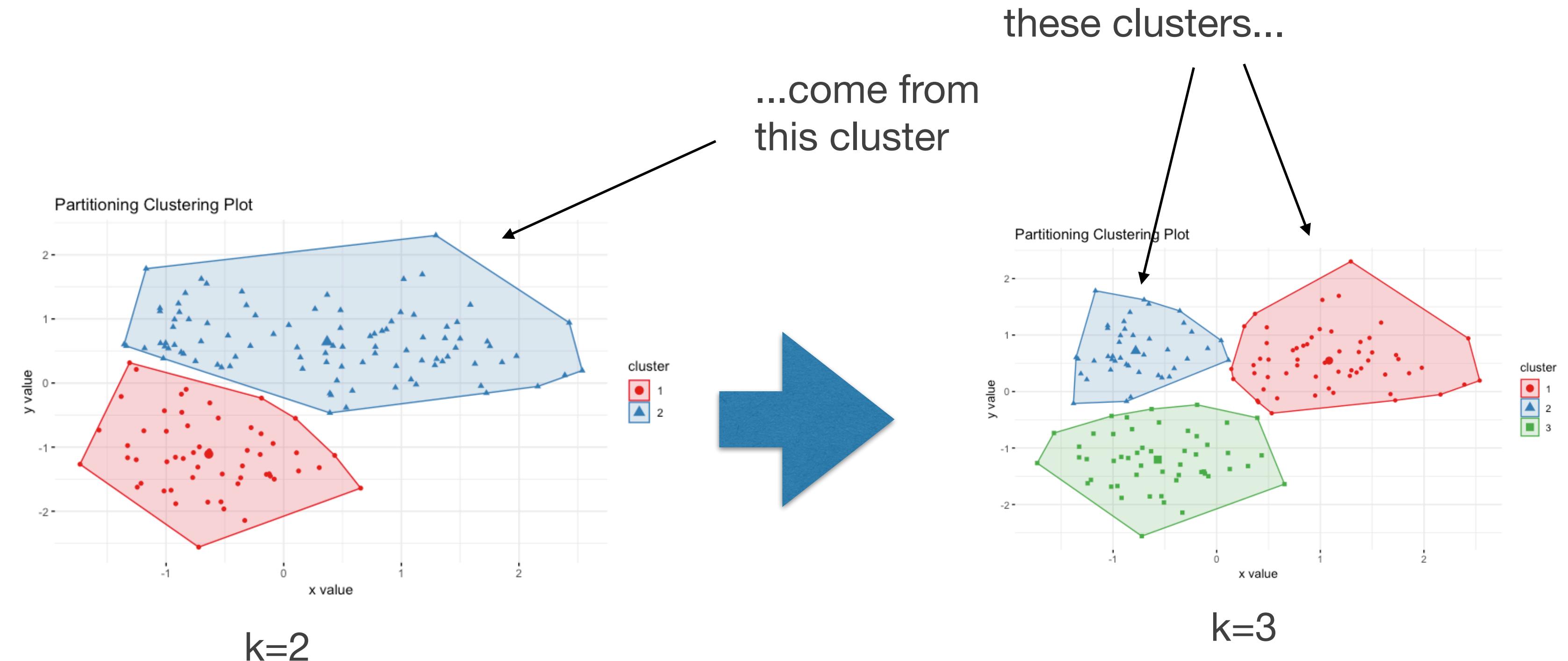
k-means: silhouette method



4. finding structure in the data hierarchical clustering

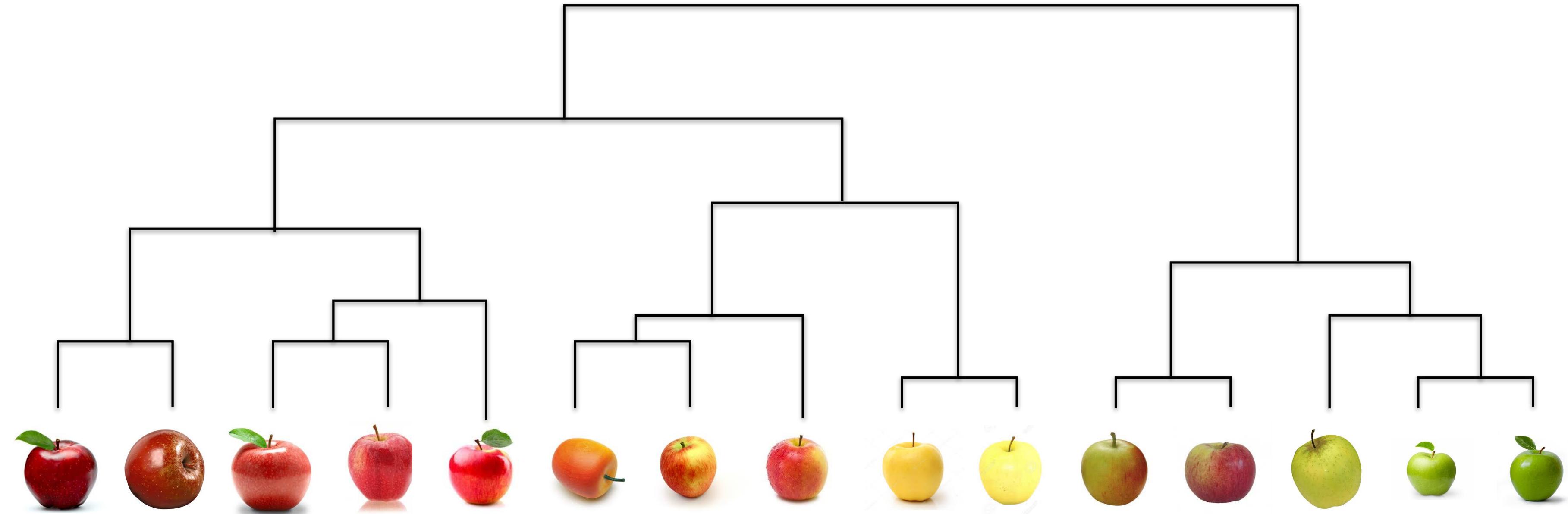
Clustering

- There is often a hierarchical structure in the data that is not captured by the k-means clustering



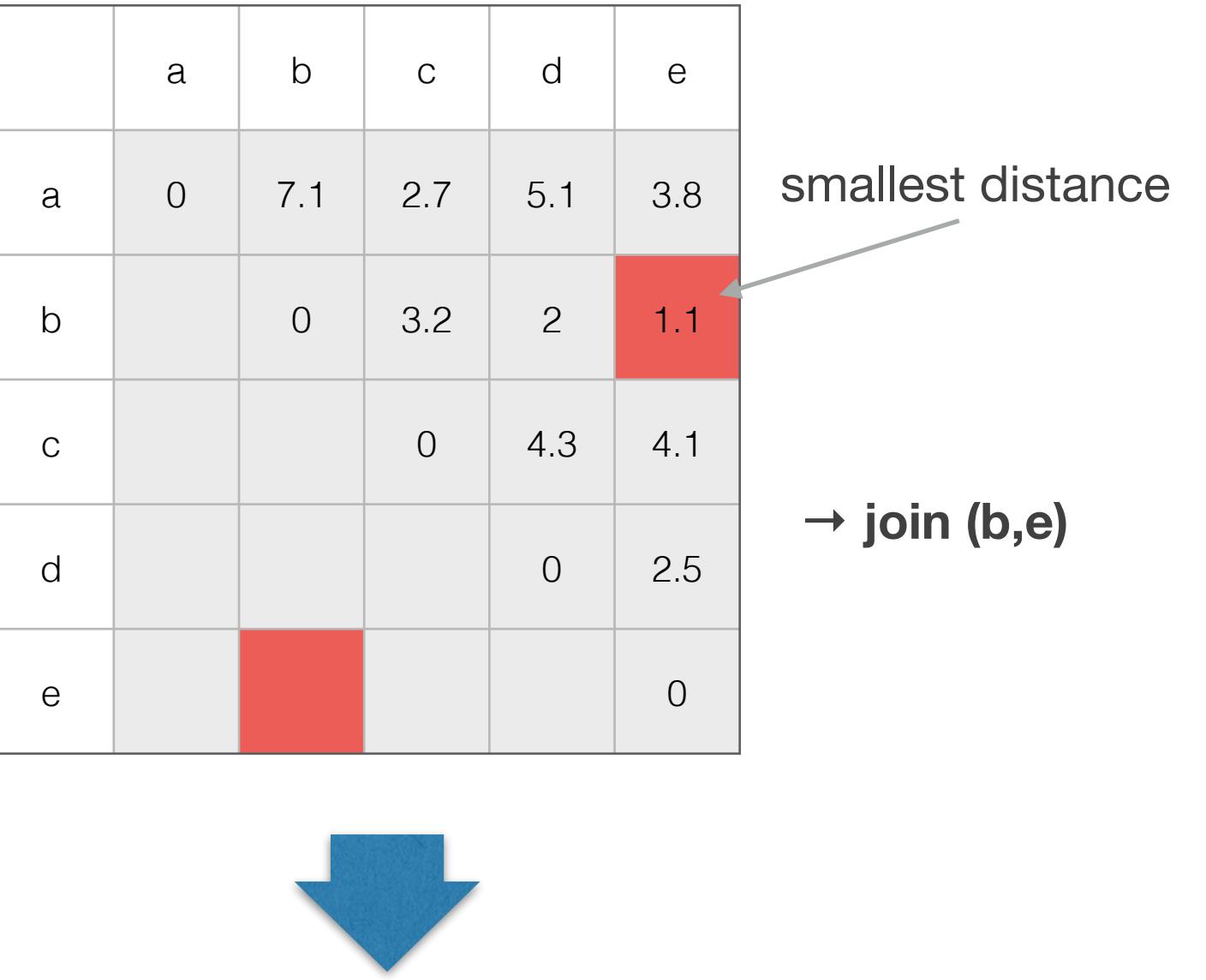
Hierarchical clustering

- **Divisive clustering** (also "top-down"): iteratively split the full group into sub-groups
- **Agglomerative clustering** (also "Bottom-up"): group iteratively objects from most similars to less similar groups
- tree-like representation of the relationships/distances between objects

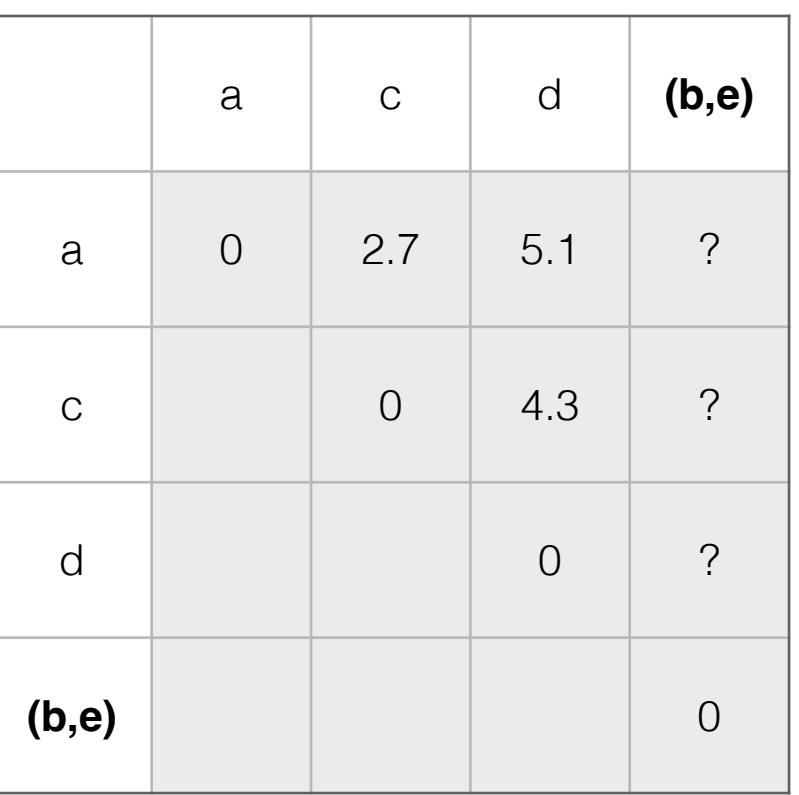


Hierarchical clustering

- Starting point: **distance / similarity matrix** between all objects
- Agglomerative** method (“bottom up”)
 - Step 1 : find closest pair (x,y)
 - Step 2 : join objects to form a cluster (x,y)
 - Step 3 : replace x and y in the distance matrix by cluster (x,y) and recompute all distances
 - Step 4 : Repeat from Step1
 - Stop when all objects have been merged!



- Distance between single objects: based on distance measure
- Distance between object and cluster?
 - several possible definitions (= “linkage methods”)



	a	c	d	(b,e)
a	0	2.7	5.1	?
c		0	4.3	?
d			0	?
(b,e)				0

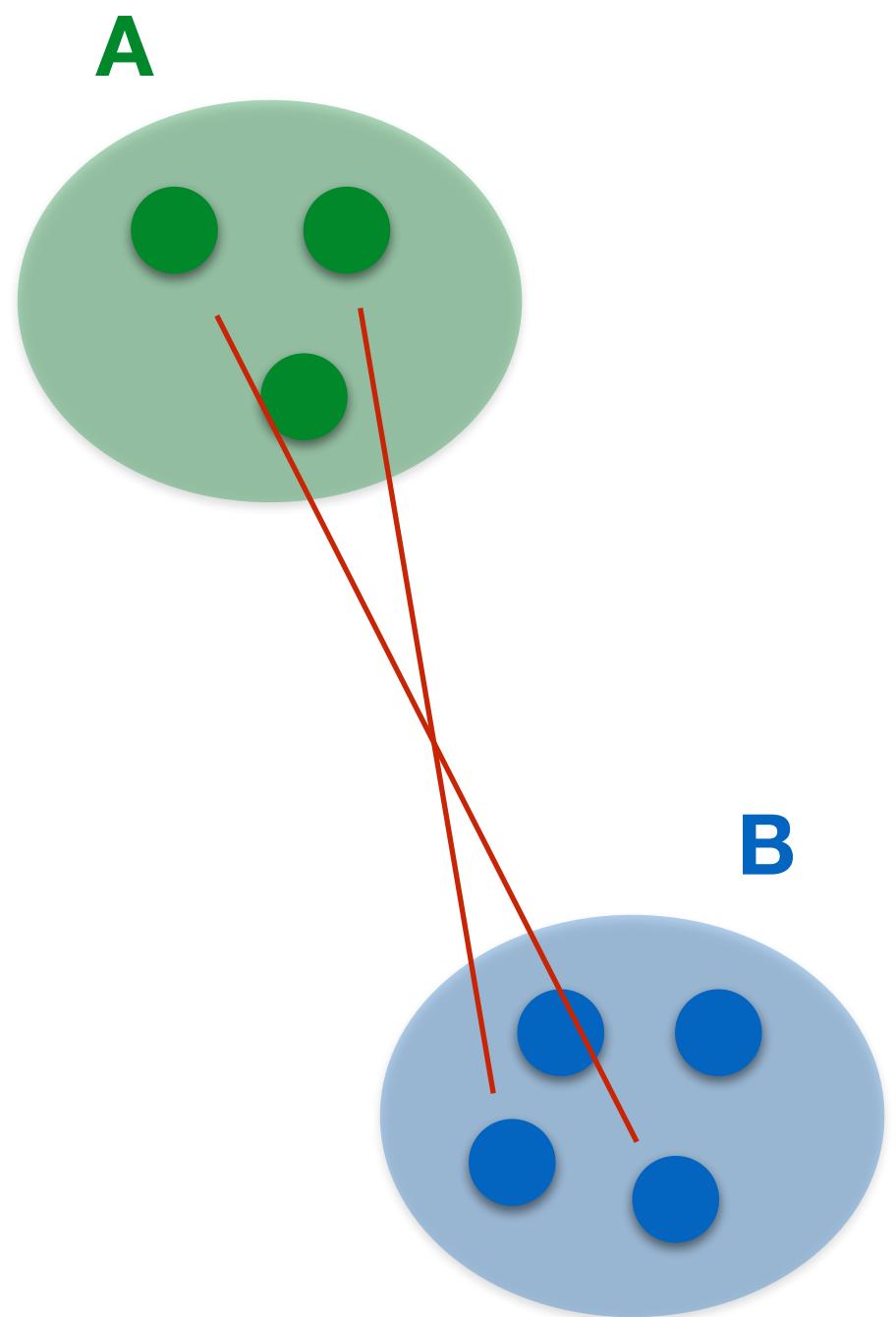
Linkage methods

- Several possible definitions of the distance between two clusters of objects (or an object and a cluster)

- **single-linkage**

$d(A, B) = \text{minimal distance between all elements of } A \text{ and } B$

$$d_{\text{single linkage}} = \min_{i,j} (d(a_i, b_j))$$



- **complete-linkage**

$d(A, B) = \text{maximal distance between all elements of } A \text{ and } B$

$$d_{\text{complete linkage}} = \max_{i,j} (d(a_i, b_j))$$

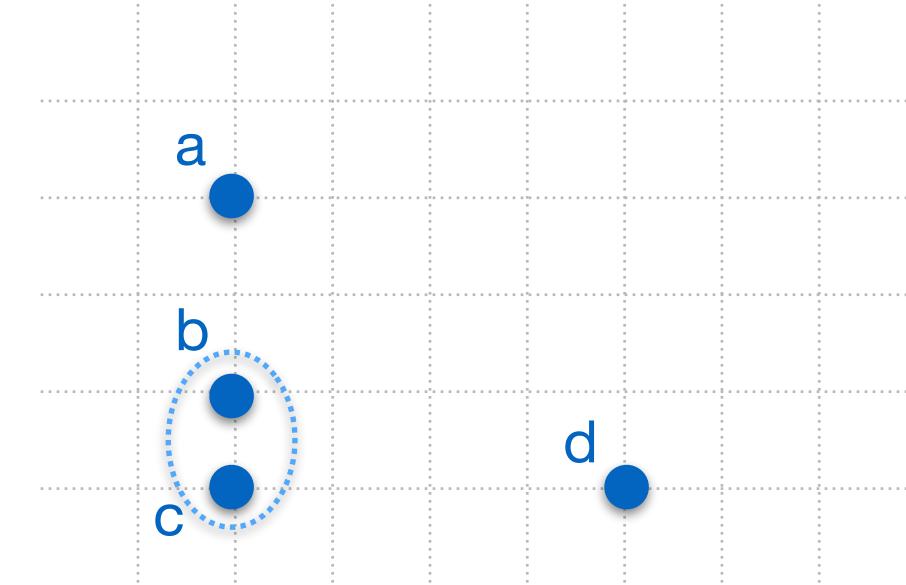
- **average linkage (UPGMA)**

$d(A, B) = \text{average of all pairwise distances between elements of } A \text{ and } B$

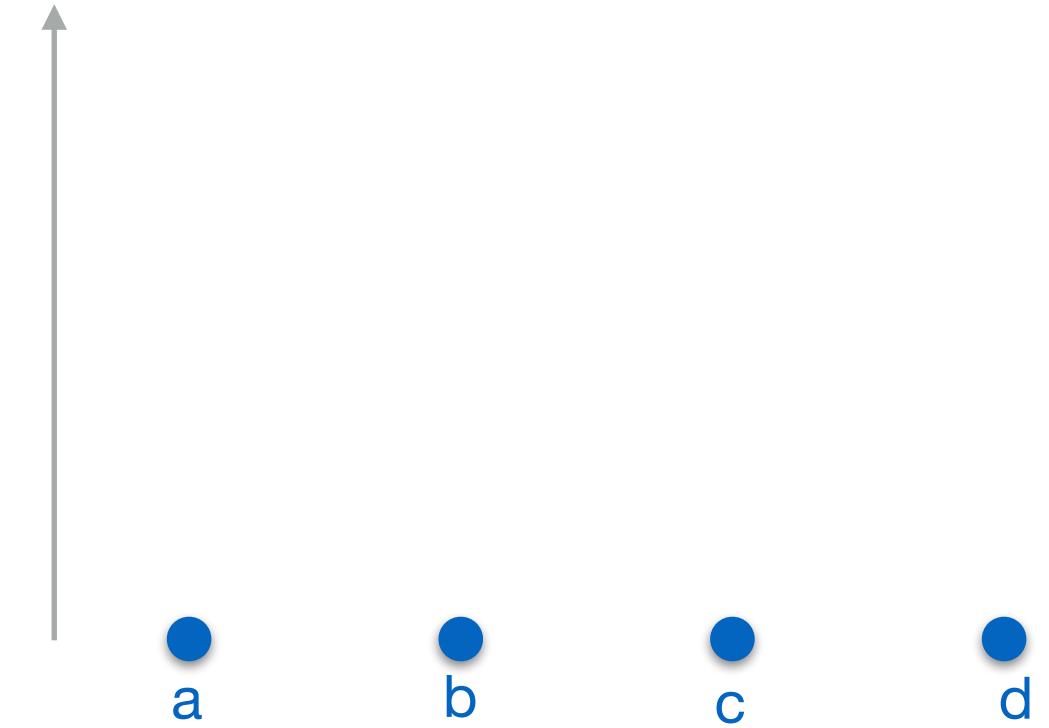
$$d_{\text{average linkage}} = \frac{1}{A \cdot B} \sum_i \sum_j d(a_i, b_j)$$

Hierarchical clustering

(single-linkage)

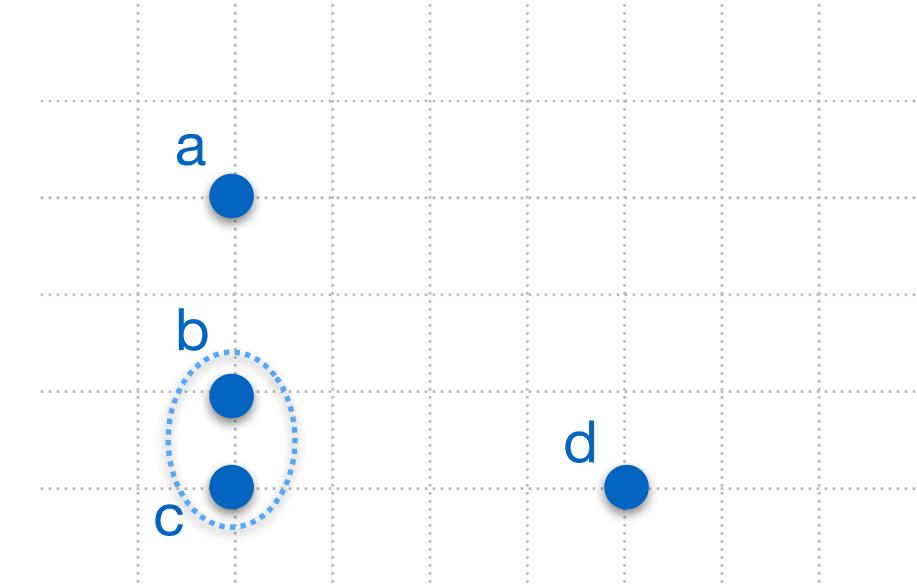


	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0

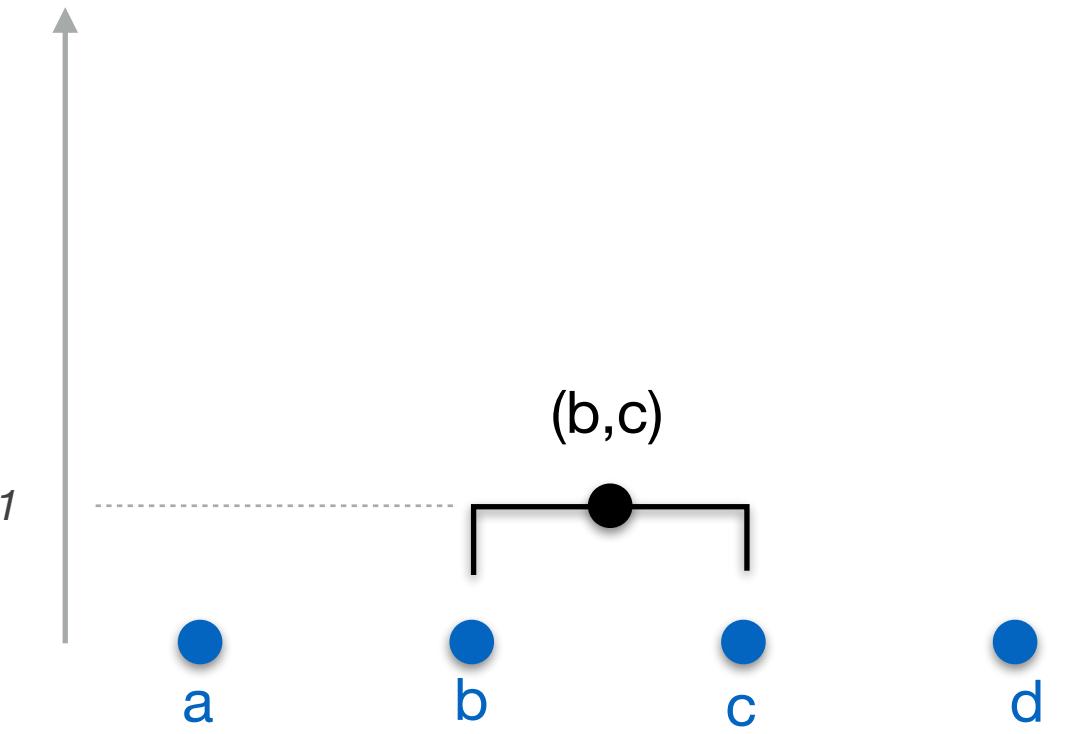


Hierarchical clustering

(single-linkage)

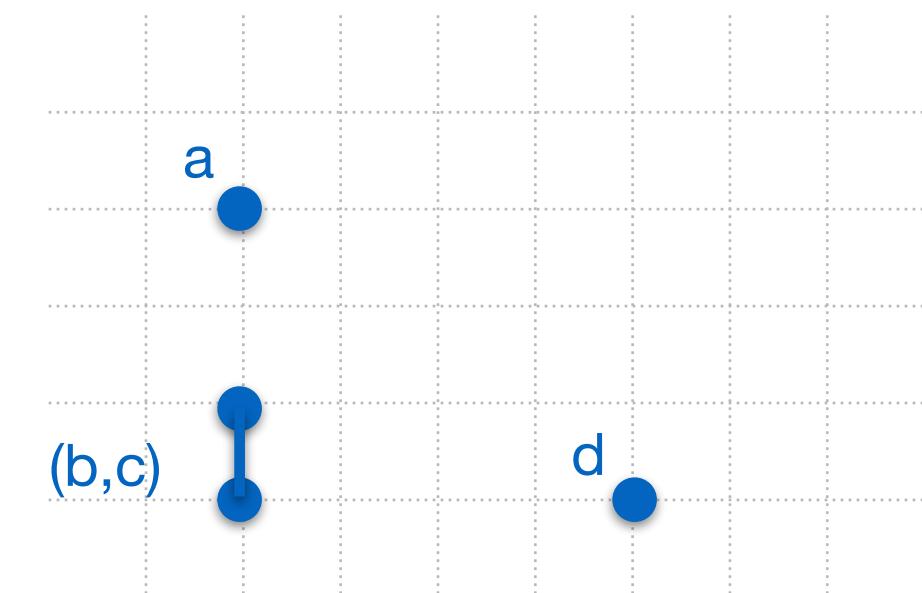
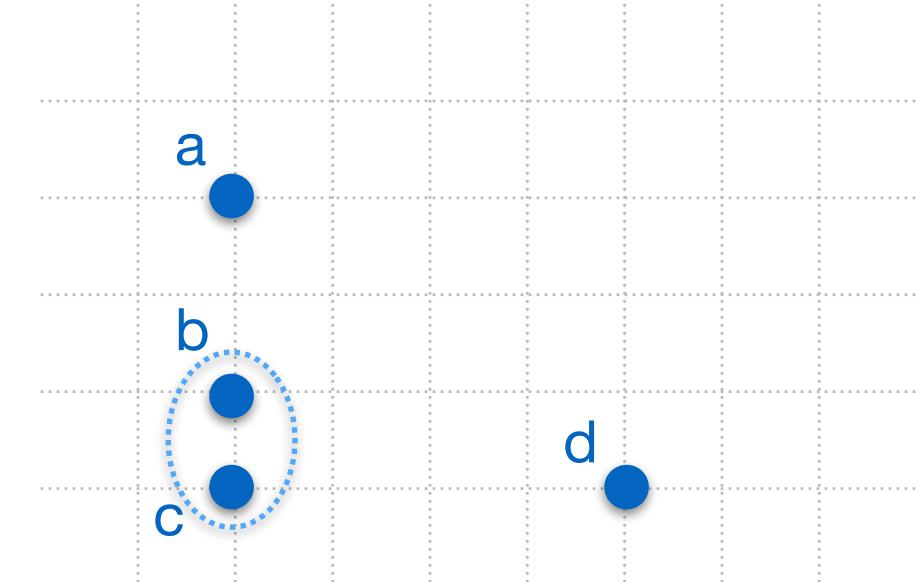


	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0



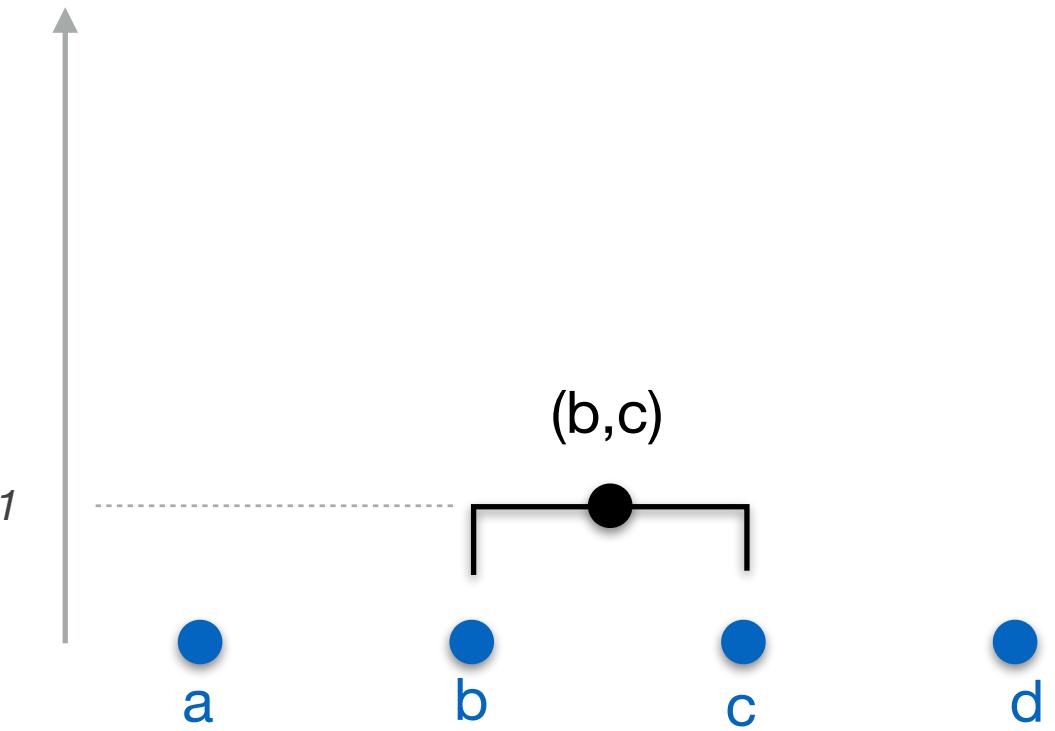
Hierarchical clustering

(single-linkage)



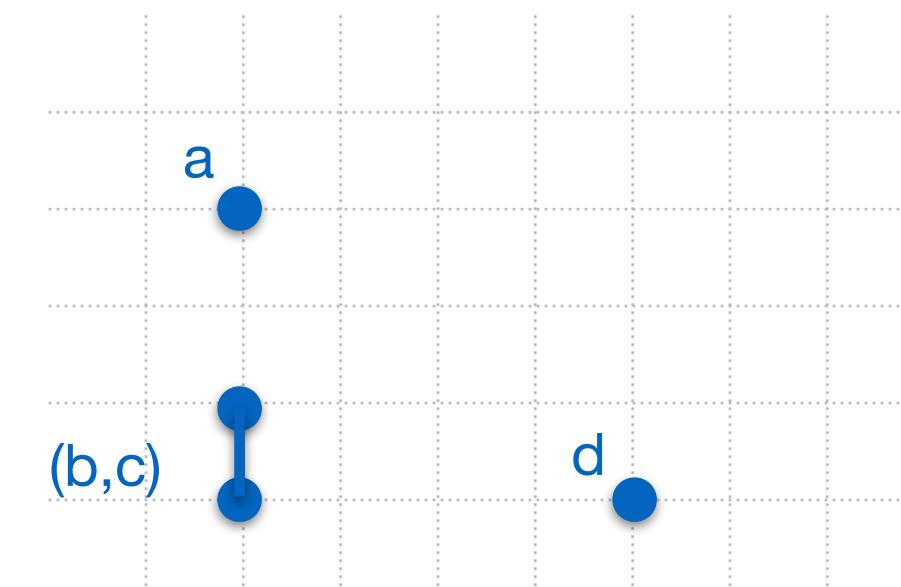
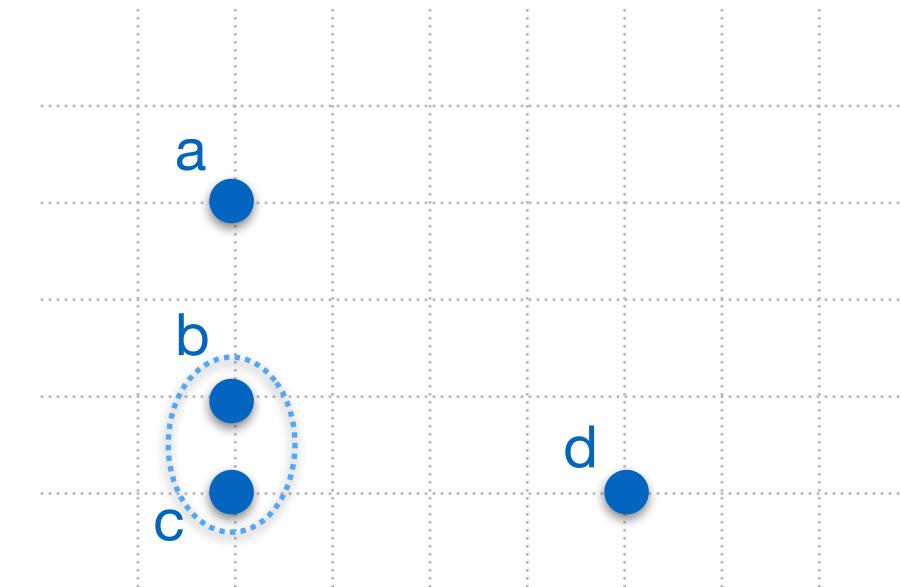
	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0

	a	(b,c)	d
a	0	2	5
(b,c)		0	4
d			0



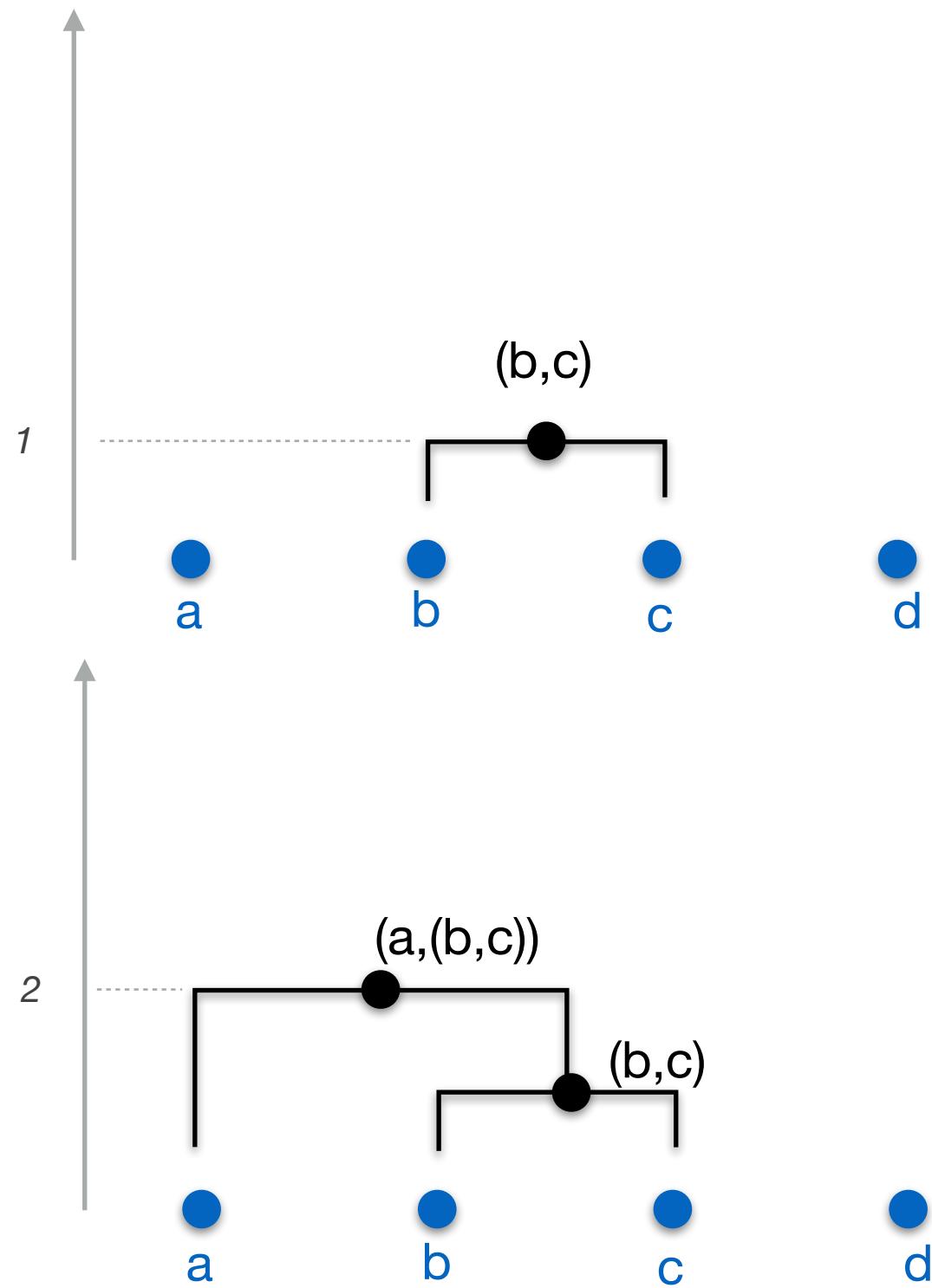
Hierarchical clustering

(single-linkage)



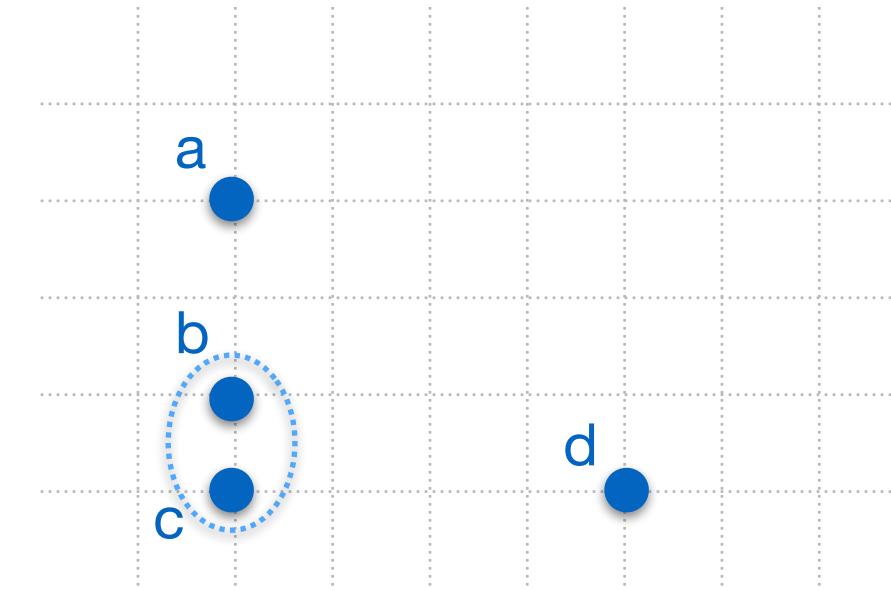
	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0

	a	(b,c)	d
a	0	2	5
(b,c)		0	4
d			0

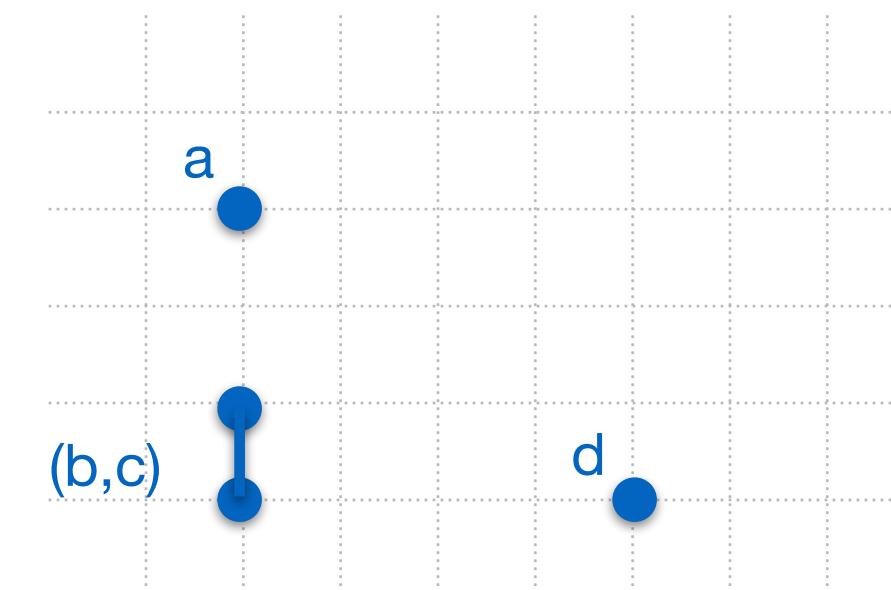


Hierarchical clustering

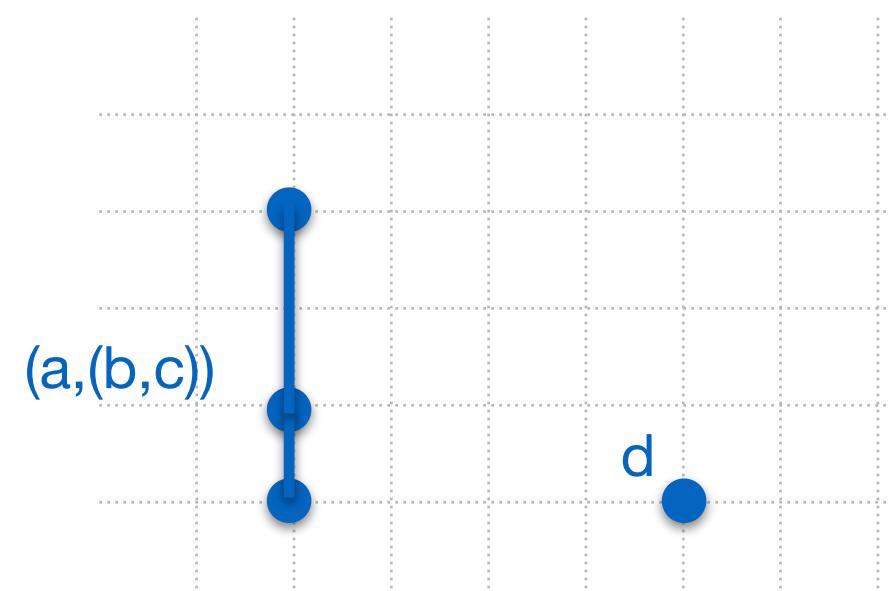
(single-linkage)



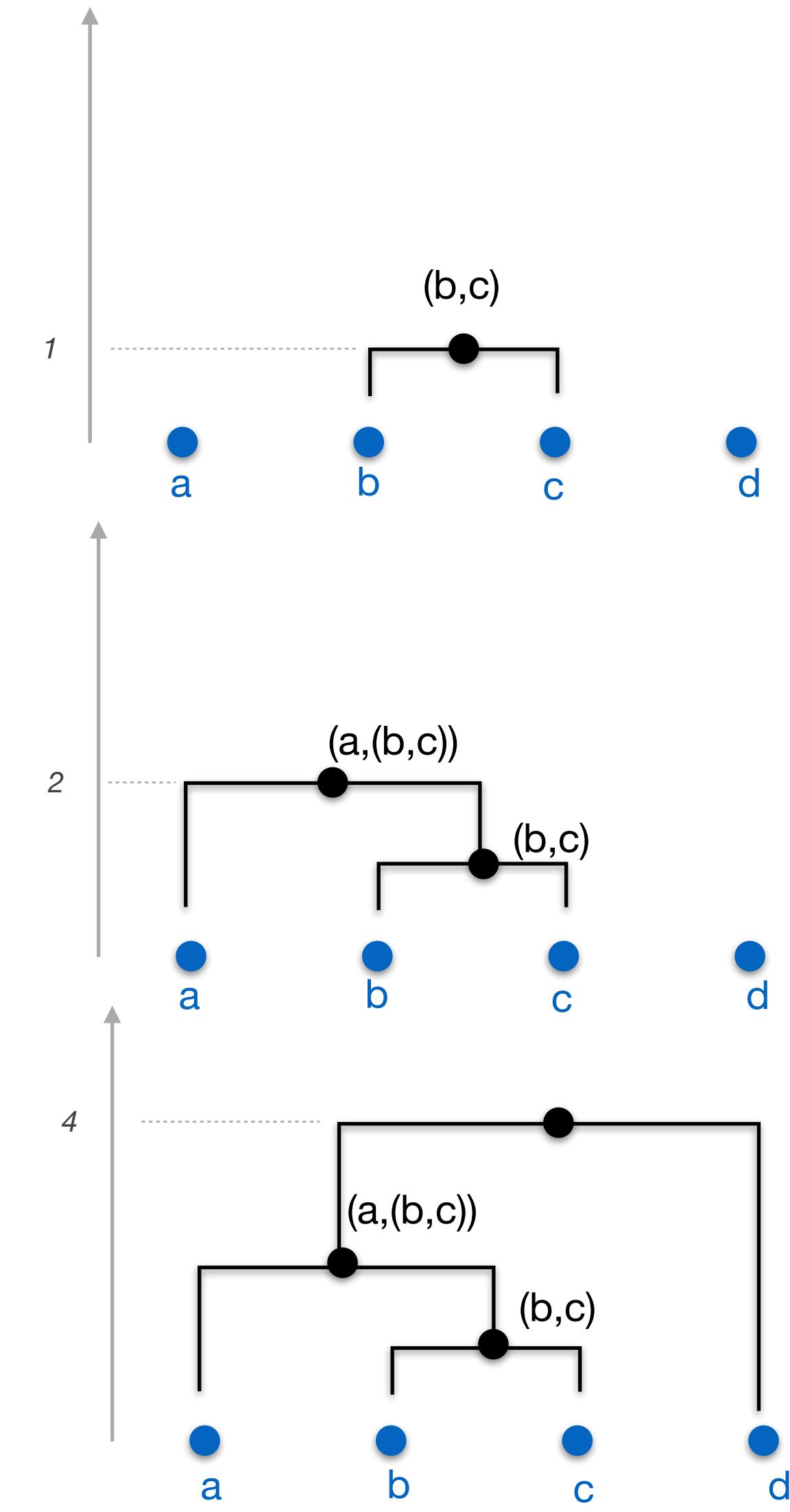
	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0



	a	(b,c)	d
a	0	2	5
(b,c)		0	4
d			0

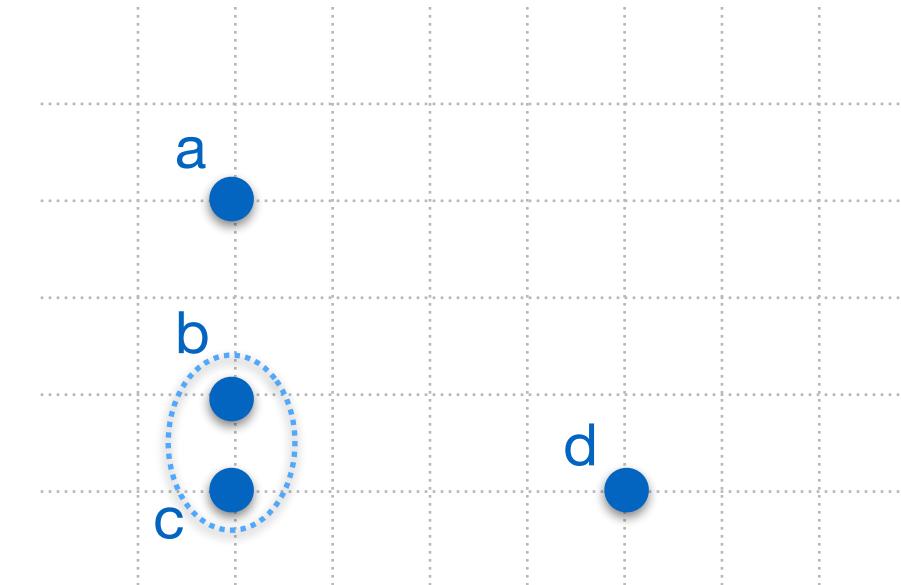


	(a,(b,c))	d
(a,(b,c))	0	4
d		0

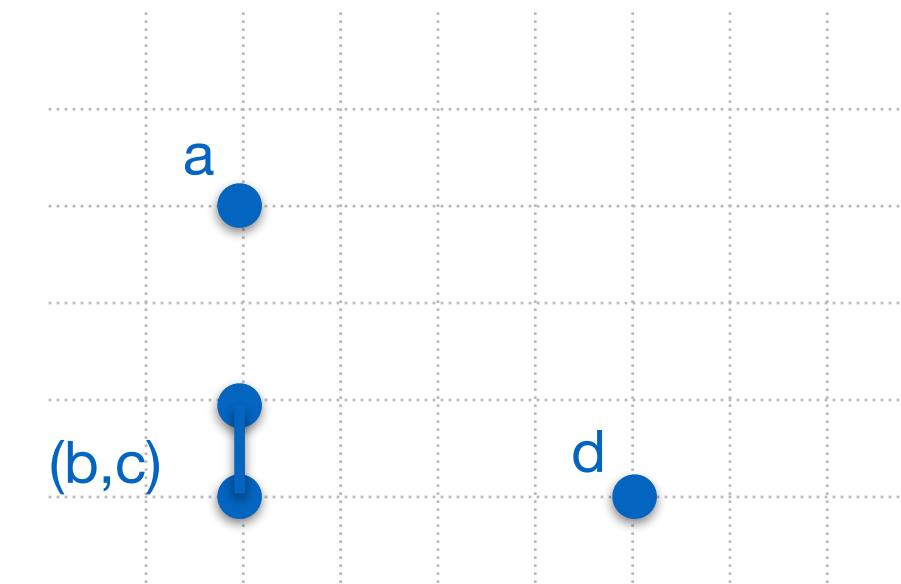


Hierarchical clustering

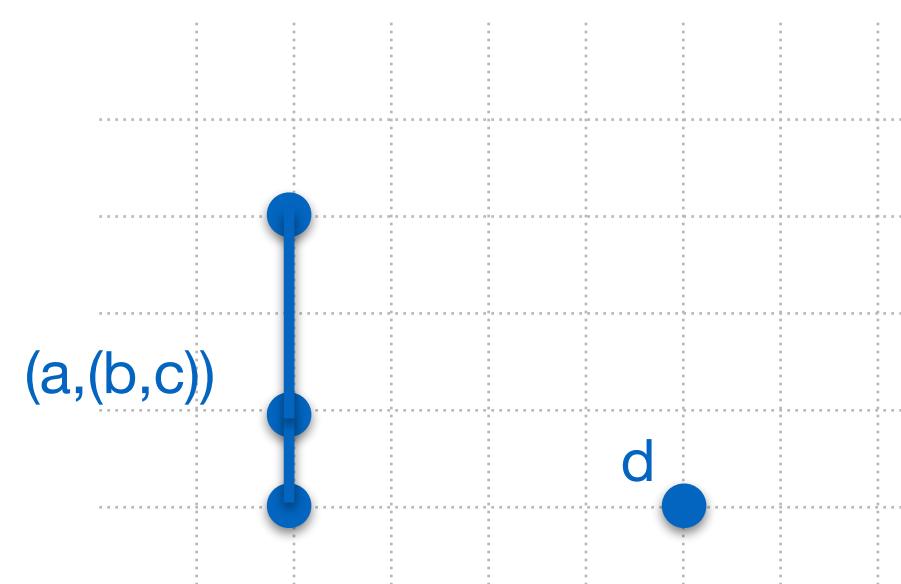
(complete-linkage)



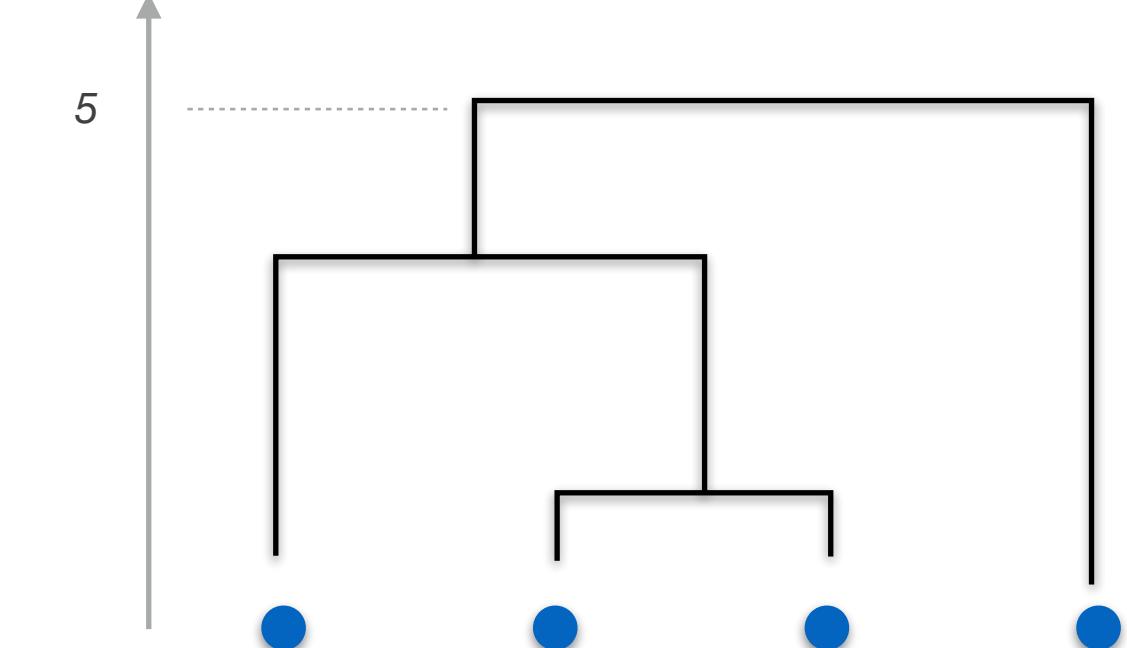
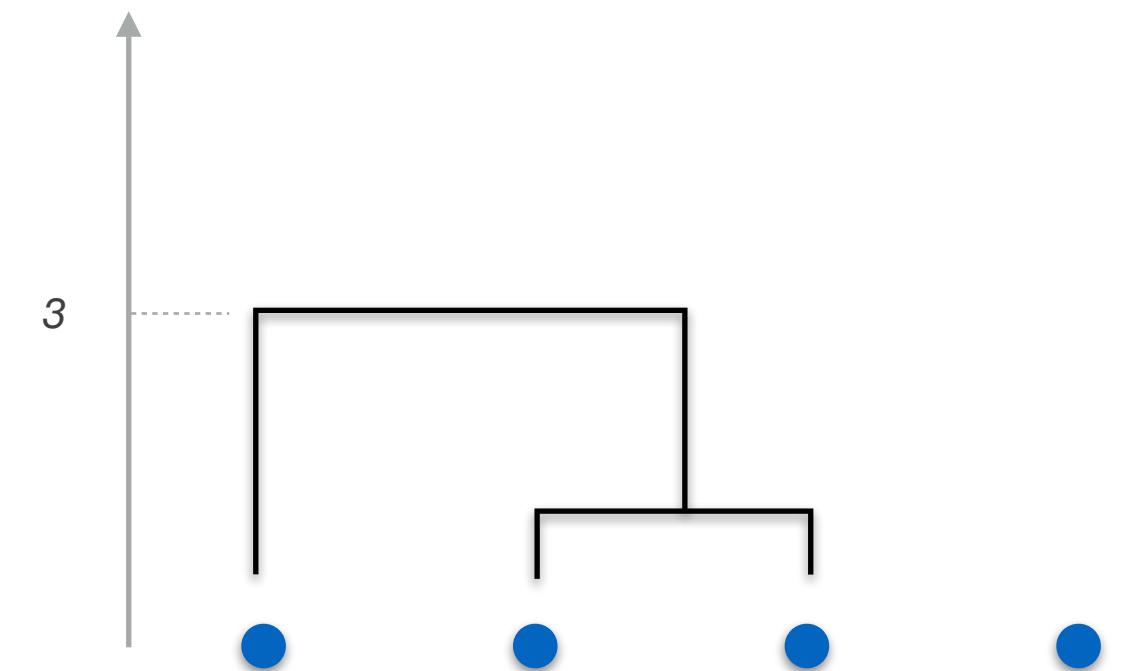
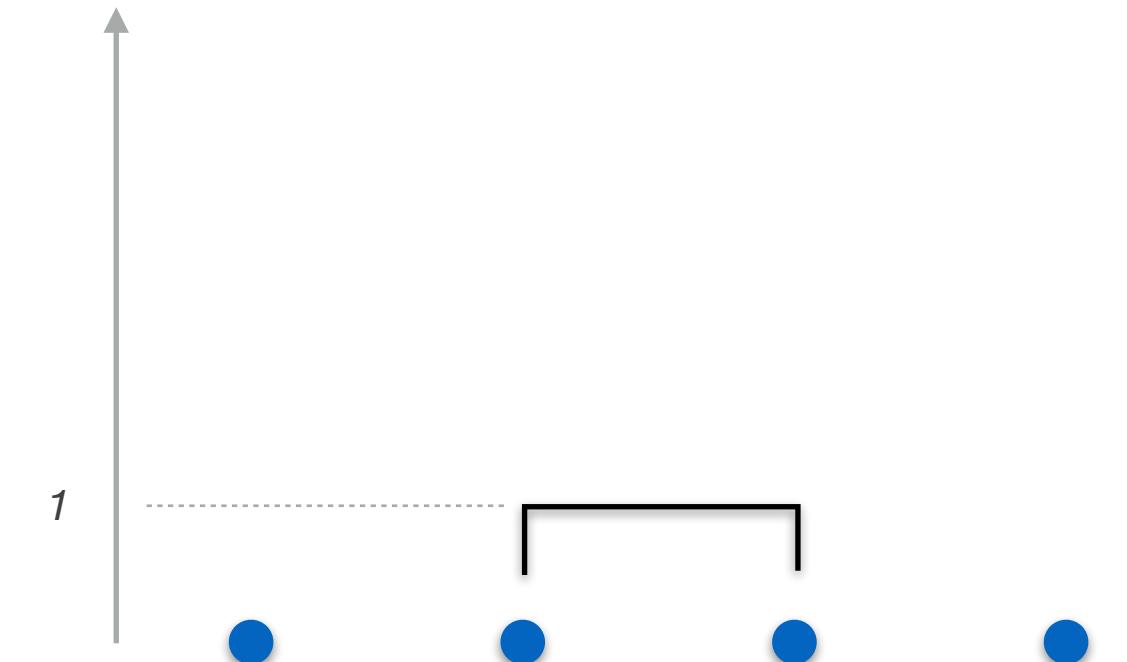
	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0



	a	(b,c)	d
a	0	3	5
(b,c)		0	4.1
d			0

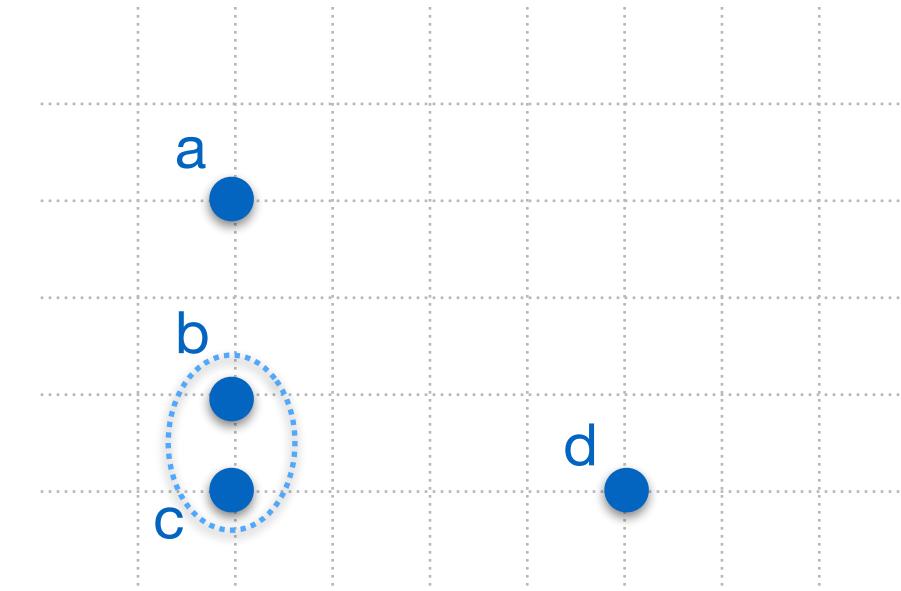


	(a,(b,c))	d
(a,(b,c))	0	5
d		0

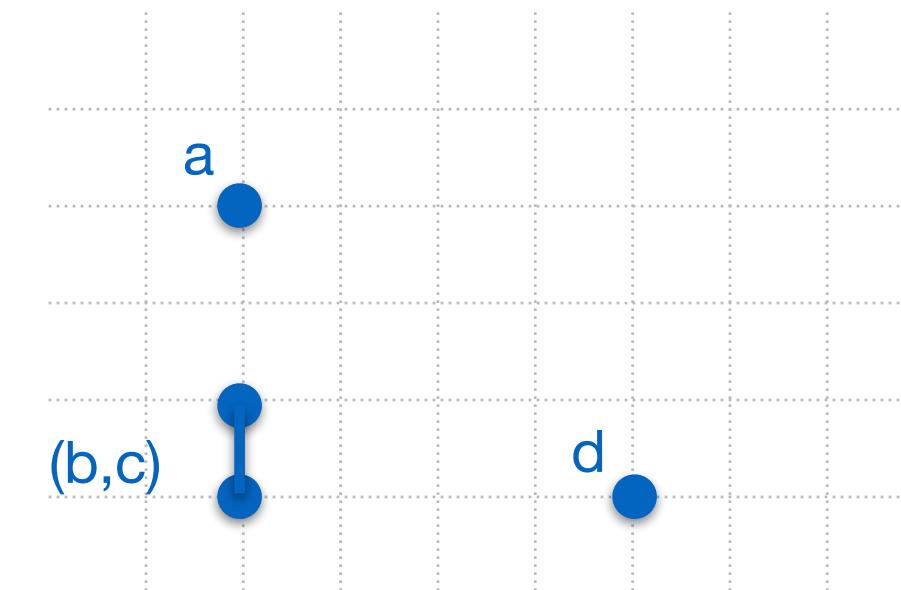


Hierarchical clustering

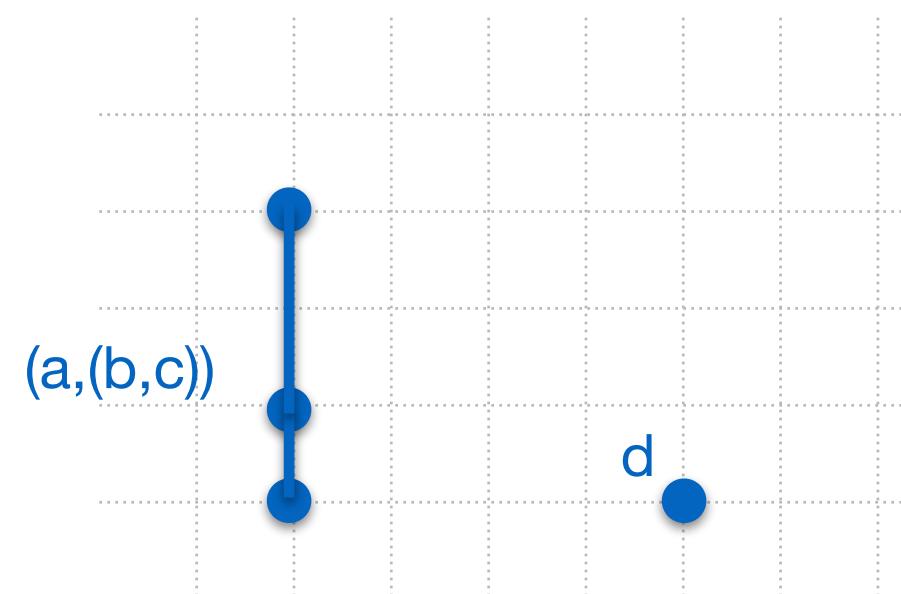
(average-linkage)



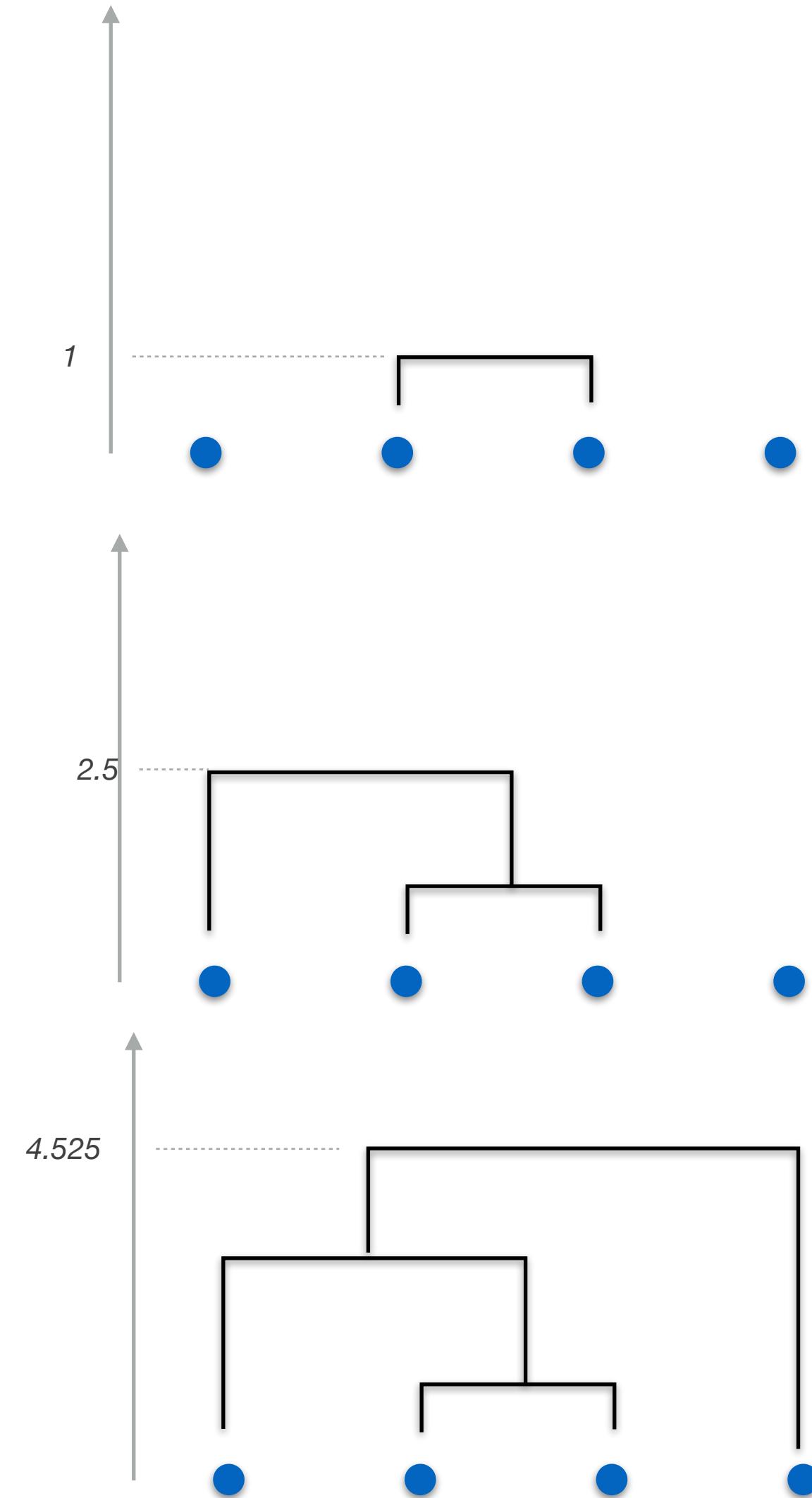
	a	b	c	d
a	0	2	3	5
b		0	1	4.1
c			0	4
d				0



	a	(b,c)	d
a	0	2.5	5
(b,c)		0	4.05
d			0



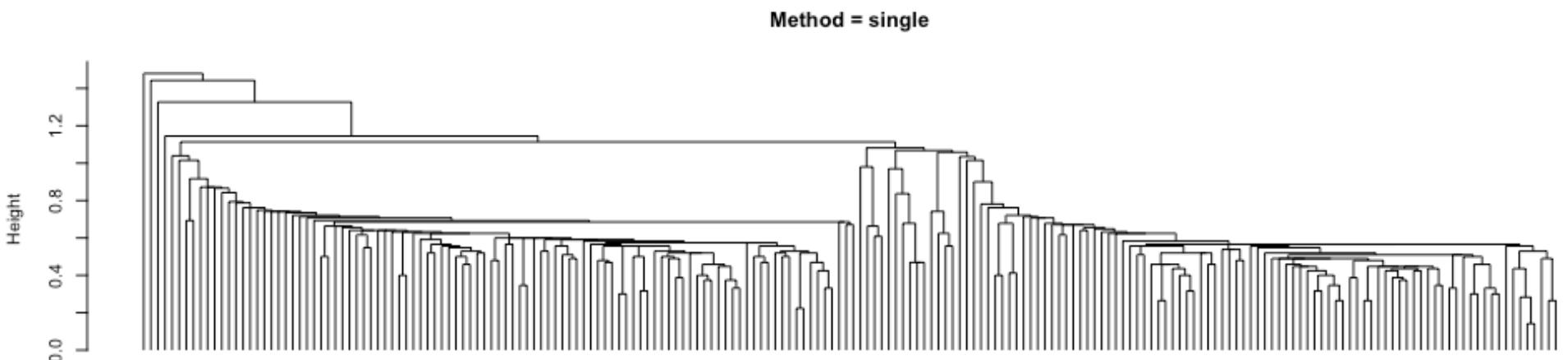
	(a,(b,c))	d
(a,(b,c))	0	4.525
d		0



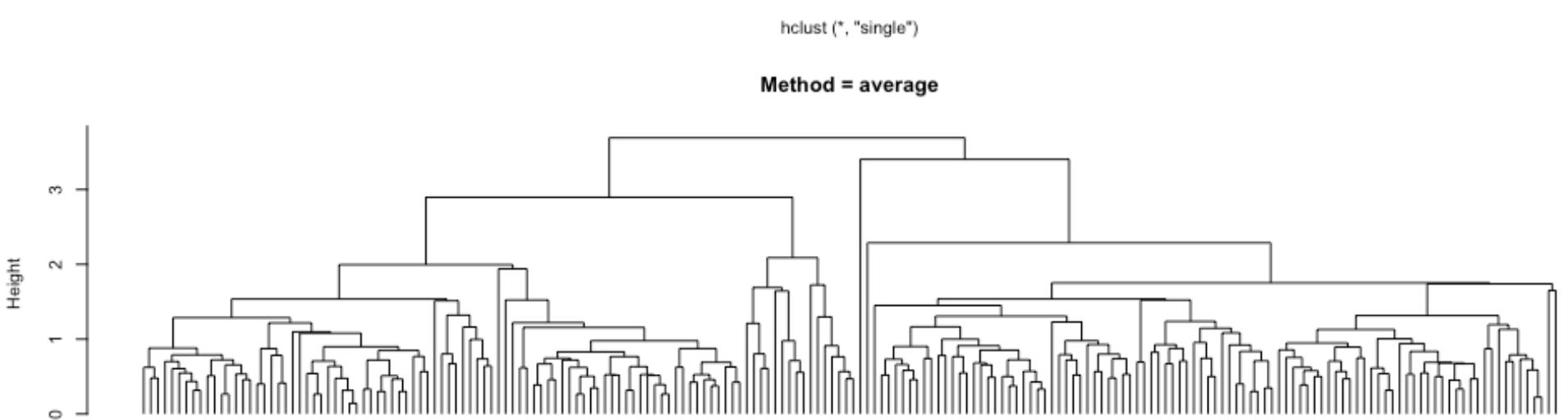
Effect of linkage

- Further linkage methods

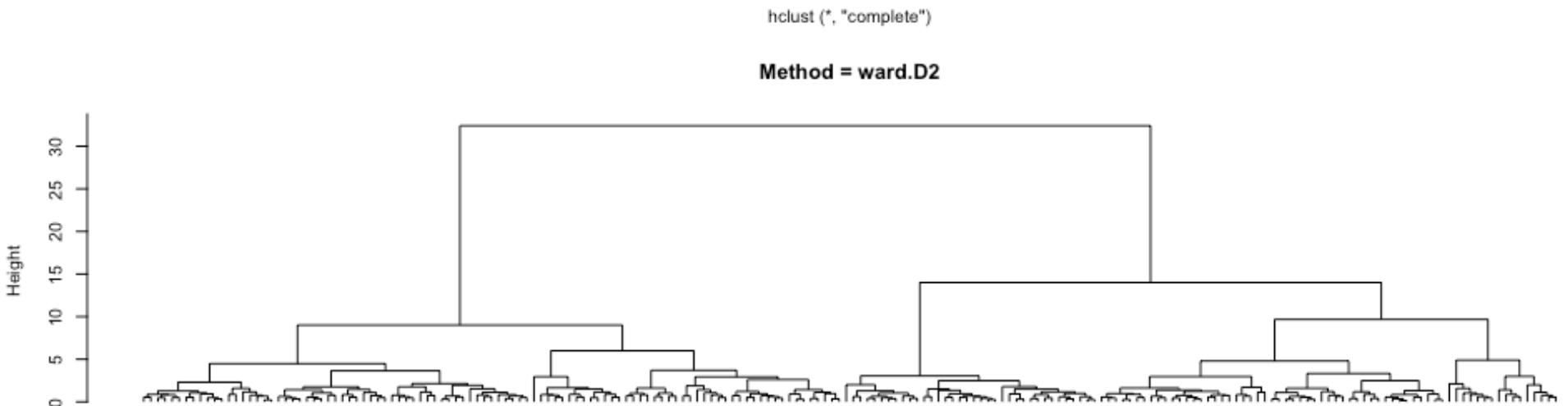
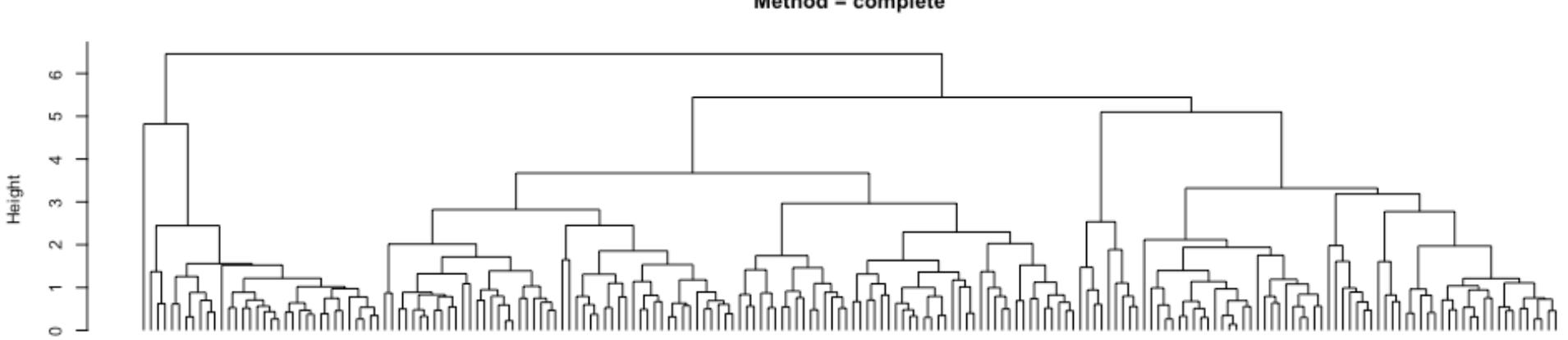
- **centroid-linkage:** distance between the clusters is given by the distance to the centers of the clusters



- **Ward linkage:** Distance between two clusters is equal to the increase of the variance of the combined cluster compared to individual clusters

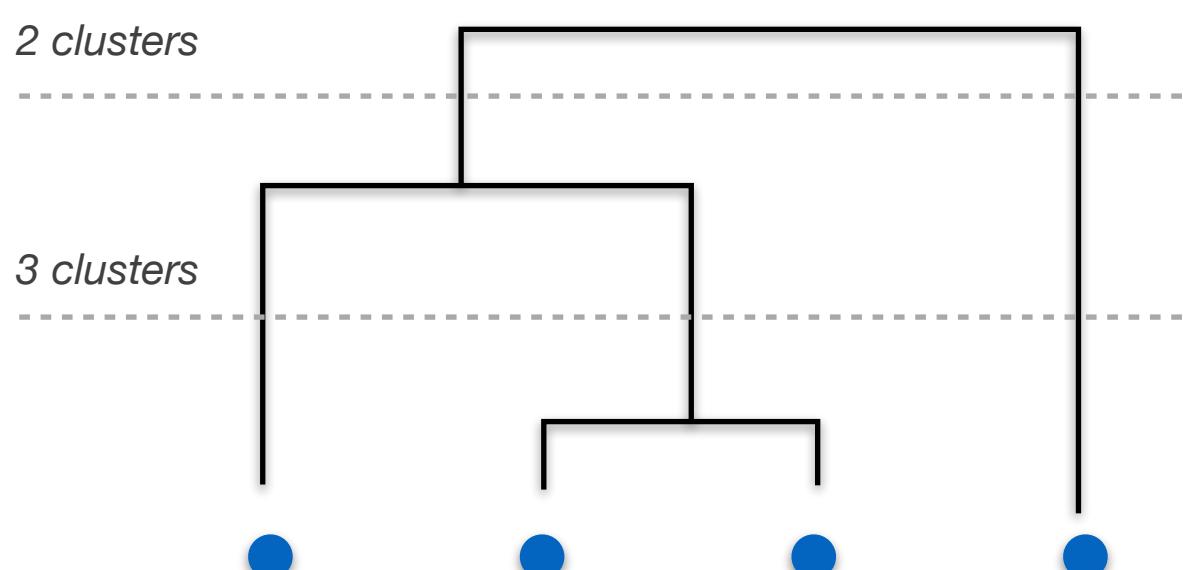


- ...

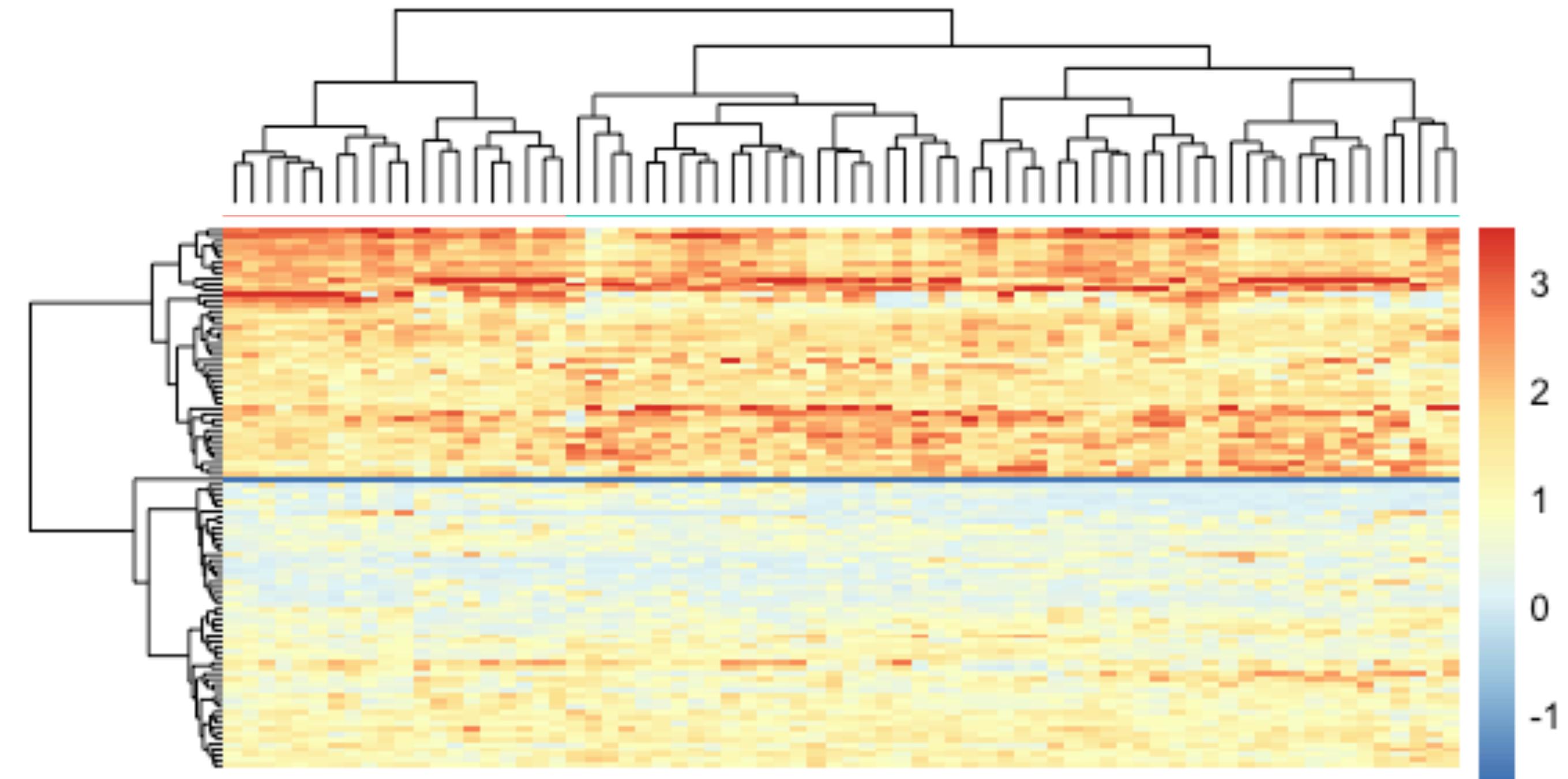


Hierarchical clustering

- Dendrogram (= rooted tree diagram)
- height of a branch = distance between the clusters it joins
- **Different linkage methods** can lead to **different tree topologies!**
- **Different similarity measures** can lead to **different tree topologies!**
- Tree does not yet define clusters:
tree can be cut at different branch height to define clusters
- Where to cut? How many clusters?
→ see criteria discussed previously!
(elbowplot, silhouette,...)



Real world example: Clustering patients

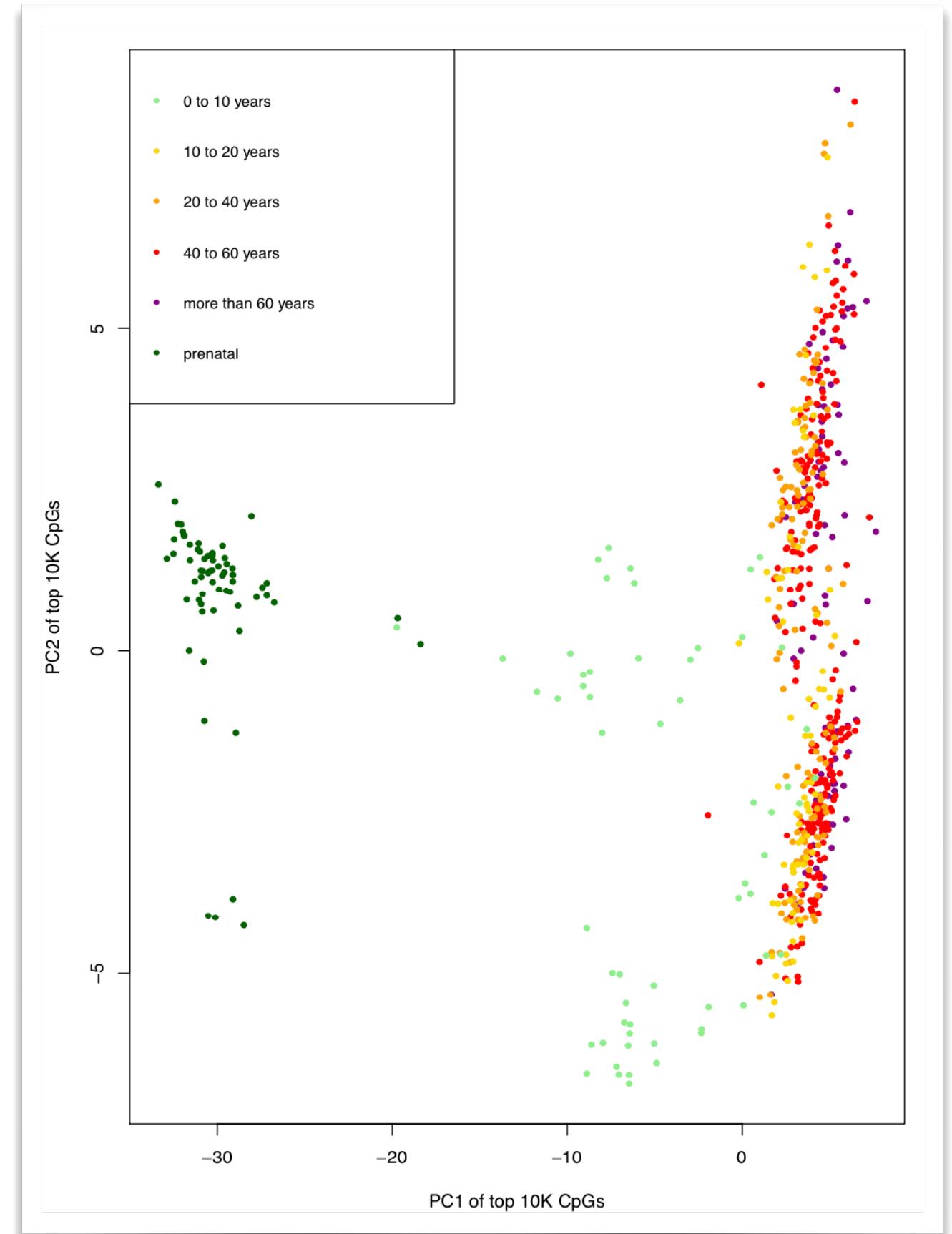


4. finding structure in the data

Principal Component Analysis

Reducing dimensions

- Dataset have a very high dimensionality (e.g. number of genes)
- Need to reduce this large number of dimensions to a smaller number of relevant variables
- Relevant variables = variables which carry most of the information (or variance) of a dataset
- Goal:
 - identify **directions** in the data corresponding to **biological effects**

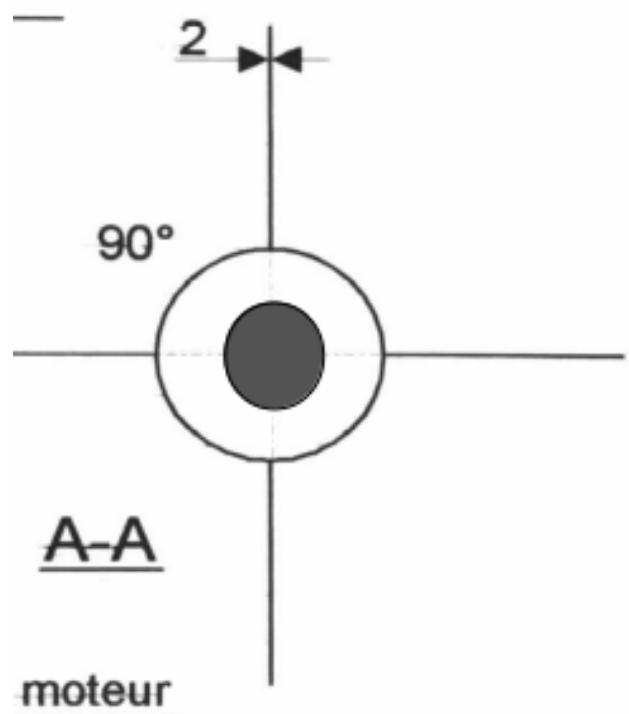


Example of DNA methylation of blood samples
in patient cohort (Jana Dalhoff)
data matrix : 400.000 CpG positions / 250 patients

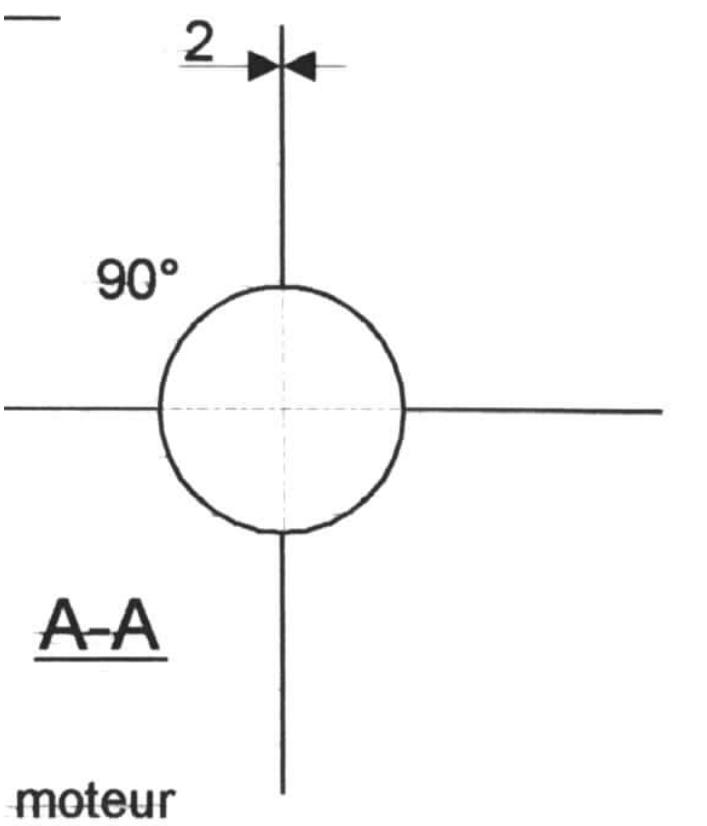
3d → 2d

- How to draw (= 2d) a rocket (= 3d) ?

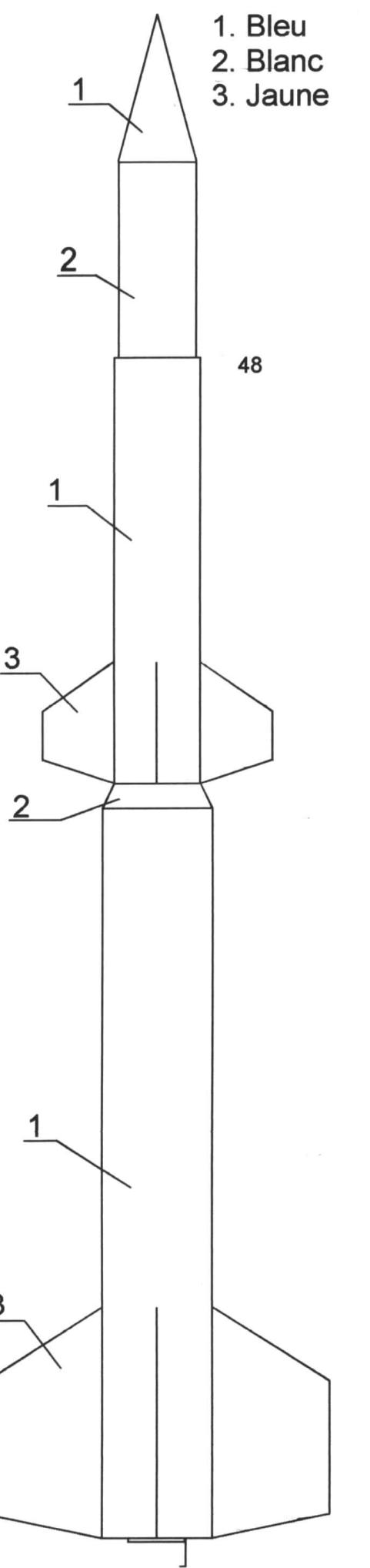
from above



from below



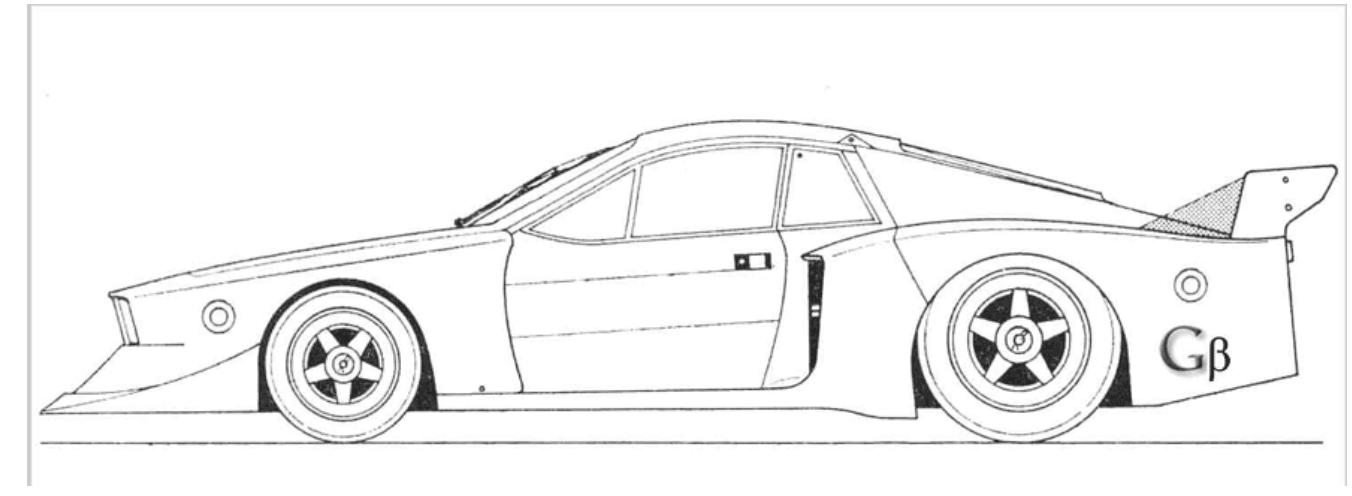
from the side



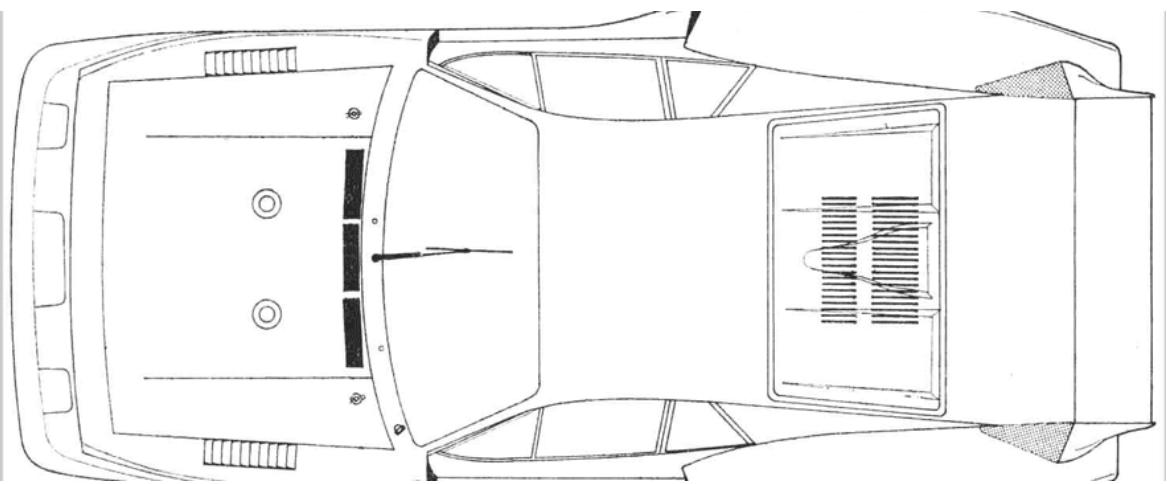
3d → 2d

- How to draw (= 2d) a car (= 3d) ?

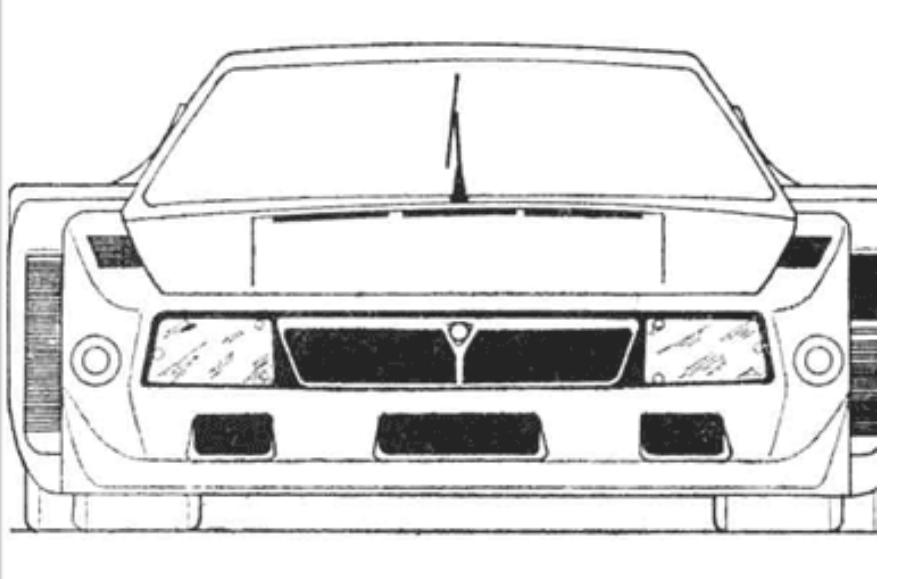
from the side



from above

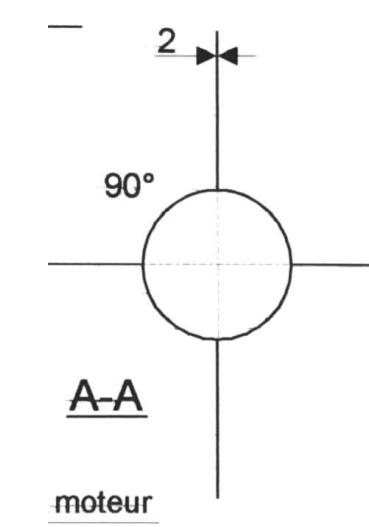
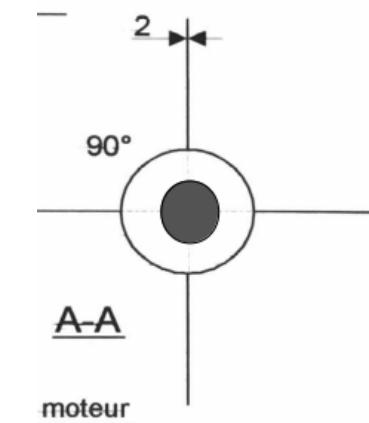


from the front



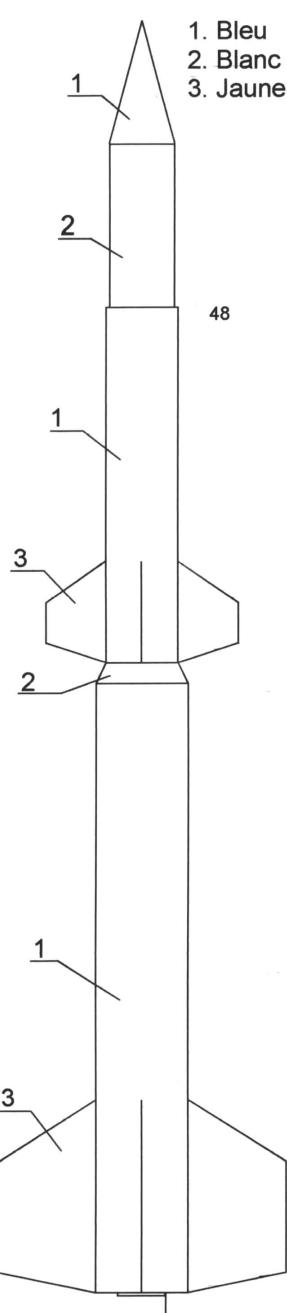
How to choose?

10% information

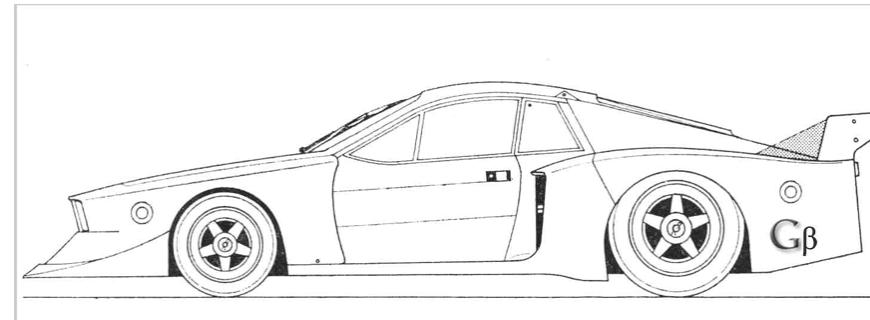


10% information

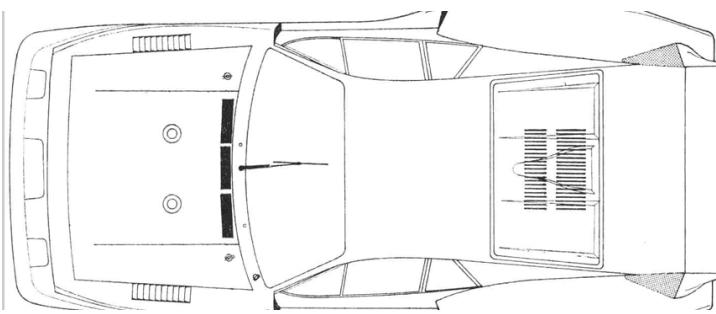
80% information



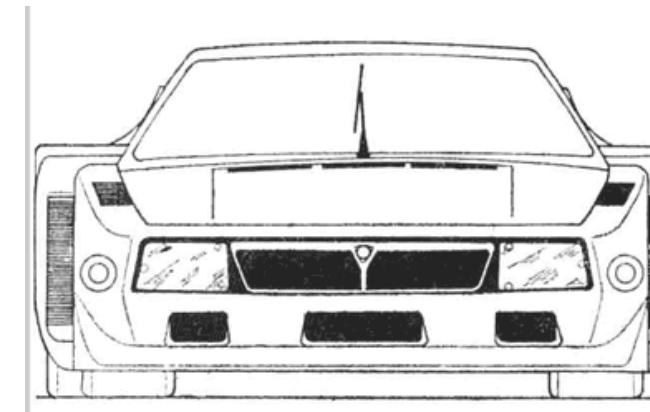
45% information



30% information



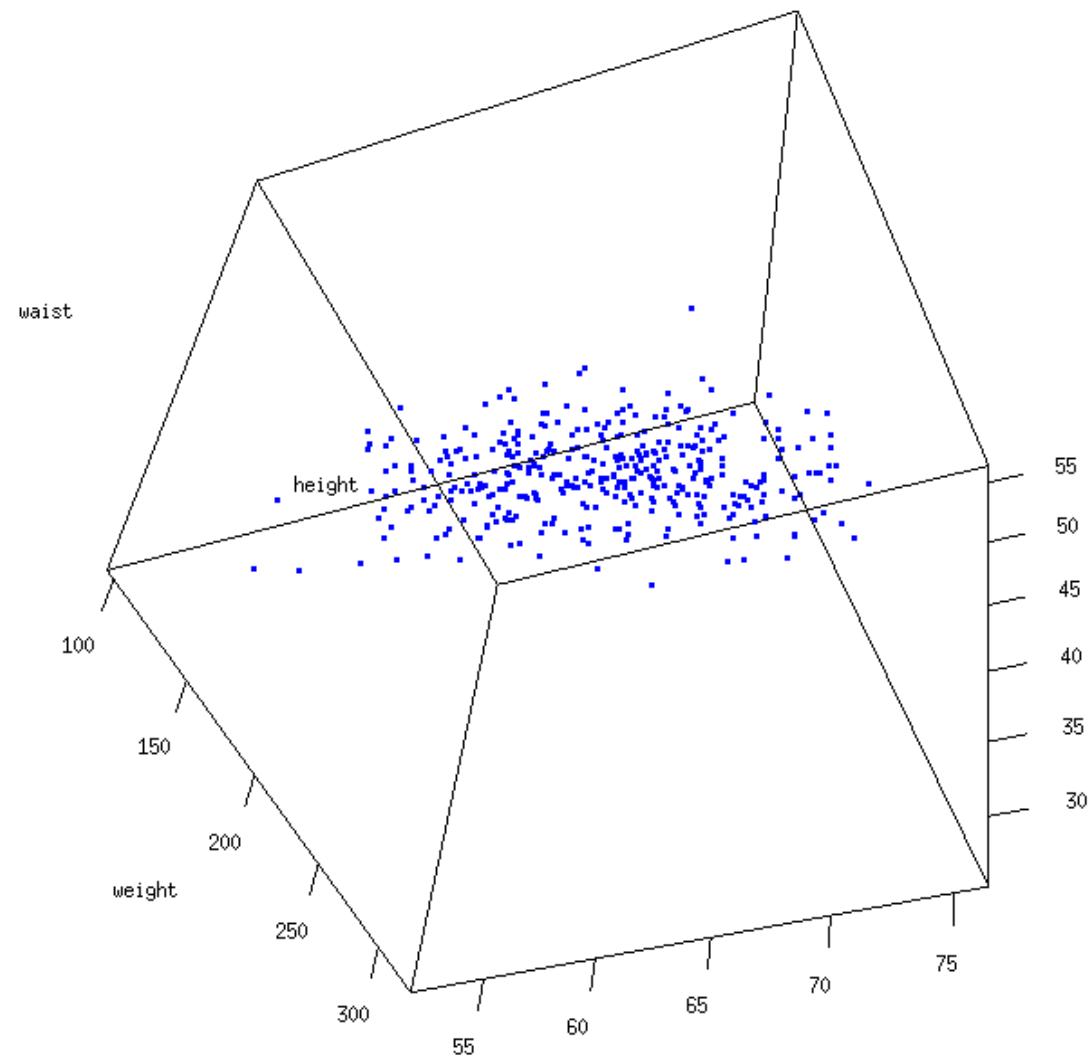
25% information



This is principal component analysis!

Principal Component Analysis

- Idea of principle component analysis: **reduce a multi-dimensional dataset to a lower number of informative dimensions** (“Principal Components”)
- Example diabetes dataset with chol / age / height / weight / waist
- each patient is a dot in a 5d space
→ how to choose new coordinate

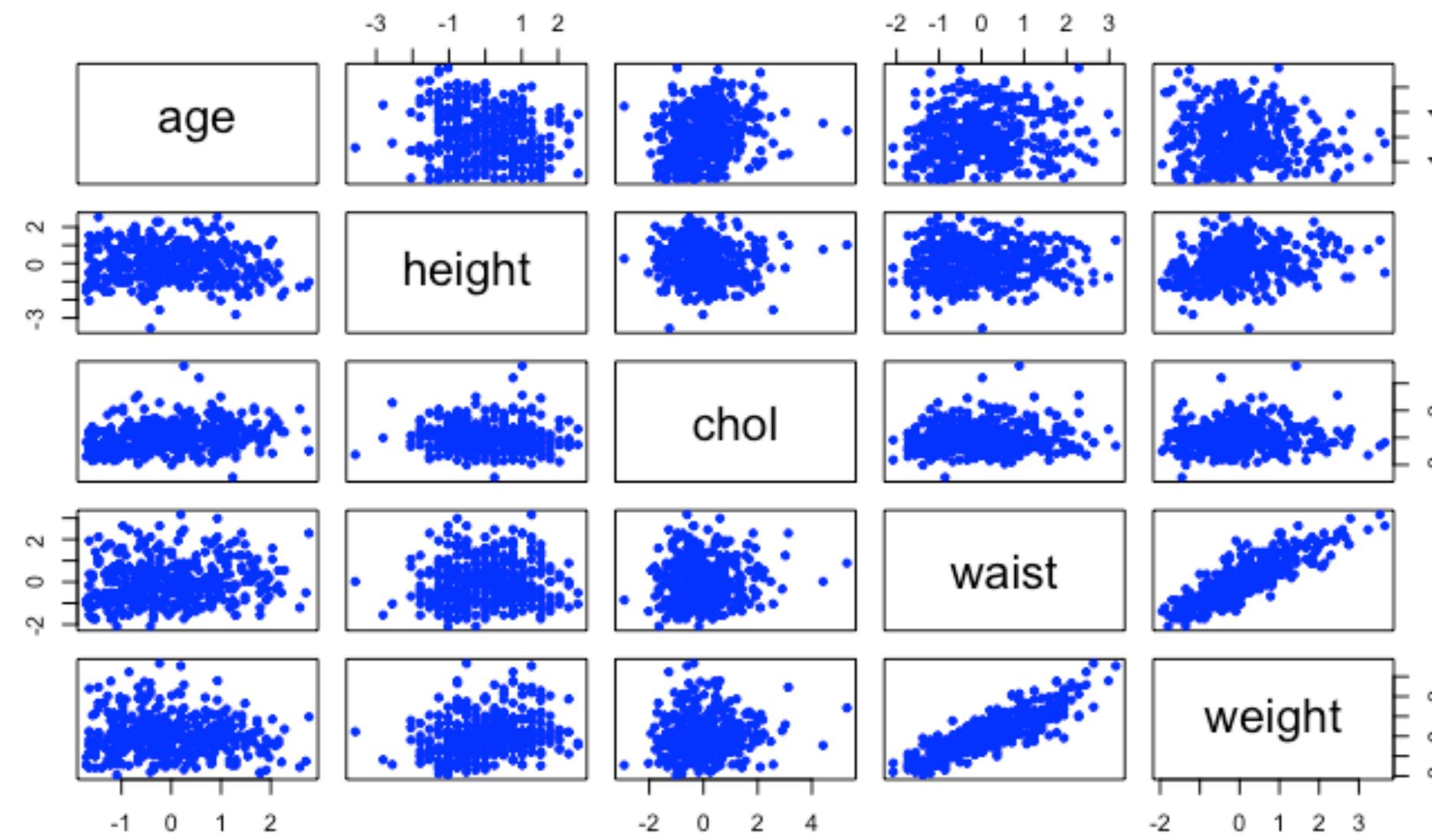


id	chol	age	height	weight	waist
1000	203	46	62	121	29
1001	165	29	64	218	46
1002	228	58	61	256	49
1003	78	67	67	119	33
1005	249	64	68	183	44
1008	248	34	71	190	36
1011	195	30	69	191	46
1015	227	37	59	170	34
1016	177	45	69	166	34
1022	263	55	63	202	45
1024	242	60	65	156	39
1029	215	38	58	195	42
1030	238	27	60	170	35
1031	183	40	59	165	37
1035	191	36	69	183	36
1036	213	33	65	157	37
1037	255	50	65	183	37

k=17 patients, n=5 variables

Correlation structure

- Some variables are **highly correlated** : knowing one gives a lot of information about the second
- Some are **unrelated** : no information gain from one over the other



	age	height	chol	waist	weight
age	1.000	-0.095	0.240	0.153	-0.063
height	-0.095	1.000	-0.059	0.057	0.253
chol	0.240	-0.059	1.000	0.112	0.059
waist	0.153	0.057	0.112	1.000	0.850
weight	-0.063	0.253	0.059	0.850	1.000



Correlation structure

- if two variables are **strongly correlated**, they are partly redundant
→ having 2 variables does not add much information w.r.t. a single variable
- if two variables have **little correlation**, each variable carries information not contained in the other
→ we need to keep these 2 variables to have information about the dataset

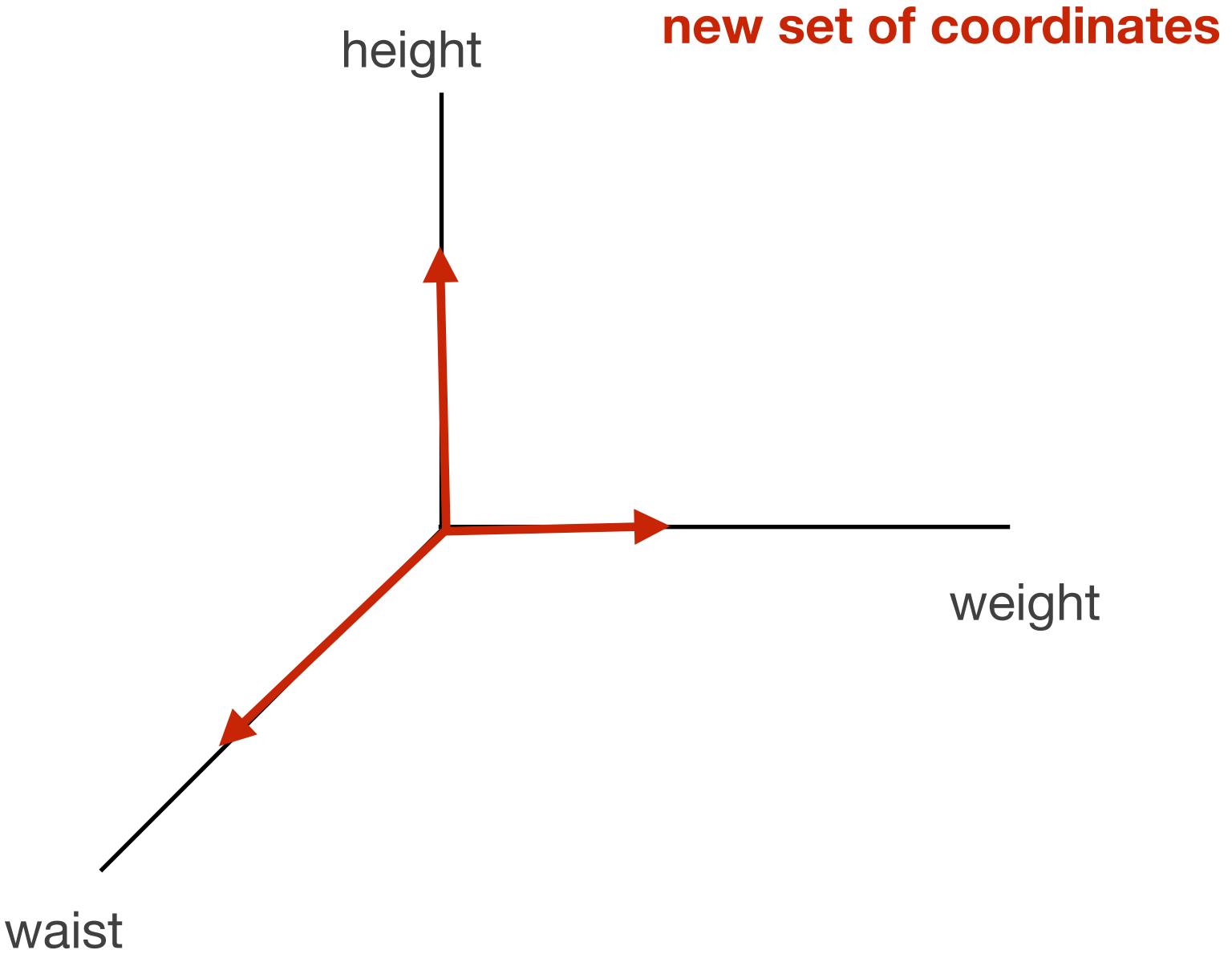


The more diagonal a correlation matrix is, the more information is revealed by the variables

	age	height	chol	waist	weight
age	1.000	-0.095	0.240	0.153	-0.063
height	-0.095	1.000	-0.059	0.057	0.253
chol	0.240	-0.059	1.000	0.112	0.059
waist	0.153	0.057	0.112	1.000	0.850
weight	-0.063	0.253	0.059	0.850	1.000

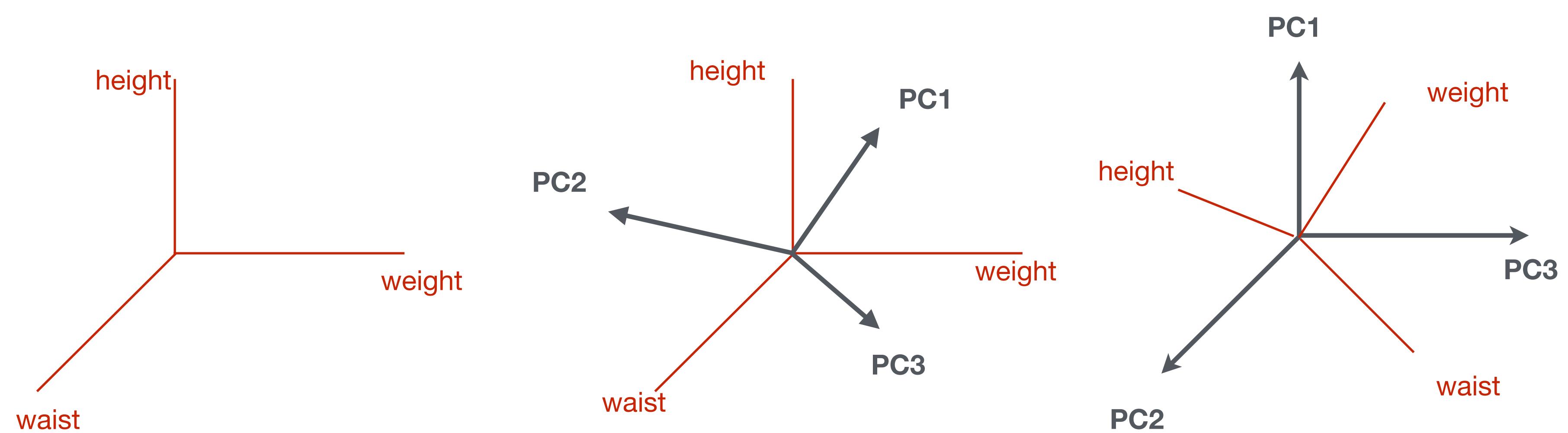
Defining new variables

- We want to express the dataset in a **new set of coordinates**
- for each of these coordinate systems, the correlation matrix will change
- **Goal:** find the rotation that makes the correlation matrix diagonal



Hello, matrix diagonalization !

New coordinate system

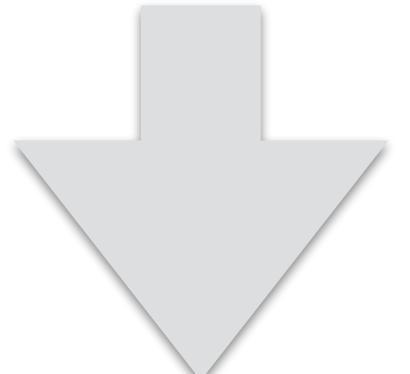


New coordinate system

patients



	age	height	chol	waist	weight
[1,]	-0.056039494	-1.023316184	-0.118183417	-1.563795253	-1.418763531
[2,]	-1.090068507	-0.514242224	-0.978388626	1.405952579	0.990046092
[3,]	0.673863338	-1.277853164	0.447741063	1.930025726	1.933703470
[4,]	1.221290463	0.249368716	-2.947805816	-0.865031057	-1.468429708
[5,]	1.038814754	0.503905697	0.923117626	1.056570481	0.120887981
[6,]	-0.785942327	1.267516637	0.900480647	-0.340957910	0.294719603
[7,]	-1.029243271	0.758442677	-0.299279250	1.405952579	0.319552692
[8,]	-0.603466618	-1.786927124	0.425104084	-0.690340008	-0.201942175
[9,]	-0.116864730	0.758442677	-0.706744876	-0.690340008	-0.301274530
[10,]	0.491387630	-0.768779204	1.240035335	1.231261530	0.592716670



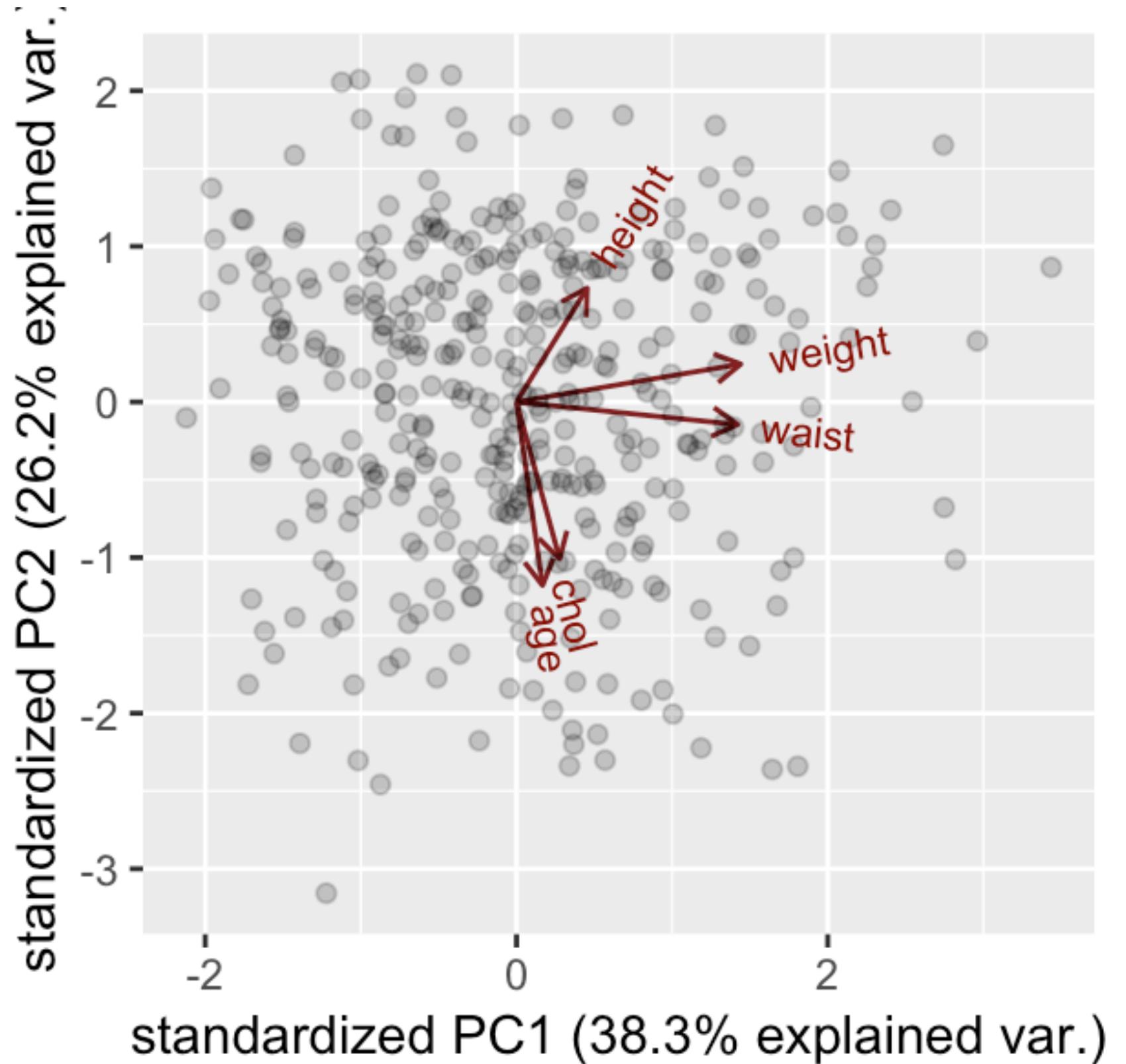
patients



	PC1	PC2	PC3	PC4	PC5
[1,]	-2.2724017805	-0.394780391	-0.37314376	0.141536224	0.2090433700
[2,]	1.3055985070	1.119481681	-1.50495414	0.143832964	-0.3513275630
[3,]	2.4741433771	-1.132974842	-1.49880577	0.037584884	0.3143758721
[4,]	-1.8399173774	0.920620520	-0.30773345	-2.959165349	-0.2523118706
[5,]	1.1108365939	-1.098650879	0.71155625	-0.241751636	-0.5571448997
[6,]	0.3019126963	0.582590104	1.33455719	1.004709292	0.1312818611
[7,]	1.2128762055	1.111191009	-0.06294882	0.315416477	-0.9968437164
[8,]	-0.9786990772	-0.575259493	-1.28325003	1.043051917	0.4815398584
[9,]	-0.6156836153	0.831267656	0.56093836	-0.500214895	0.1441444663
[10,]	1.2805031592	-1.405669315	-0.45963453	0.587442537	-0.2560111460

PCA biplot

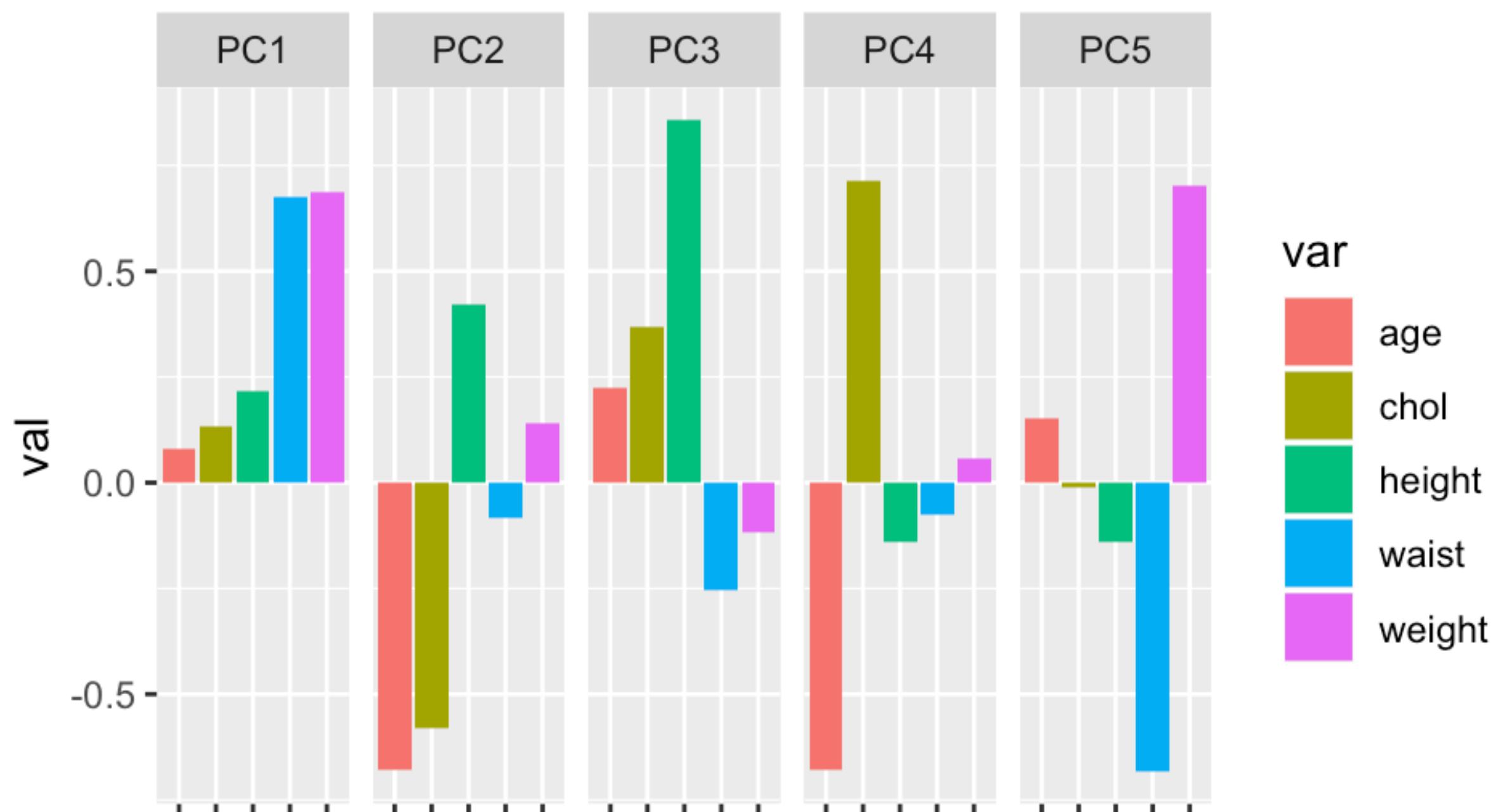
- each **dot** is a sample / patient
- new coordinate system is (PC_1, PC_2)
- Red arrows** indicate the contribution of each “old” coordinate to the PCs



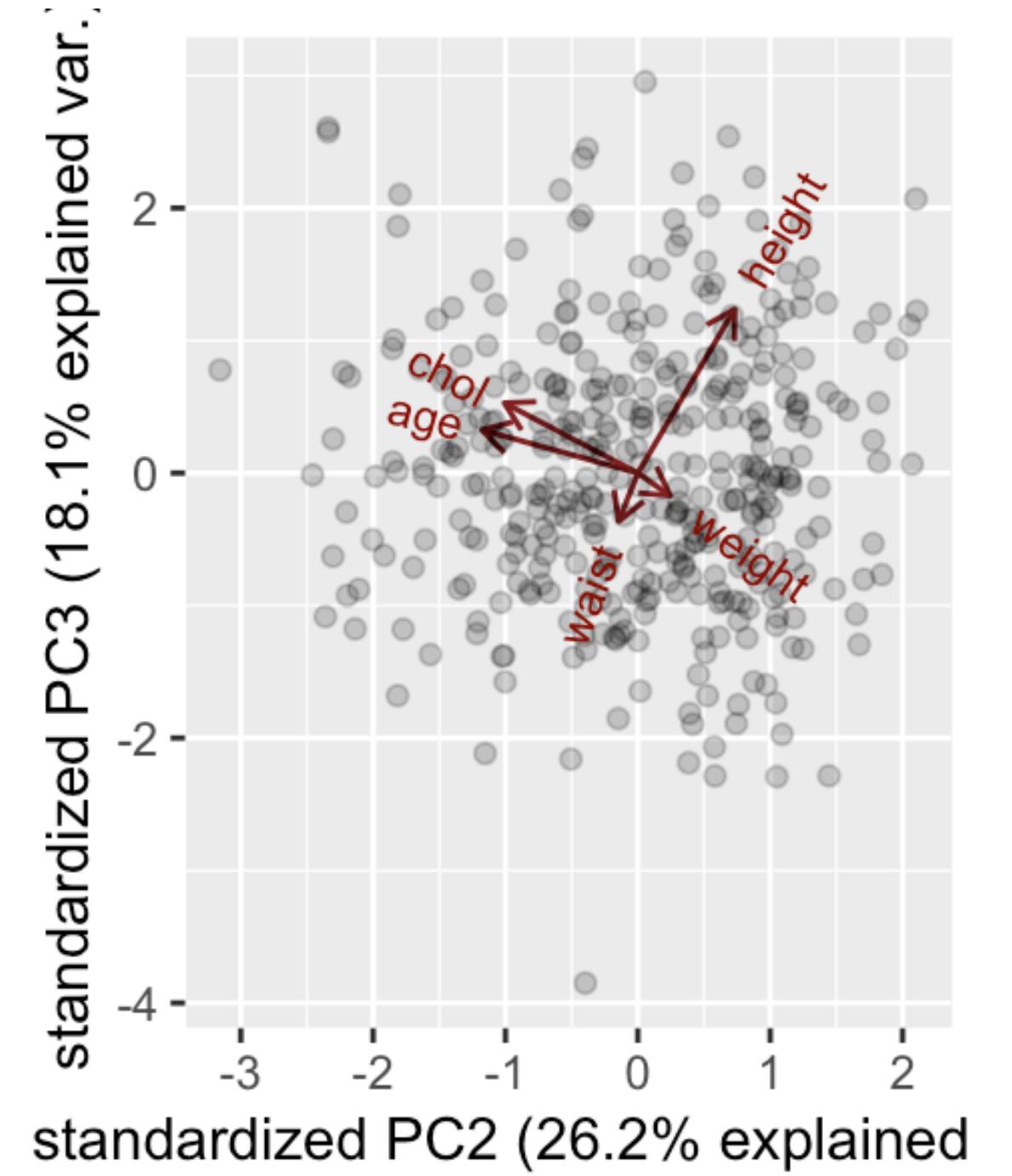
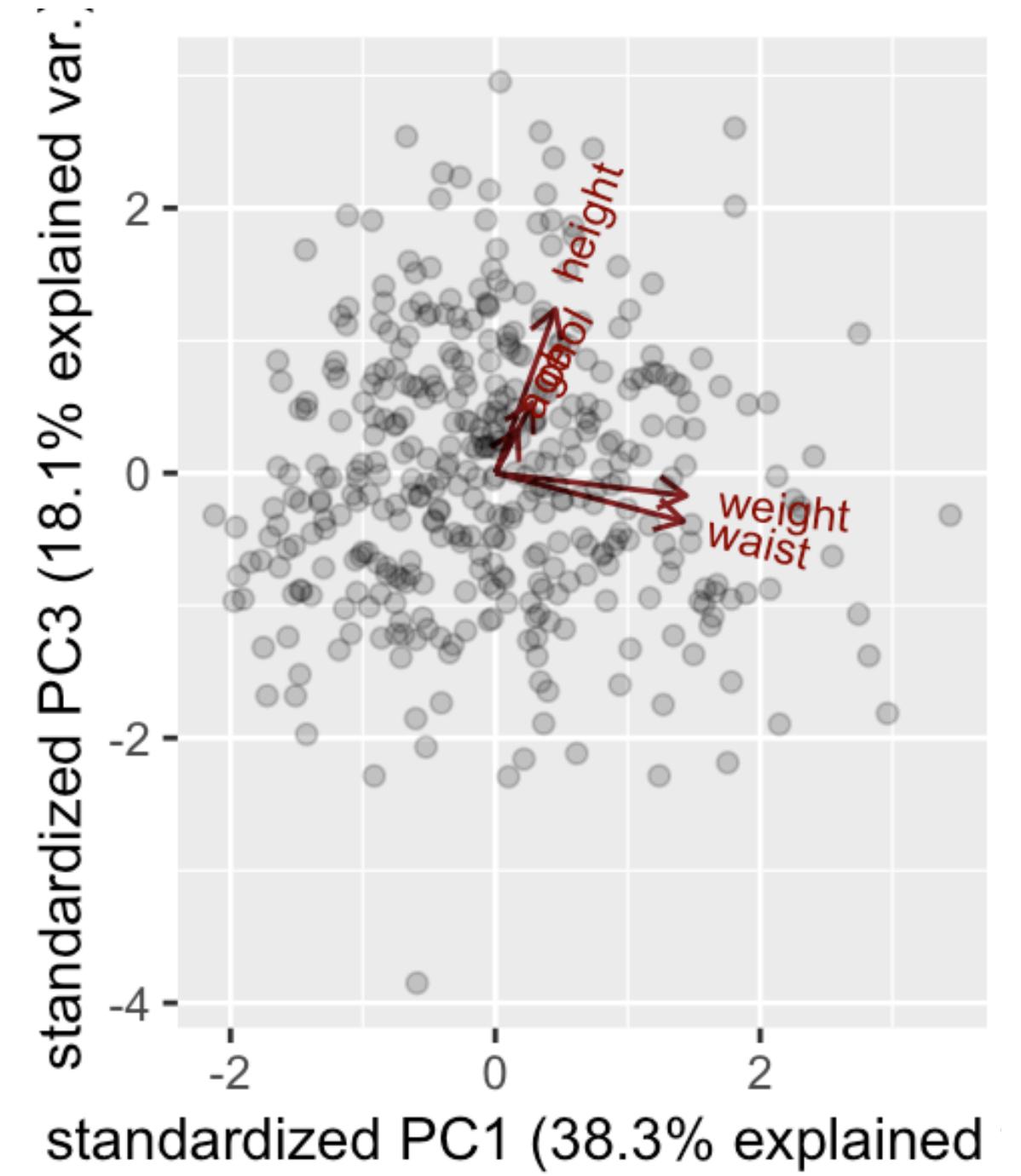
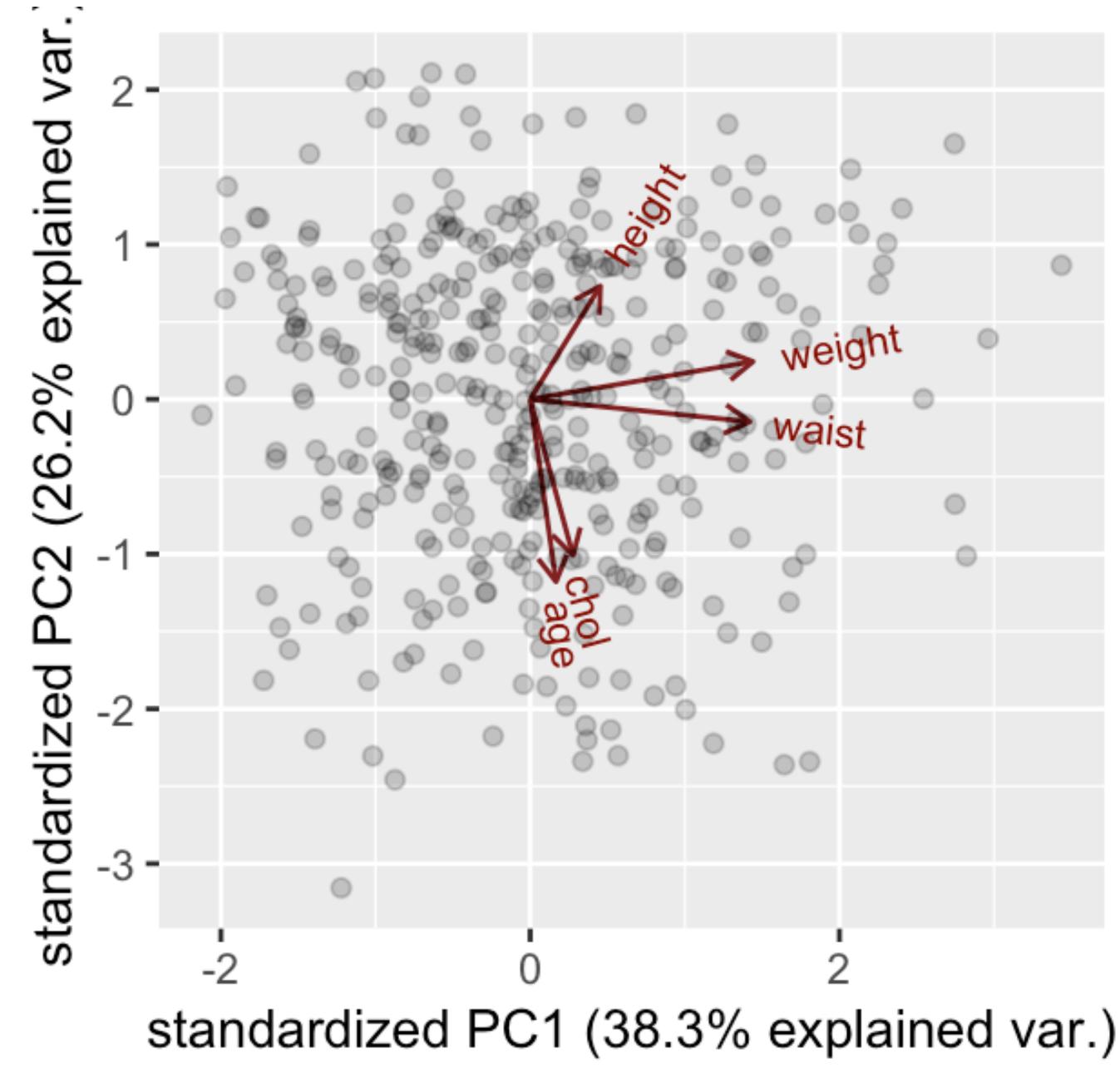
Principal components

$$PC_i = \alpha_i \cdot \text{age} + \beta_i \cdot \text{chol} + \gamma_i \cdot \text{height} + \delta_i \cdot \text{waist} + \epsilon_i \cdot \text{weight}$$

- contribution of each variable to the principal components (coefficients are called "*loadings*")
- some variables contribute in the same direction to some PCs (e.g. waist and height for PC1), but opposite to others (PC5)

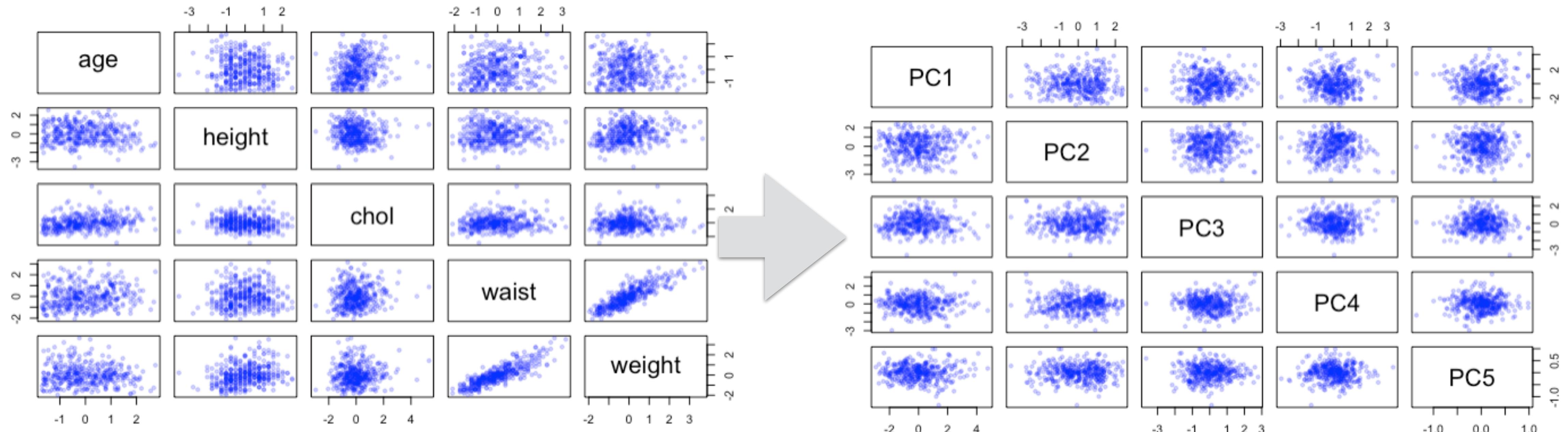


Principal components



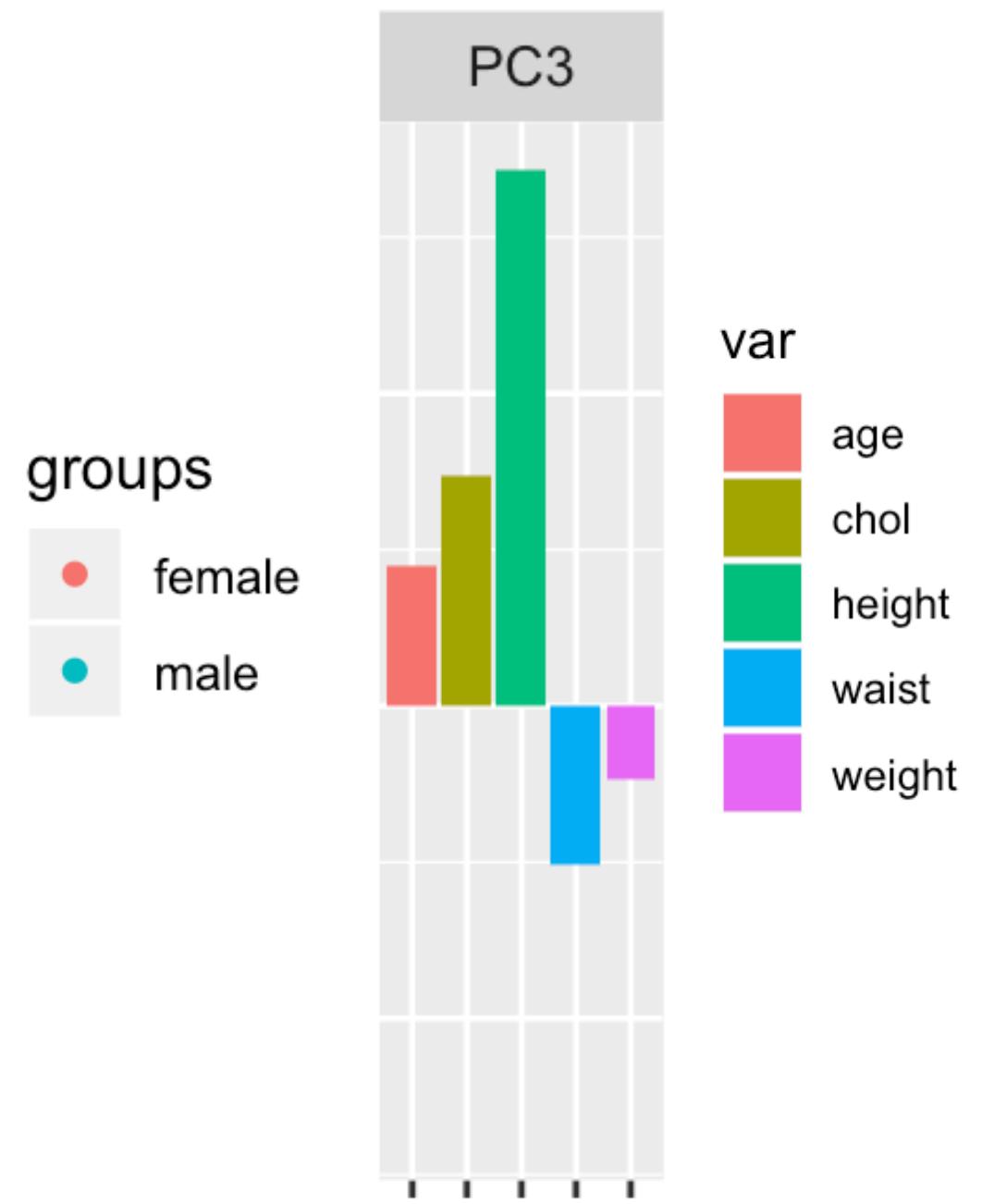
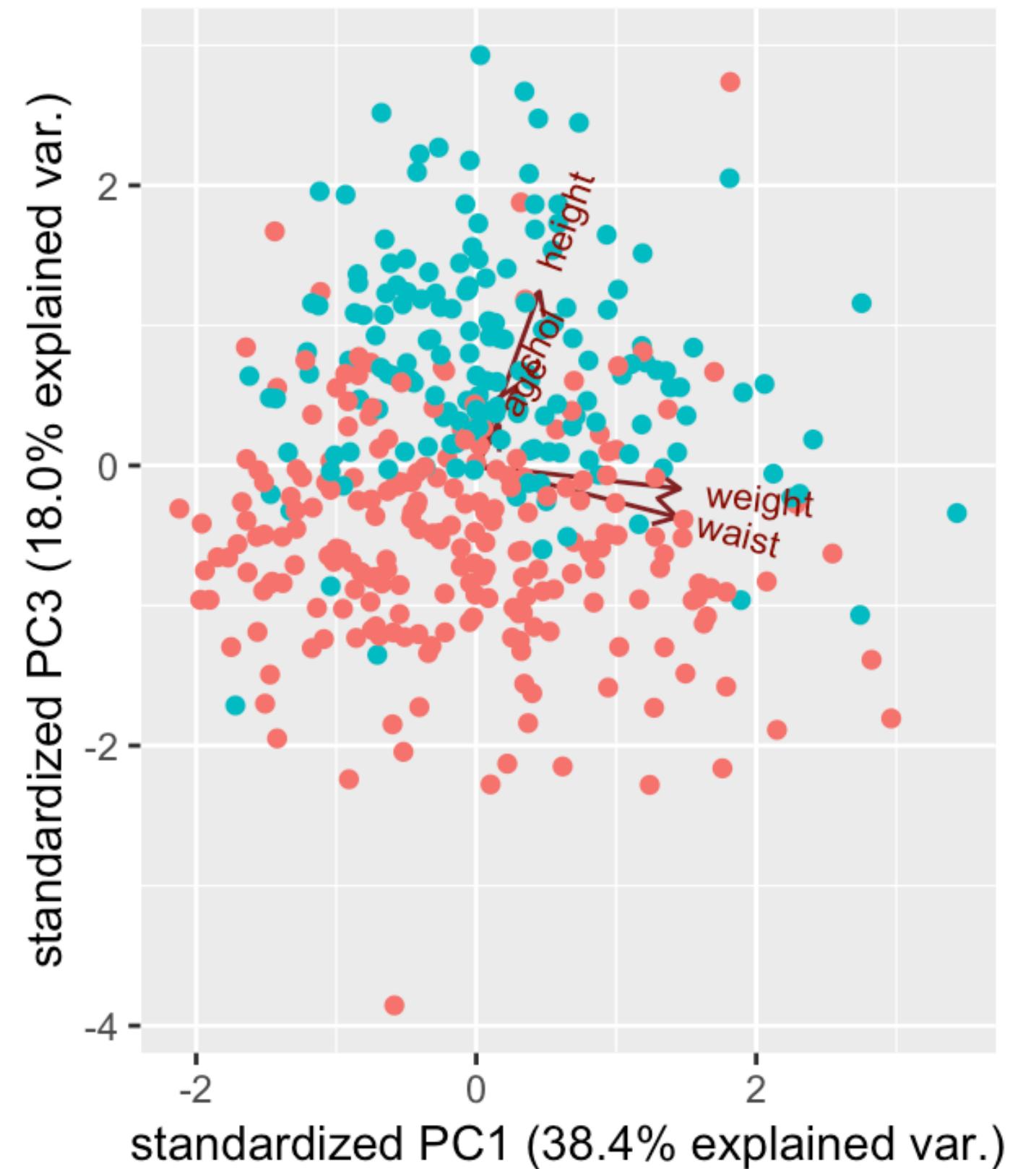
Principal components

- by construction, **Principal Components have no correlation to each other!**



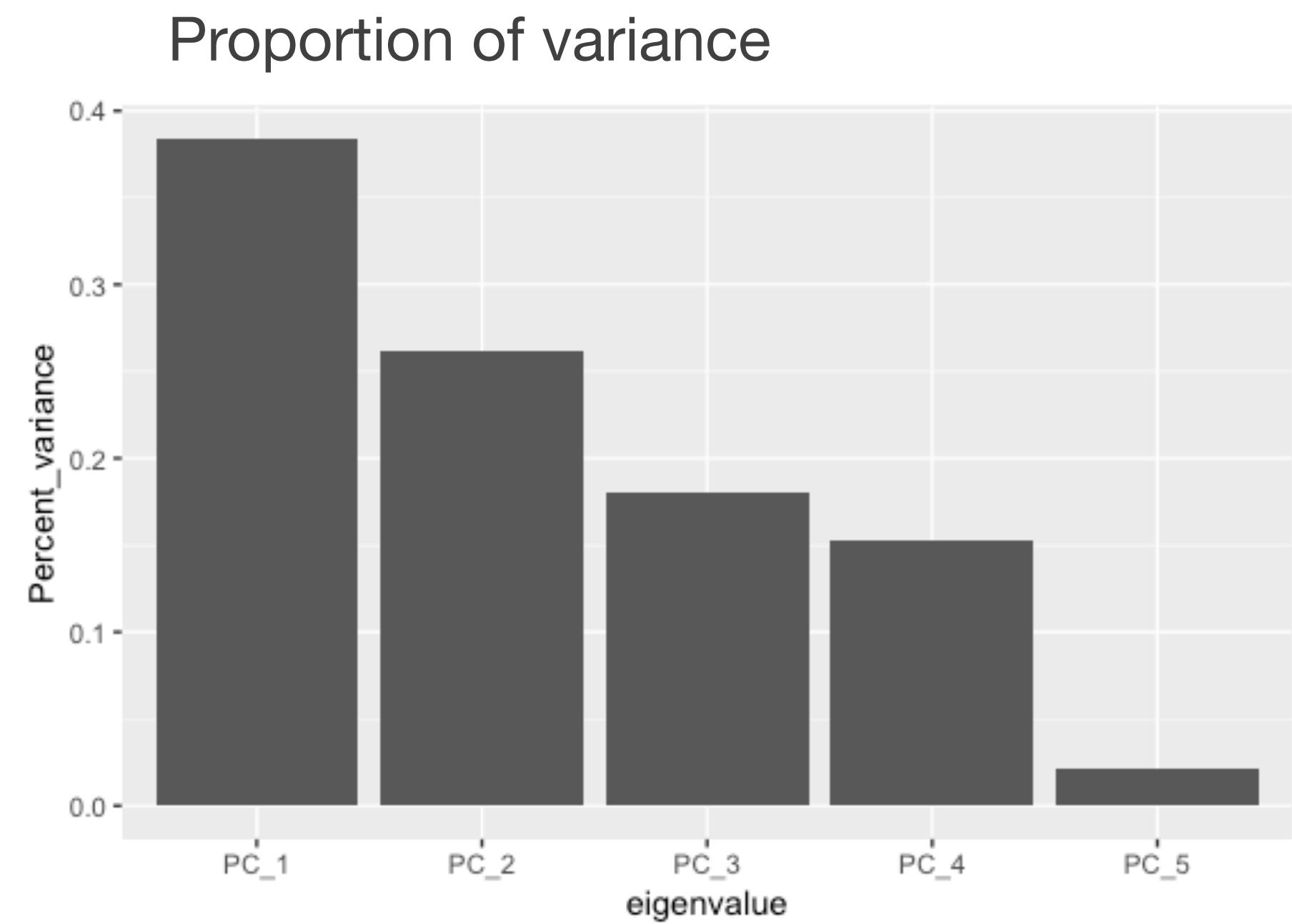
Identifying interesting PCs

- PC plots can highlight a new group structure
- Example: **PC3** seems very associated to gender
- indicates that a combination of height and cholesterol does separate men / women



Number of PCs?

- Each PC explains some part of the **total variance** of the dataset
- PCs are ordered by **decreasing explained variance**



*Considering PC1 & PC2 explains
63% of the total variance*

Choosing the number of PCs

- several criteria to select the optimal subset of PCs, without loosing too much information

- Proportion of variance:**
keep PCs such that the cumulative variance is above threshold

$$\sum_{i=1}^k \frac{\lambda_i}{\sum \lambda_i} \geq \text{var}_{min}$$

- Average eigenvalue criteria:**
keep PCs which have eigenvalue larger than
 - mean eigenvalue (Kaiser rule) or
 - 70% of mean eigenvalue (Jottcliffe rule)

$$\lambda_i \geq \bar{\lambda}$$