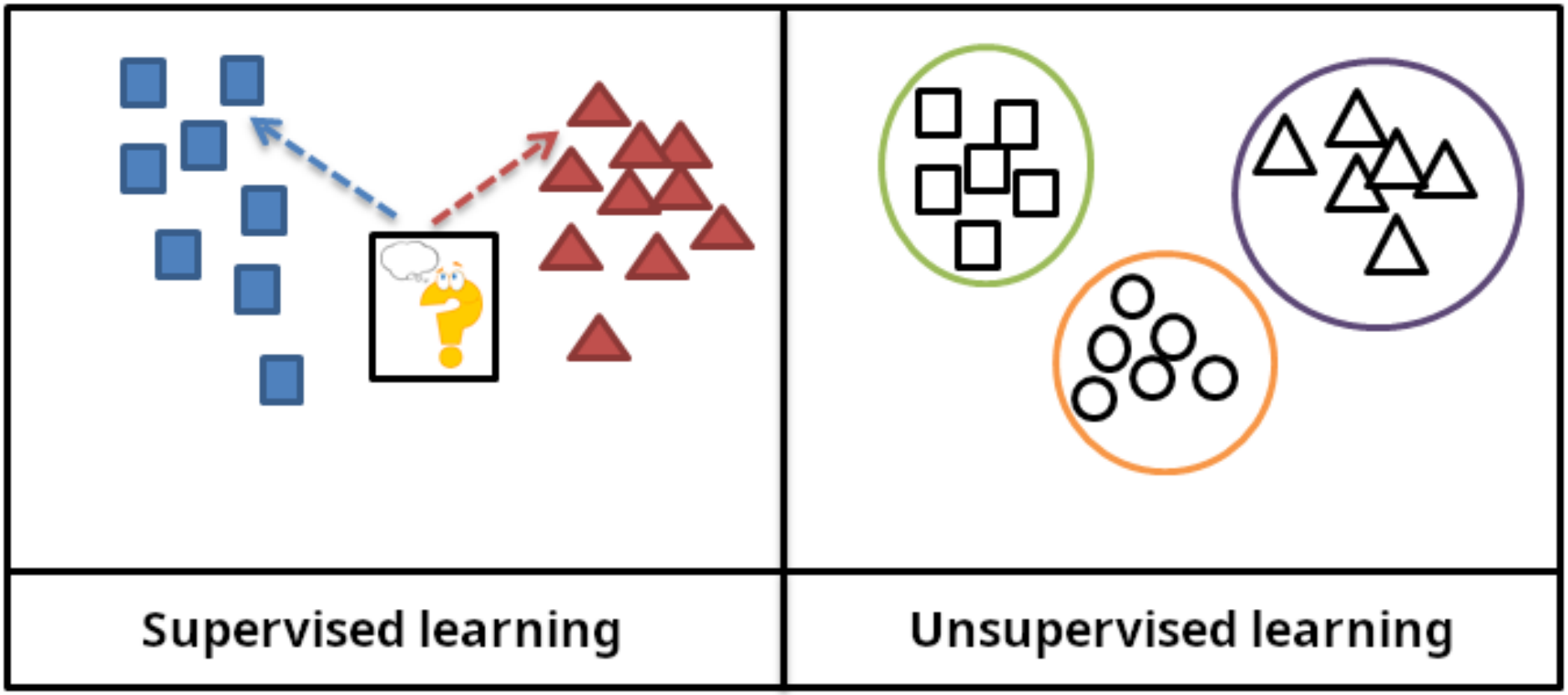
Introduction to Machine Learning

Soumya Banerjee

Types of Machine Learning



Slide courtesy Paul Fannon

Supervised Learning - Regression

Student	ML Trained?	Cohort %ile	Salary
0001	Y	12	120000
0002	Y	58	42000
0003	N	22	36000
0004	Y	58	35000
…	…	…	…
1784	Y	34	???

Slide courtesy Paul Fannon

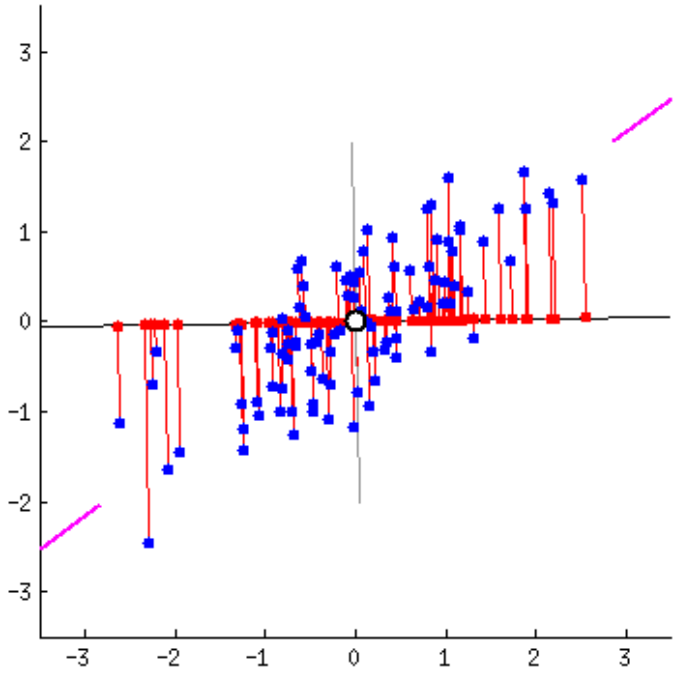
Unsupervised methods

- History
- Exploratory data analysis
- Hypothesis generation
- Data visualization
- Data imputation
- Detecting outliers

High-dimensional data

- How to visualize high-dimensional data
- Pairwise

High-dimensional data



https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

Interactive tool

- https://projector.tensorflow.org/

PCA

The *first principal component* of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p \tag{10.1}$$

that has the largest variance. By *normalized*, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the *loadings* of the first principal

loading

PCA

Given a $n \times p$ data set $\mathbf{X}$, how do we compute the first principal component? Since we are only interested in variance, we assume that each of the variables in $\mathbf{X}$ has been centered to have mean zero (that is, the column means of $\mathbf{X}$ are zero). We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \ldots + \phi_{p1}x_{ip} \tag{10.2}$$

that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. In other words, the first principal component loading vector solves the optimization problem

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1. \tag{10.3}$$

From (10.2) we can write the objective in (10.3) as $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$. Since $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$, the average of the $z_{11}, \ldots, z_{n1}$ will be zero as well. Hence the objective that we are maximizing in (10.3) is just the sample variance of the n values of z_{i1}. We refer to $z_{11}, \ldots, z_{n1}$ as the *scores* of the first principal component. Problem (10.3) can be solved via an eigen decomposition, a standard technique in linear algebra, but details are outside of the scope of this book.

score

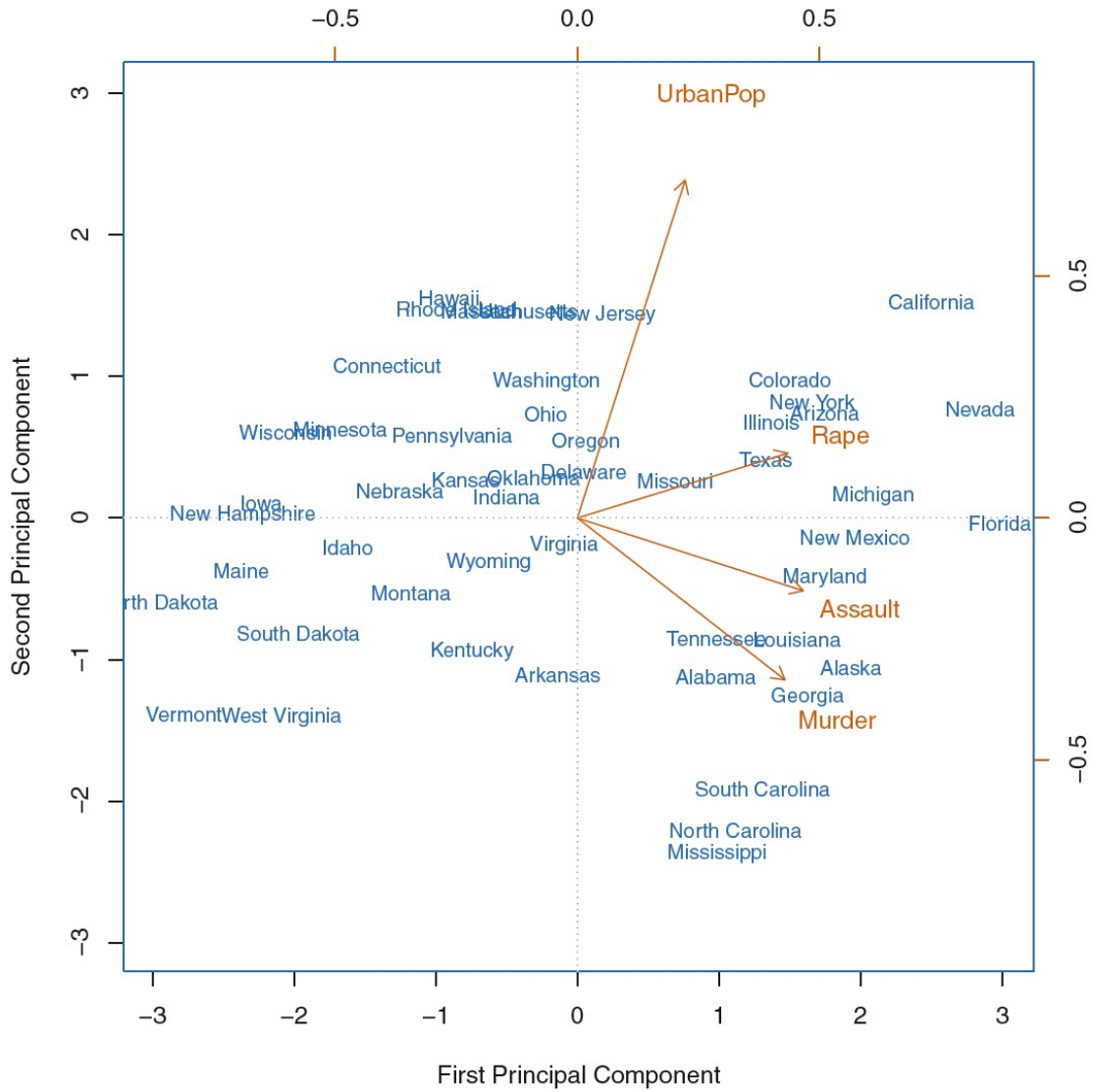


FIGURE 10.1. *The first two principal components for the* `USArrests` *data. The blue state names represent the scores for the first two principal components. The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for* `Rape` *on the first component is* 0.54*, and its loading on the second principal component* 0.17 *(the word* `Rape` *is centered at the point* $(0.54, 0.17)$*). This figure is known as a biplot, because it displays both the principal component scores and the principal component loadings.*

PCA

	PC1	PC2
Murder	0.5358995	−0.4181809
Assault	0.5831836	−0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

TABLE 10.1. *The principal component loading vectors, ϕ_1 and ϕ_2, for the* `USArrests` *data. These are also displayed in Figure 10.1.*

PCA

We illustrate the use of PCA on the `USArrests` data set. For each of the 50 states in the United States, the data set contains the number of arrests per 100,000 residents for each of three crimes: `Assault`, `Murder`, and `Rape`. We also record `UrbanPop` (the percent of the population in each state living in urban areas). The principal component score vectors have length $n = 50$, and the principal component loading vectors have length $p = 4$. PCA was performed after standardizing each variable to have mean zero and standard deviation one. Figure 10.1 plots the first two principal components of these data. The figure represents both the principal component scores and the loading vectors in a single *biplot* display. The loadings are also given in Table 10.1.

biplot

In Figure 10.1, we see that the first loading vector places approximately equal weight on `Assault`, `Murder`, and `Rape`, with much less weight on

PCA

`UrbanPop`. Hence this component roughly corresponds to a measure of overall rates of serious crimes. The second loading vector places most of its weight on `UrbanPop` and much less weight on the other three features. Hence, this component roughly corresponds to the level of urbanization of the state. Overall, we see that the crime-related variables (`Murder`, `Assault`, and `Rape`) are located close to each other, and that the `UrbanPop` variable is far from the other three. This indicates that the crime-related variables are correlated with each other—states with high murder rates tend to have high assault and rape rates—and that the `UrbanPop` variable is less correlated with the other three.

PCA

The first principal component loading vector has a very special property: it is the line in p-dimensional space that is *closest* to the n observations (using average squared Euclidean distance as a measure of closeness). This interpretation can be seen in the left-hand panel of Figure 6.15; the dashed lines indicate the distance between each observation and the first principal component loading vector. The appeal of this interpretation is clear: we seek a single dimension of the data that lies as close as possible to all of the data points, since such a line will likely provide a good summary of the data.

PCA

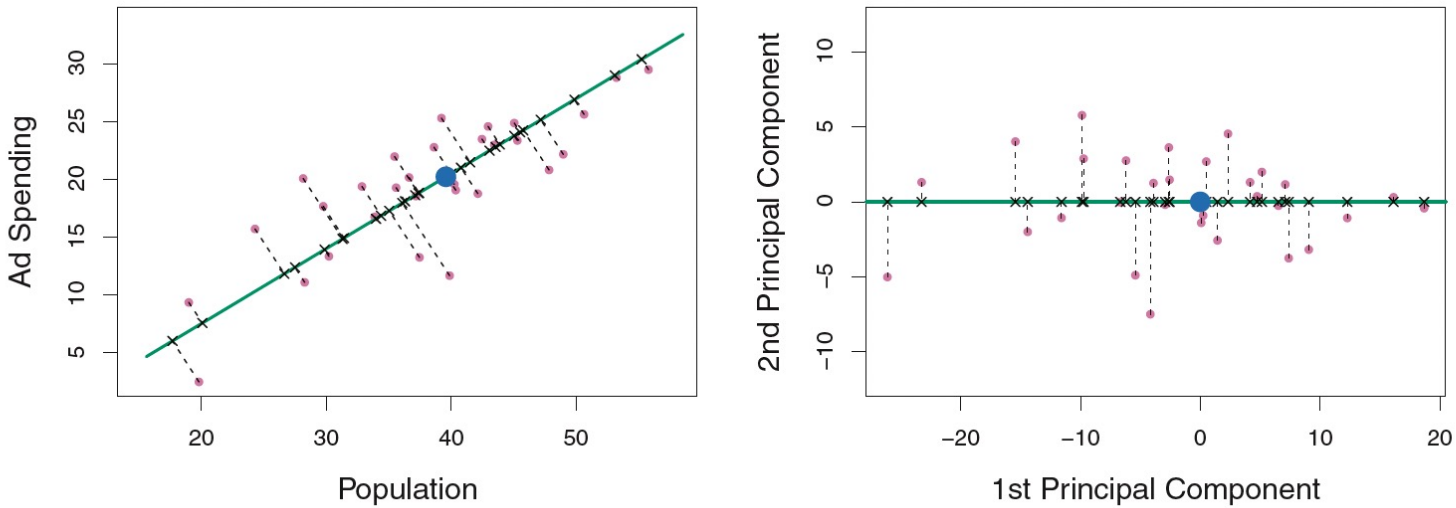


FIGURE 6.15. *A subset of the advertising data. The mean* pop *and* ad *budgets are indicated with a blue circle.* Left: *The first principal component direction is shown in green. It is the dimension along which the data vary the most, and it also defines the line that is closest to all* n *of the observations. The distances from each observation to the principal component are represented using the black dashed line segments. The blue dot represents* $(\overline{\texttt{pop}}, \overline{\texttt{ad}})$. Right: *The left-hand panel has been rotated so that the first principal component direction coincides with the* x*-axis.*

Scree plots (PRACTICAL)

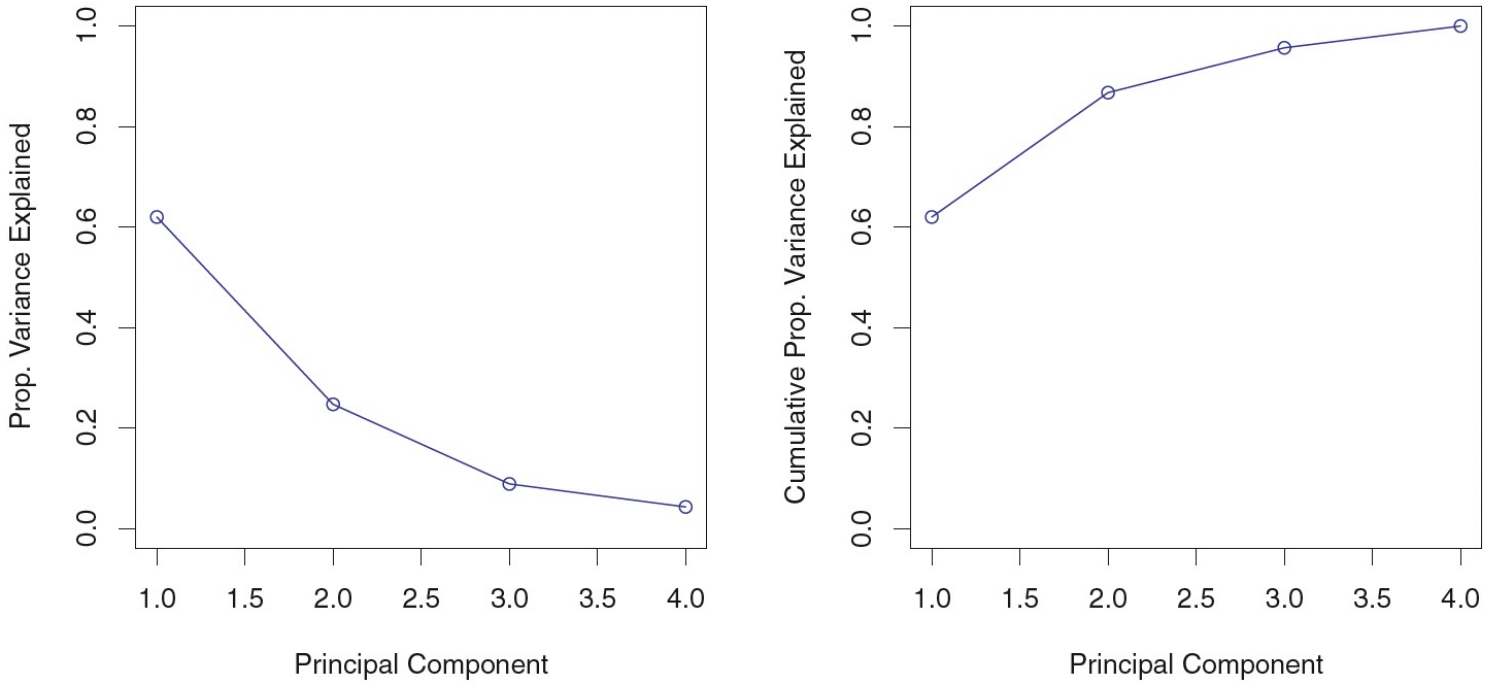
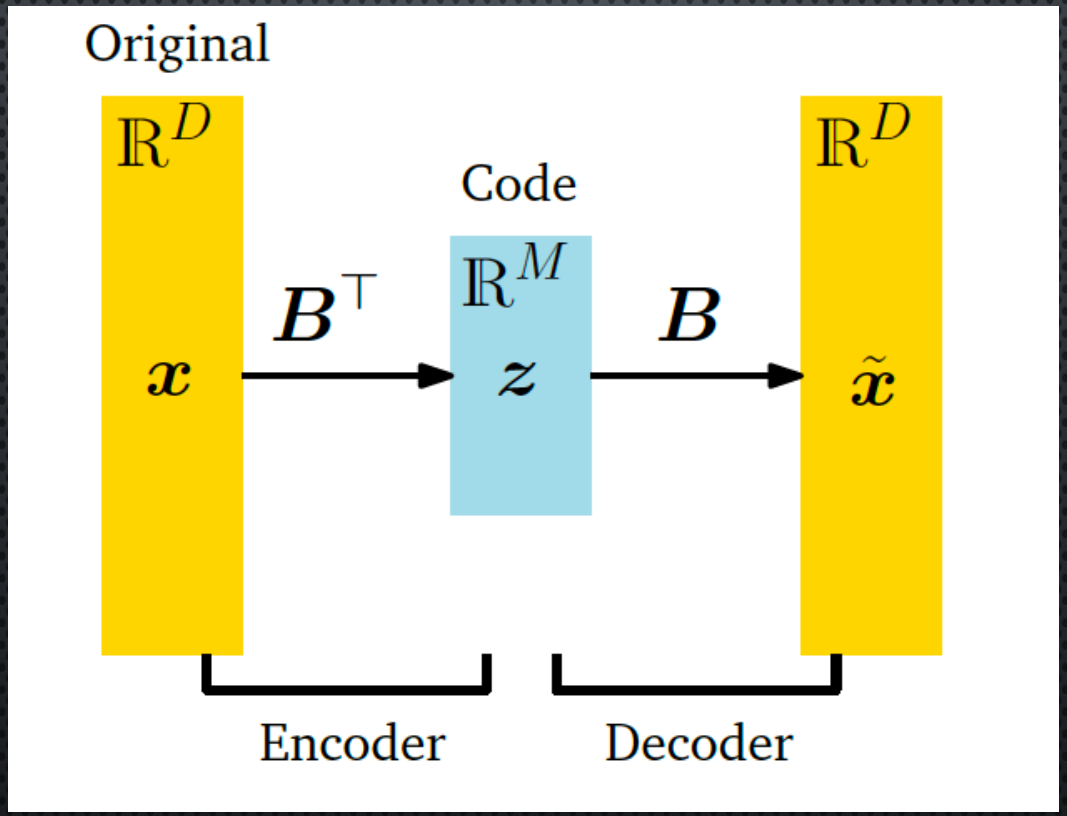


FIGURE 10.4. Left: *a scree plot depicting the proportion of variance explained by each of the four principal components in the* `USArrests` *data.* Right: *the cumulative proportion of variance explained by the four principal components in the* `USArrests` *data.*

Non-linear

- How to make a non-linear dimensionality reduction technique?

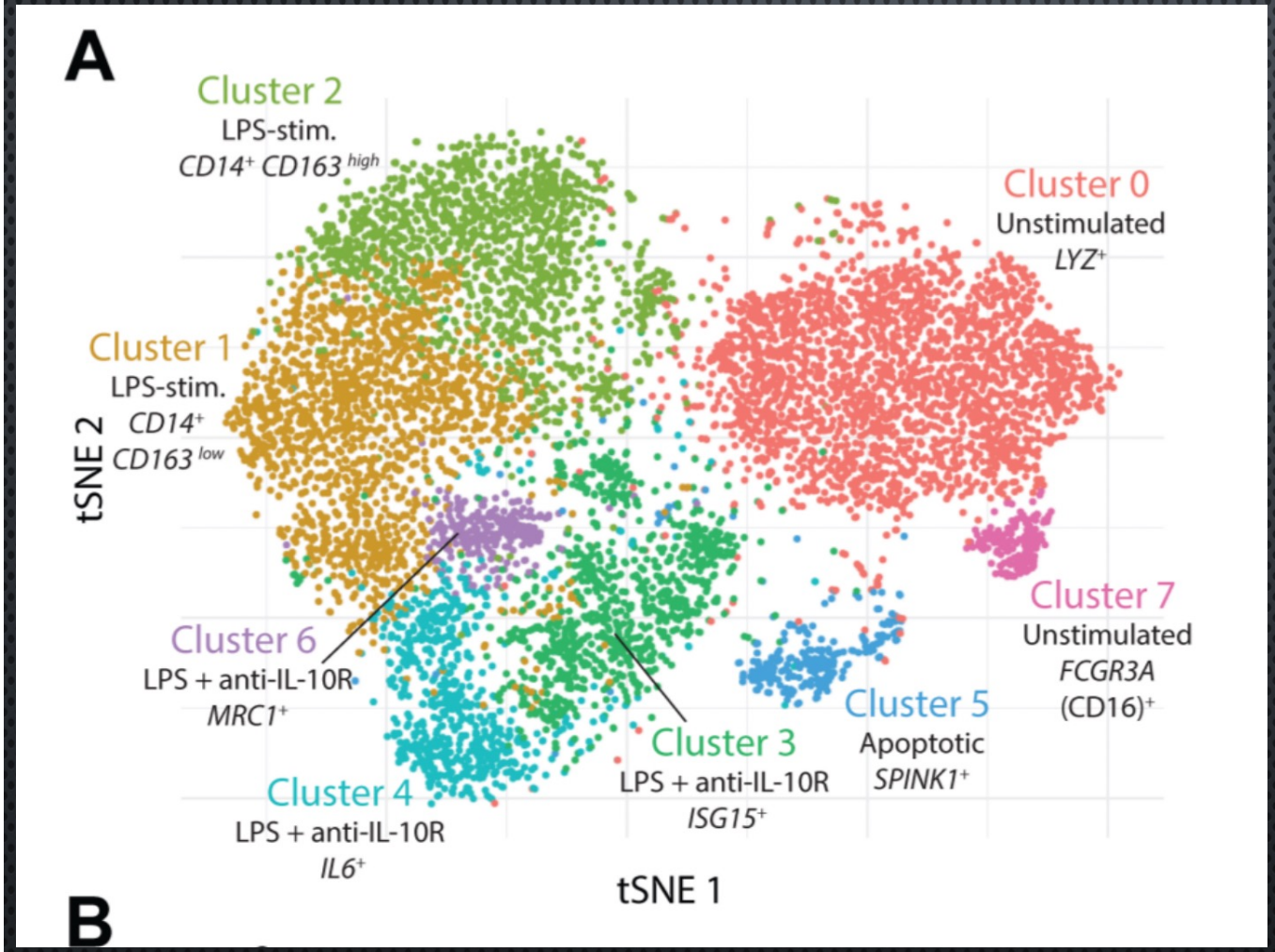
IMPORTANT CONCEPT



INFORMATION BOTTLENECK

GENERALIZATIONS OF THIS IDEA

- tSNE
- Autoencoder (non-linear loss function)



CASE STUDY

a. stochastic
b. distances not preserved
c. difficult to communicate to non-technical experts

VISUALIZATIONS CAN BE MISLEADING

1. DISTANCES NOT PRESERVED IN TSNE

   HTTPS://DISTILL.PUB/2016/MISREAD-TSNE/

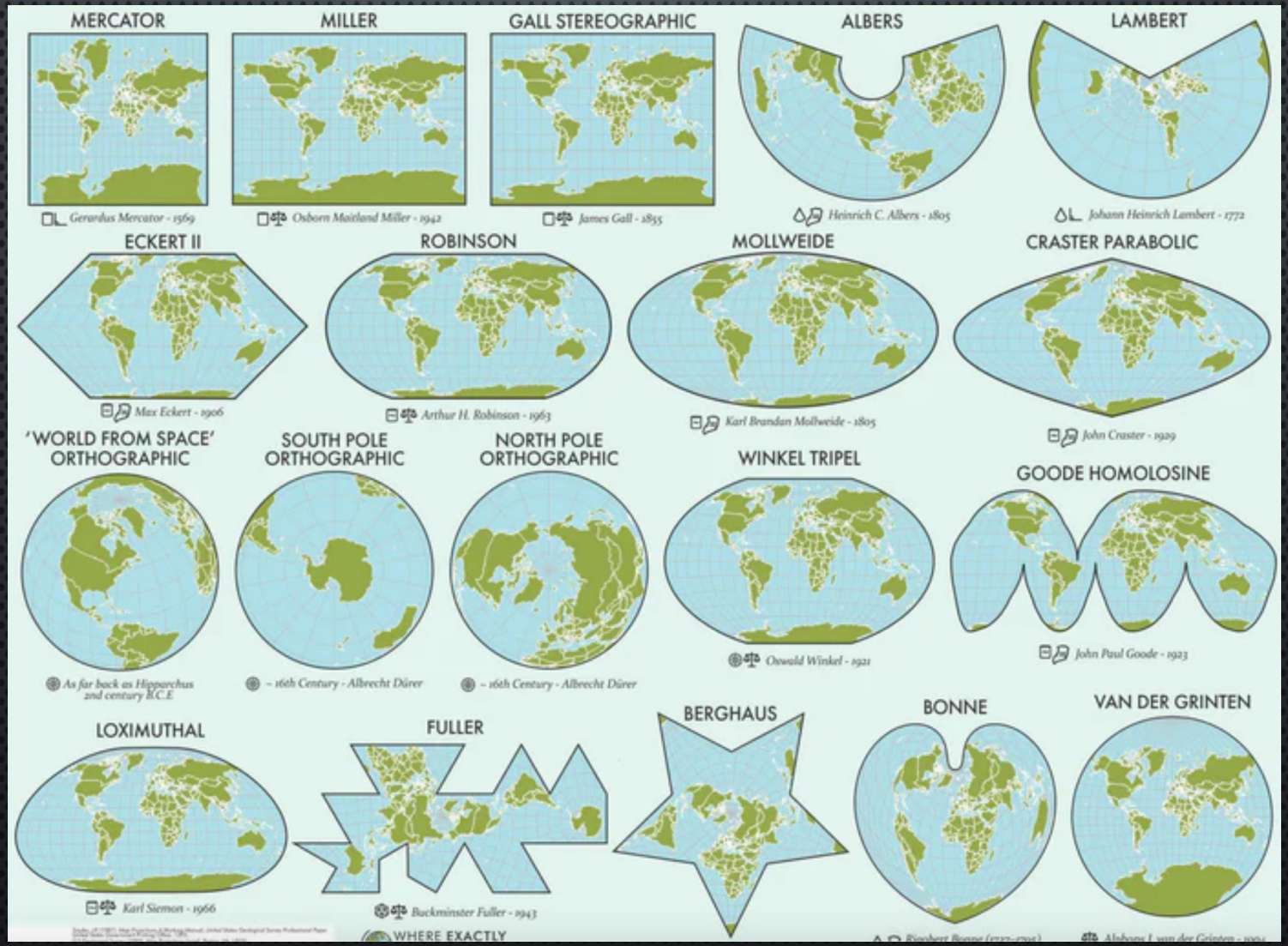
2. CLUSTER SIZES DO NOT MATTER

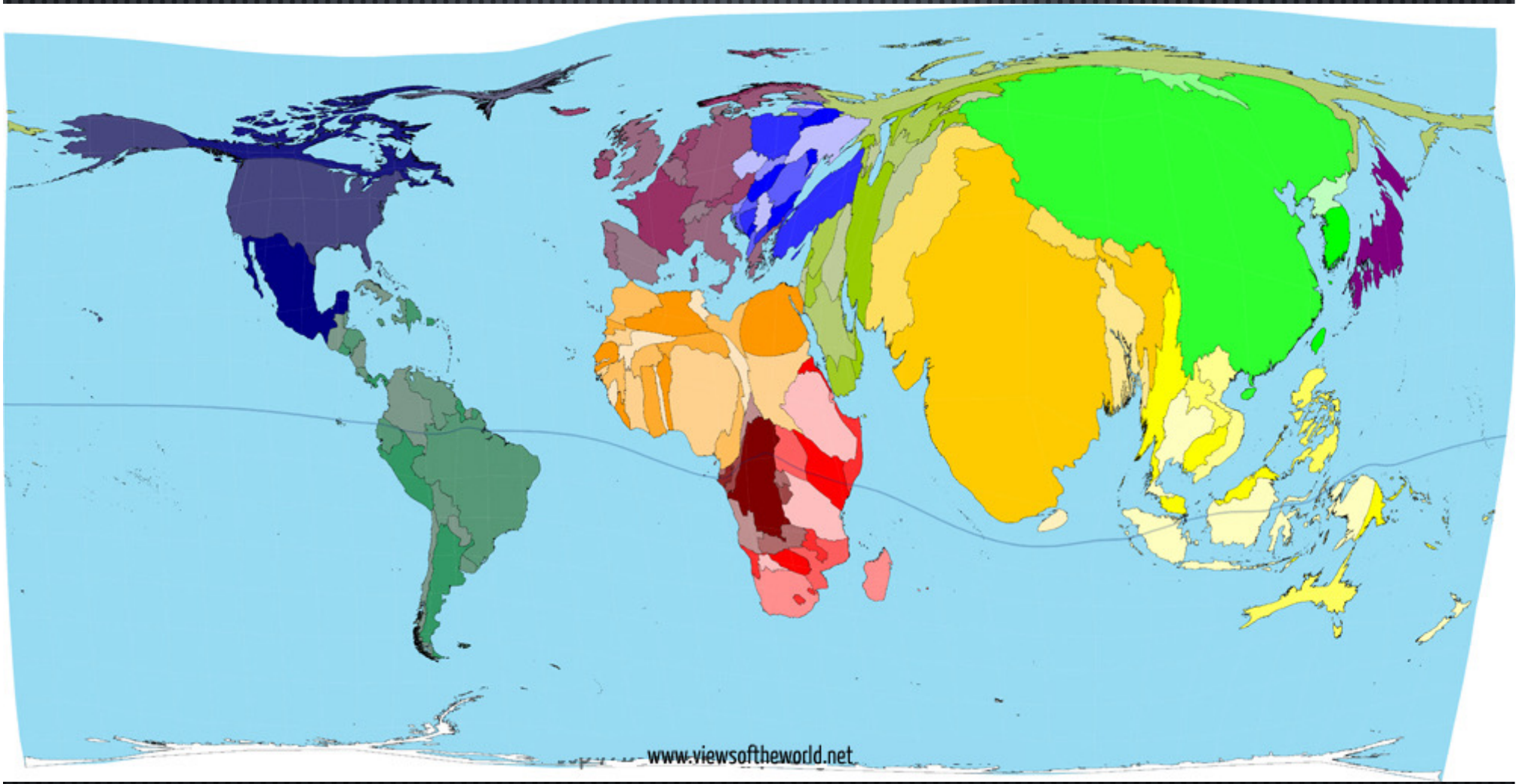
3. YOU CAN SEE SOME SHAPES SOMETIMES

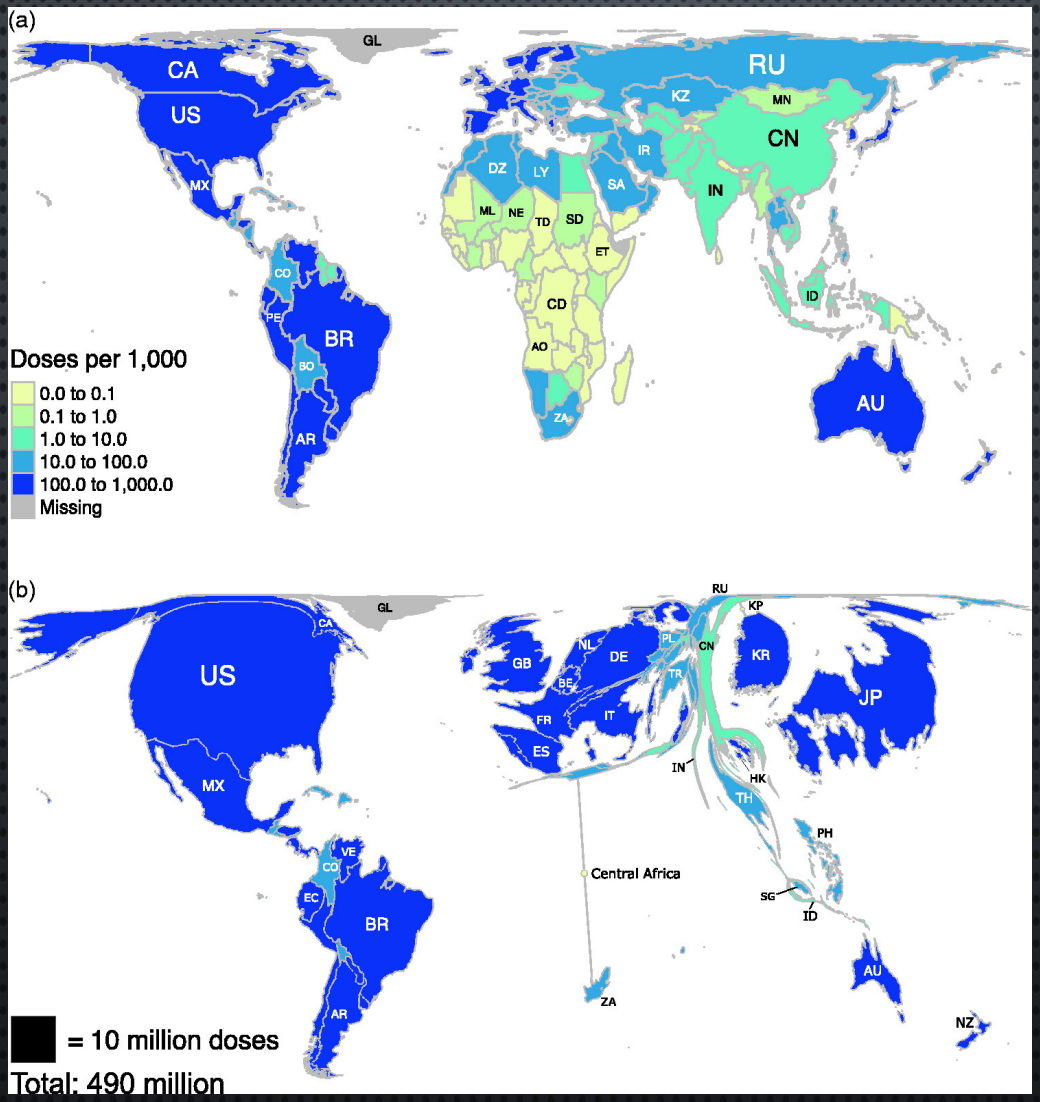
4. RANDOM DOES NOT ALWAYS LOOK RANDOM

VISUALIZING HIGH-DIMENSIONS

1. PROBLEMS WITH COMMUNICATING HIGH DIMENSIONAL DATA
2. HIGH DIMENSIONS ARE DIFFICULT TO VISUALIZE


MERCATOR
Gerardus Mercator - 1569
MILLER
Osborn Maitland Miller - 1942
GALL STEREOGRAPHIC
James Gall - 1855
ALBERS
Heinrich C. Albers - 1805
LAMBERT
Johann Heinrich Lambert - 1772
ECKERT II
Max Eckert - 1906
ROBINSON
Arthur H. Robinson - 1963
MOLLWEIDE
Karl Brandan Mollweide - 1805
CRASTER PARABOLIC
John Craster - 1929
'WORLD FROM SPACE' ORTHOGRAPHIC
As far back as Hipparchus 2nd century B.C.E
SOUTH POLE ORTHOGRAPHIC
~ 16th Century - Albrecht Dürer
NORTH POLE ORTHOGRAPHIC
~ 16th Century - Albrecht Dürer
WINKEL TRIPEL
Oswald Winkel - 1921
GOODE HOMOLOSINE
John Paul Goode - 1923
LOXIMUTHAL
Karl Siemon - 1966
FULLER
Buckminster Fuller - 1943
BERGHAUS
BONNE
VAN DER GRINTEN
WHERE EXACTLY


www.viewsoftheworld.net

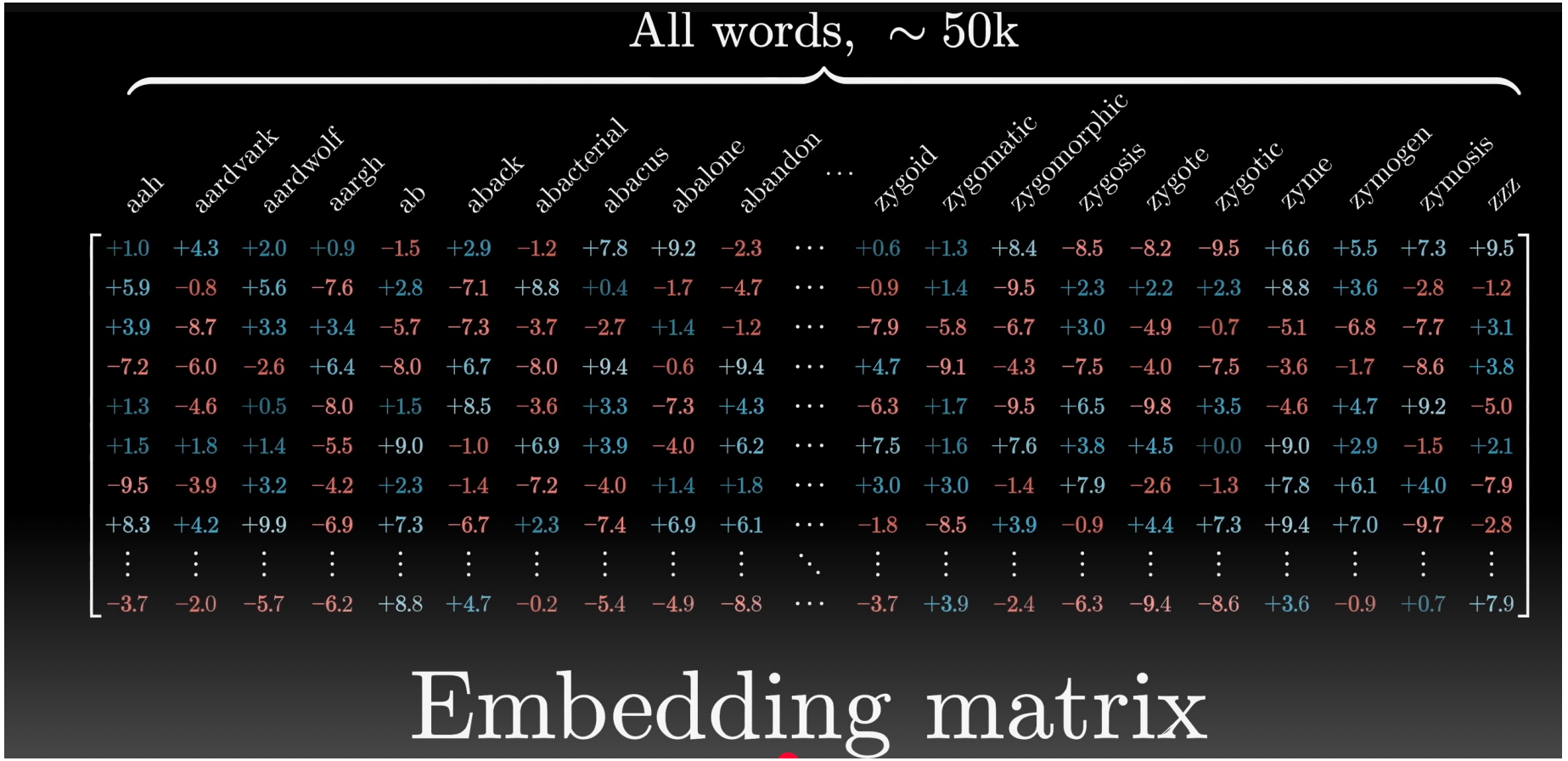


Area proportional to number of vaccine doses

CARTOGRAMS

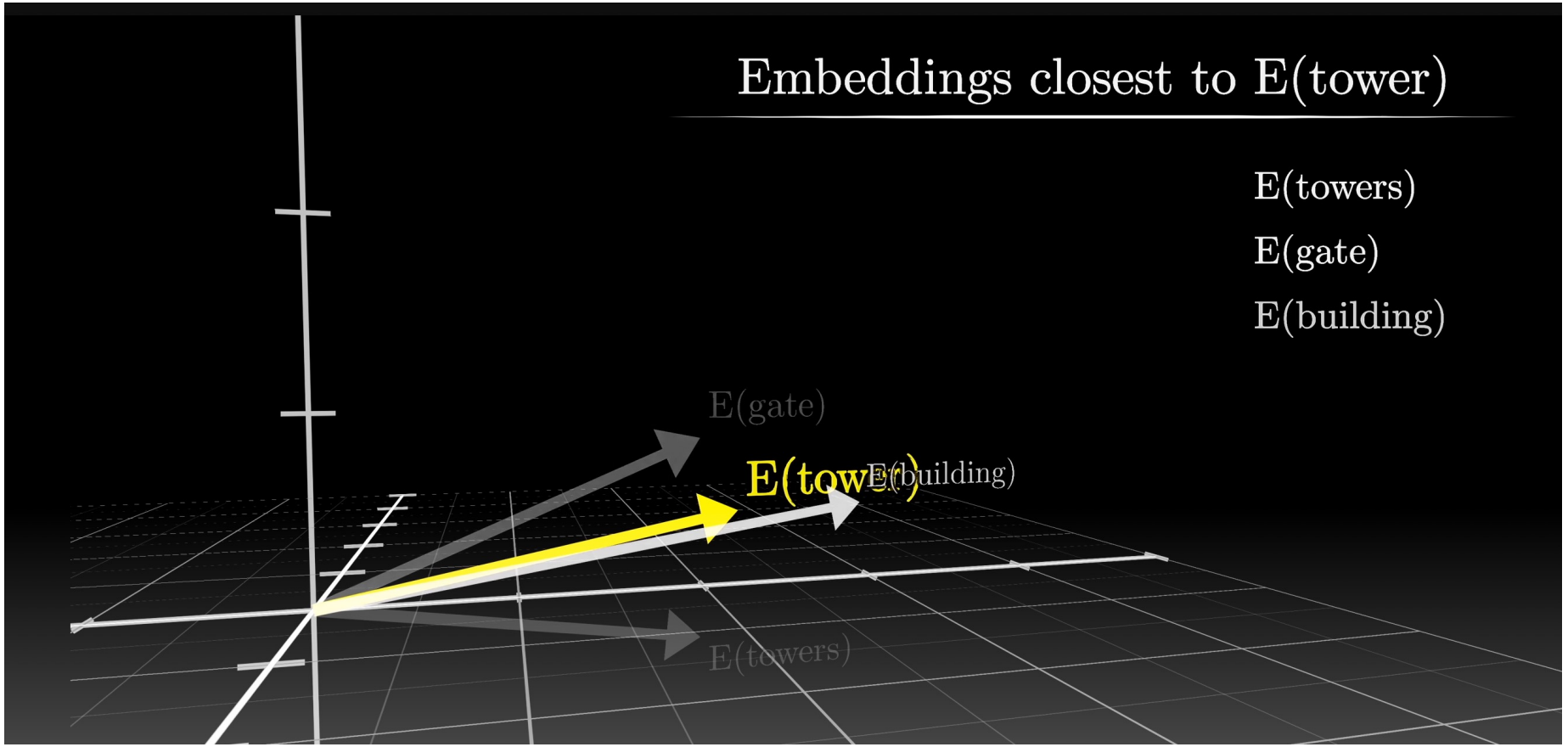
https://go-cart.io/tutorial

Applications to Large-Language Models



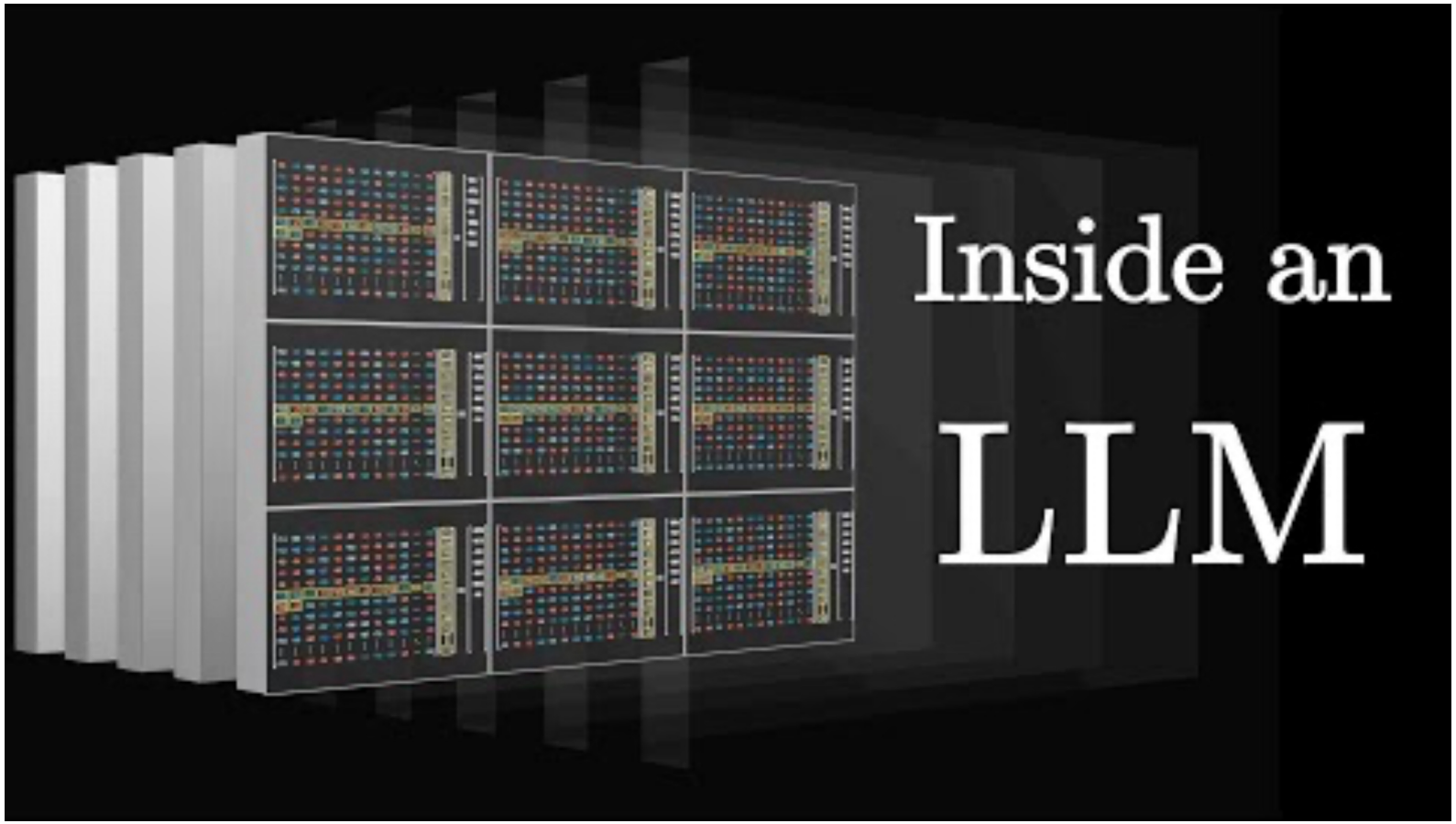
Video by 3blue1brown

Applications to Large-Language Models



Video by 3blue1brown

Applications to Large-Language Models



Video by 3blue1brown

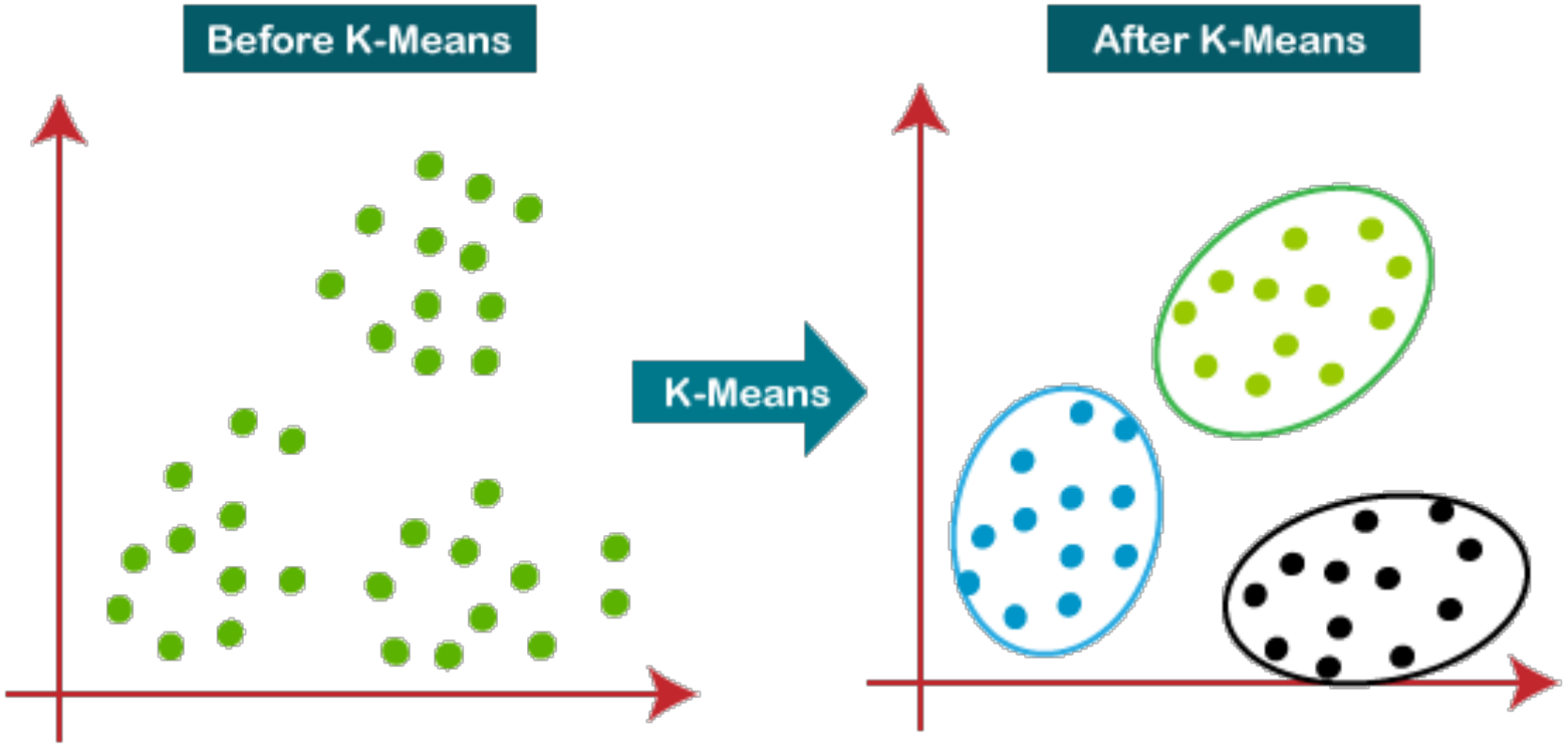
K means Clustering

Clustering

Both clustering and PCA seek to simplify the data via a small number of summaries, but their mechanisms are different:

- PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance;
- Clustering looks to find homogeneous subgroups among the observations.

K-means Clustering



K-means Clustering

Algorithm 10.1 *K-Means Clustering*

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

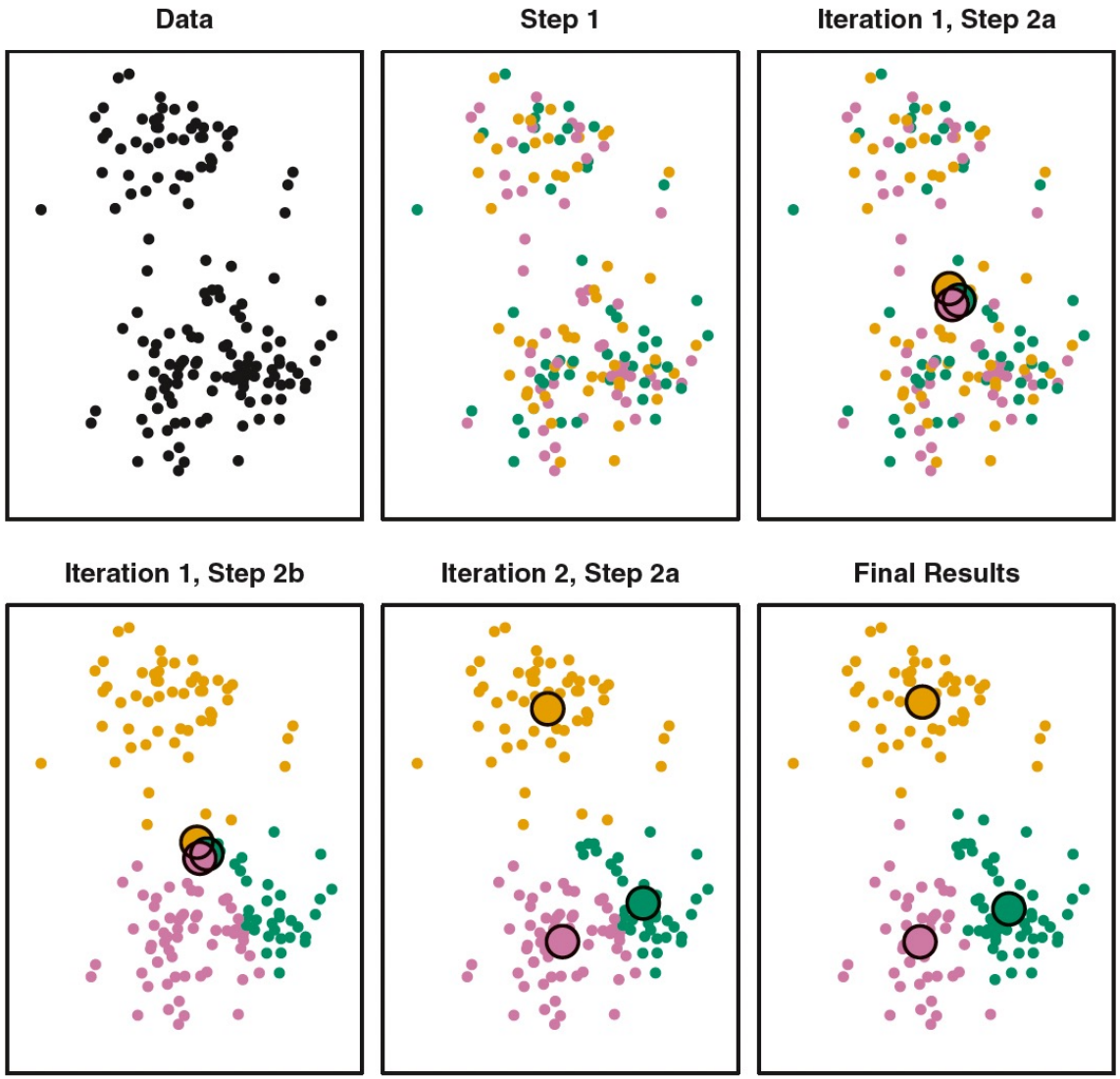


FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with $K=3$.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.* Bottom left: *in Step 2(b), each observation is assigned to the nearest centroid.* Bottom center: *Step 2(a) is once again performed, leading to new cluster centroids.* Bottom right: *the results obtained after ten iterations.*

for cluster C_k is a measure $W(C_k)$ of the amount by which the observations within a cluster differ from each other. Hence we want to solve the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}. \tag{10.9}$$

In words, this formula says that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible.

Solving (10.9) seems like a reasonable idea, but in order to make it actionable we need to define the within-cluster variation. There are many possible ways to define this concept, but by far the most common choice involves *squared Euclidean distance*. That is, we define

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2, \tag{10.10}$$

K-means Clustering

1. Assign each data point to their closest centroid.

2. Work out the mean of each resulting cluster and make this the new centroid.

https://www.youtube.com/watch?v=5I3Ei69I40s&ab_channel=bitLectures

https://www.youtube.com/watch?v=R2e3Ls9H_fc

K-means Clustering

- How to apply this to high-dimensional data (more than 2 variables/features)

- How to visualize the result of k-means on this data?

K means example

A 2-means algorithm is applied to the following 1 dimensional data:

$$0,2,3,7,8,9$$

The algorithm is initialised with the centroids $m_A = 5$ and $m_B = 12$

(a) State which values are in cluster A after the first iteration.

(b) Update the new centroids.

(c) State which values are in cluster A after the second iteration.

(d) Where do the cluster centres converge to?

Slide courtesy Paul Fannon

K means example

Iteration 1:

Cluster A: 0,2,3,7,8

Cluster B: 9

Centroid of cluster A: 4

Centroid of Cluster B: 9

Within Cluster Sum of Squares $= (4-0)^2 + (4-2)^2 \ldots = 46$

Iteration 2:

Cluster A: 0,2,3

Cluster B: 7,8,9

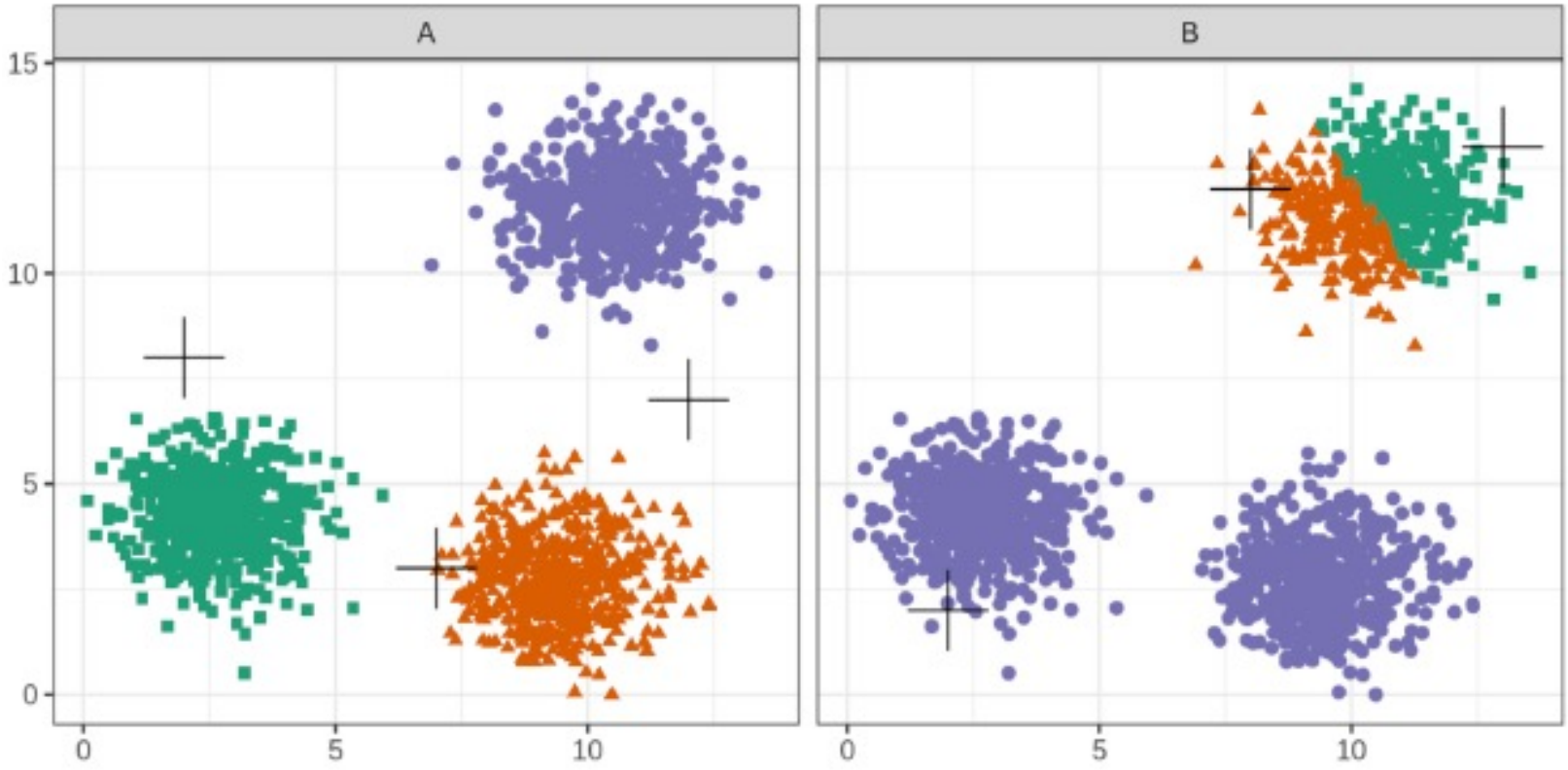
Centroid of cluster A: 1.666…

Centroid of Cluster B: 8

Within Cluster Sum of Squares = 6.666…

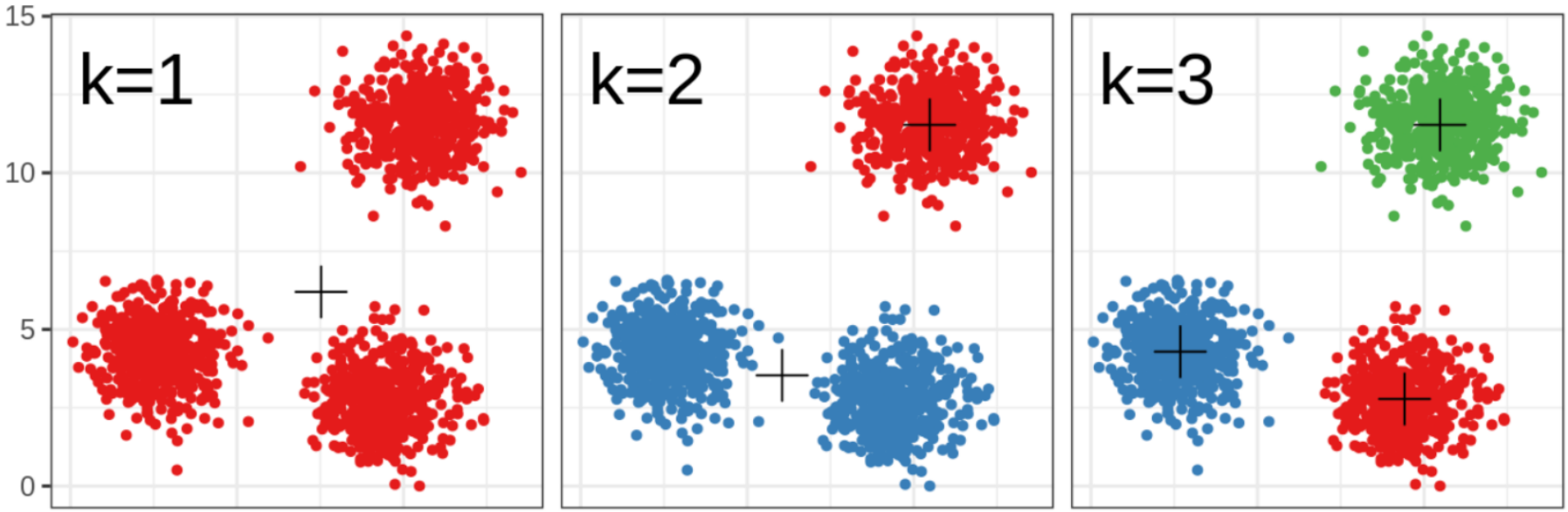
Slide courtesy Paul Fannon

Choice of initial centroids



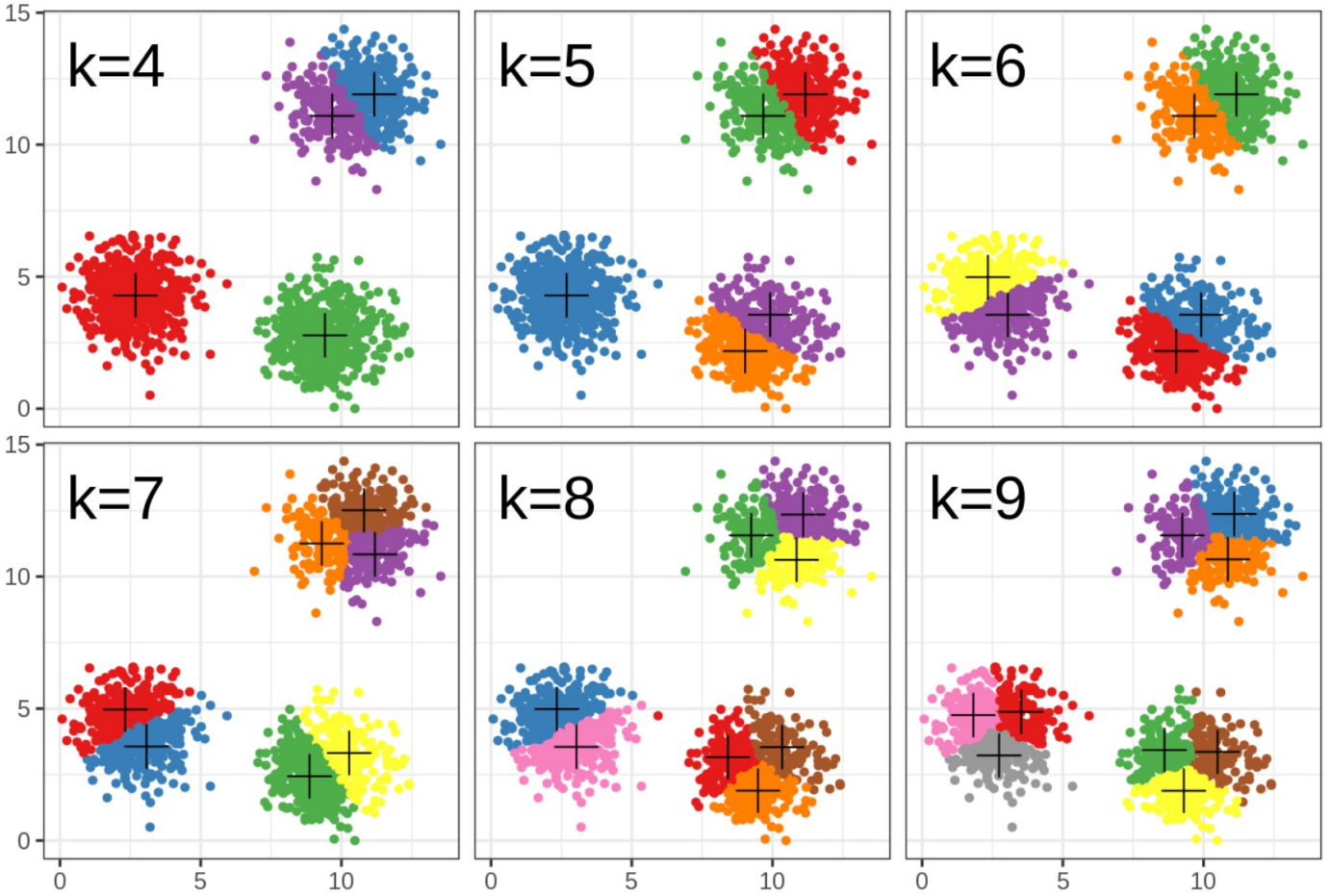
Slide courtesy Paul Fannon

Choosing k



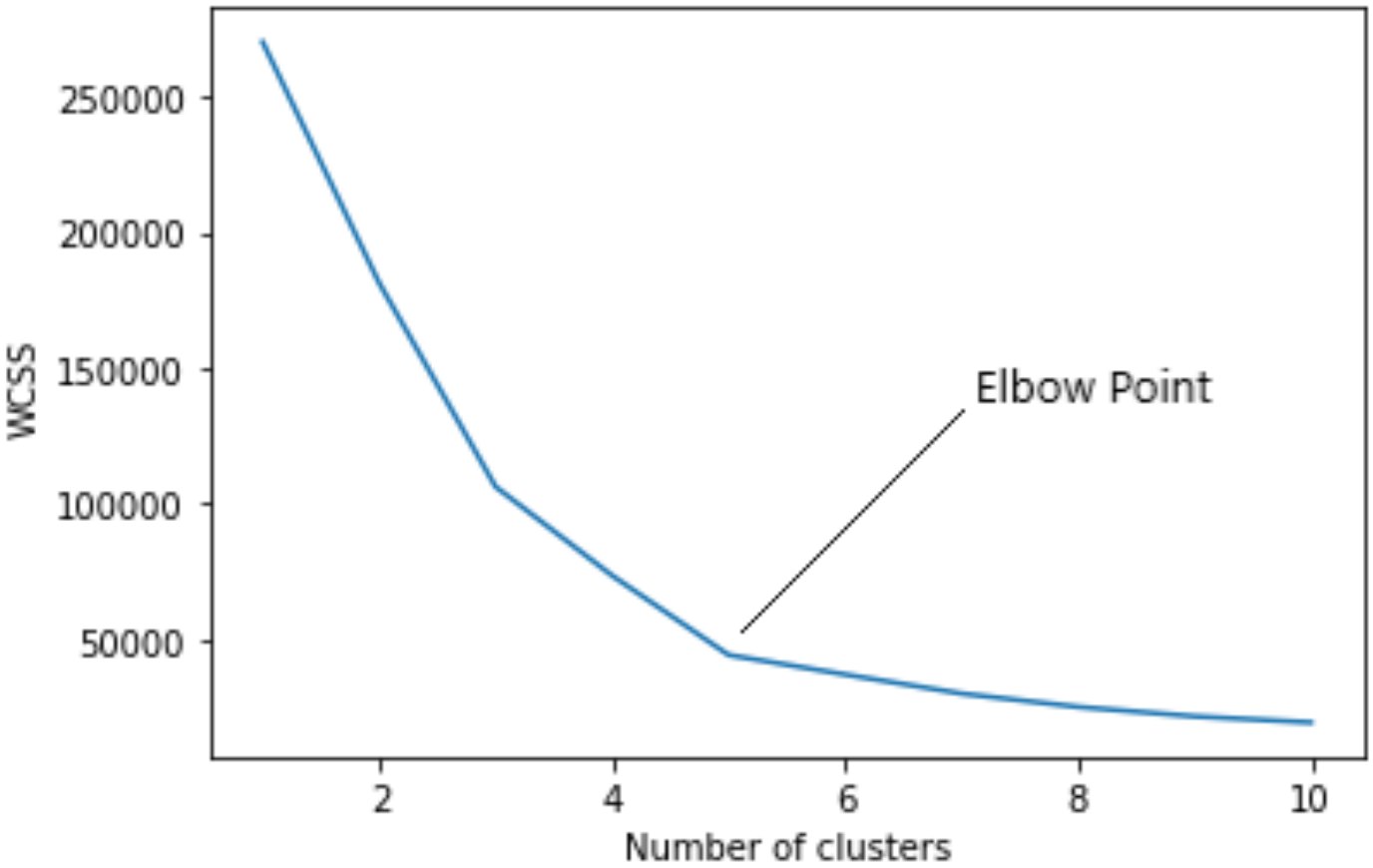
Slide courtesy Paul Fannon

Choosing k



Slide courtesy Paul Fannon

How to find the value of K



Slide courtesy Paul Fannon

Clustering Distances

What is the distance between$(1,3,10)$ and $(0,6,20)$?

- We could use the Euclidean distance:

$$\sqrt{(0-1)^2+(6-3)^2+(20-10)^2} \approx 10.5$$

- We could use the Manhattan distance:

$$|0-1|+|6-3|+|20-10| = 14$$

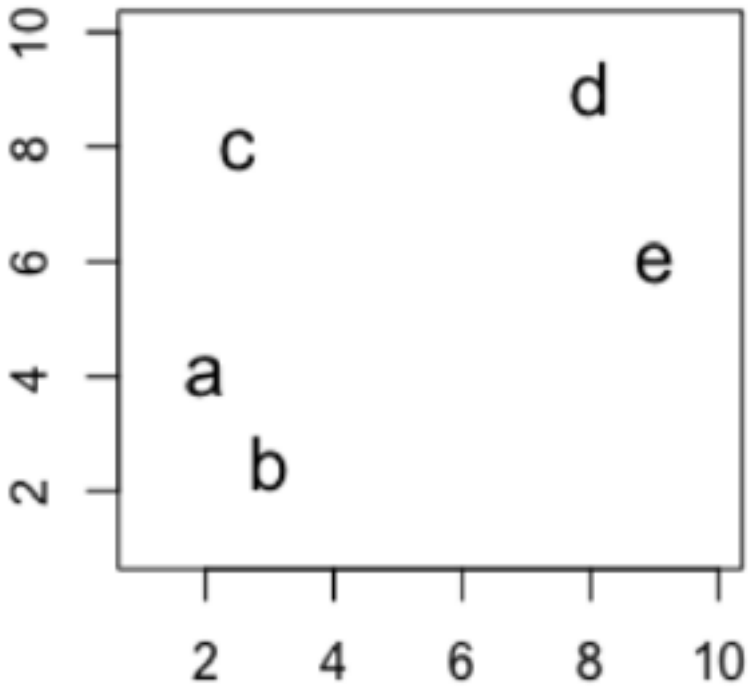
- We could use a correlation distance:

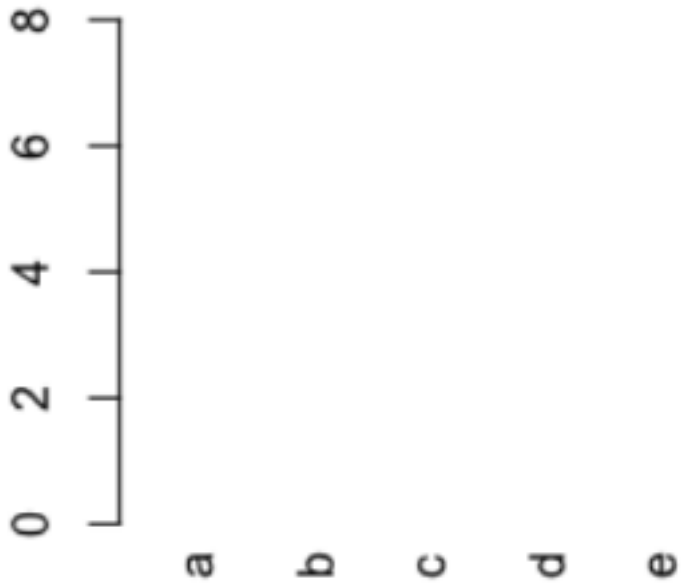
$$1-r \approx 0.0035$$

Slide courtesy Paul Fannon

Hierarchical Clustering

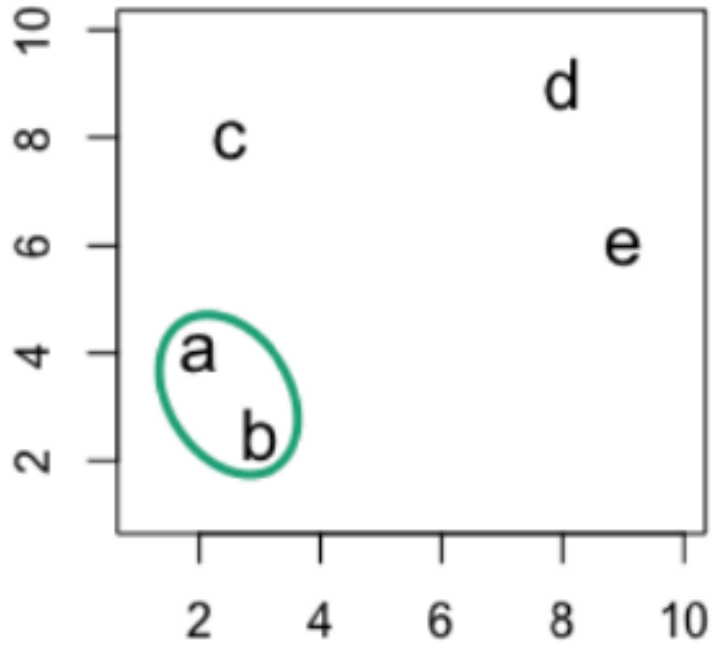
Hierarchical clustering

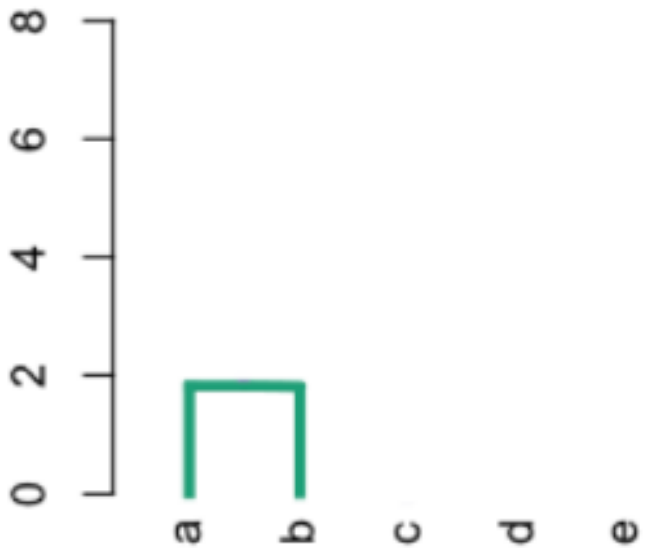




Slide courtesy Paul Fannon

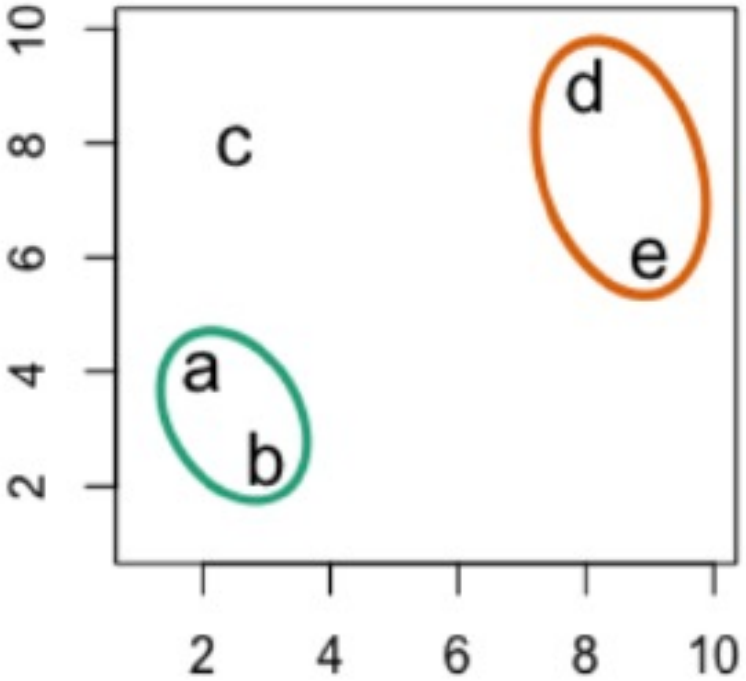
Hierarchical clustering

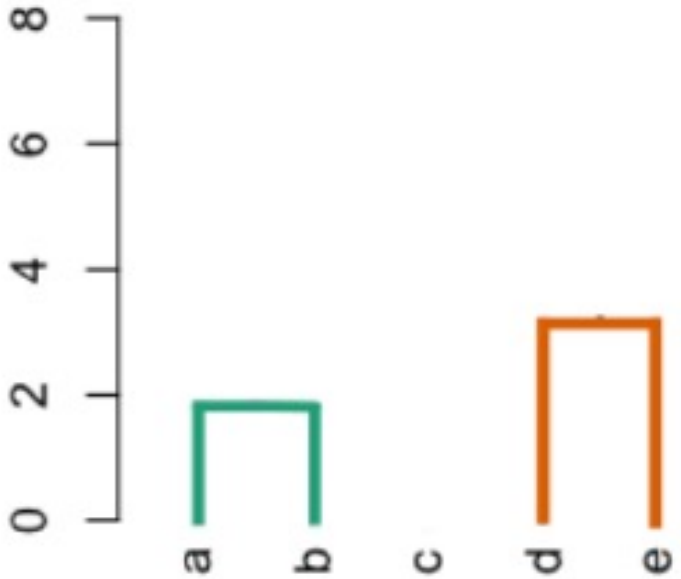




Slide courtesy Paul Fannon

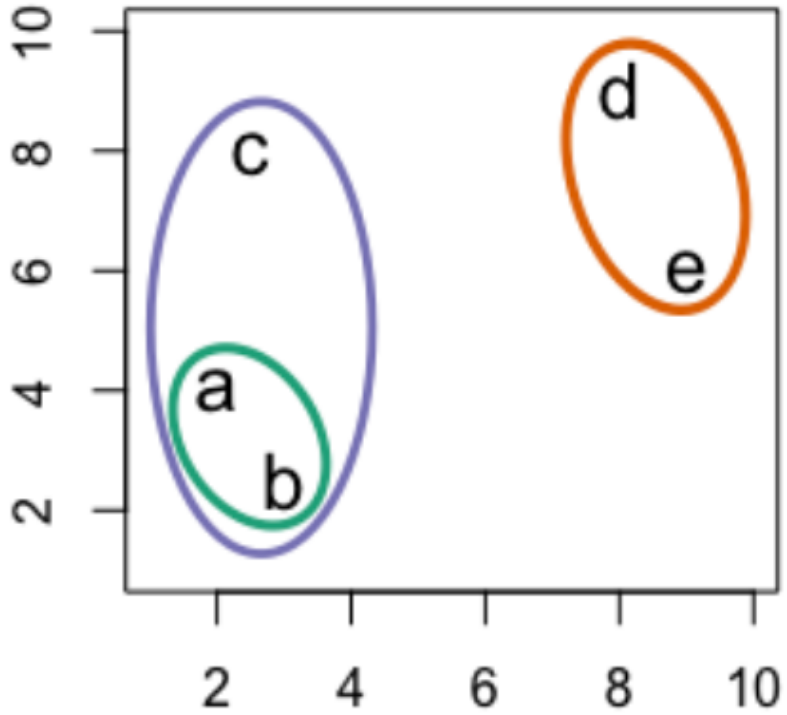
Hierarchical clustering

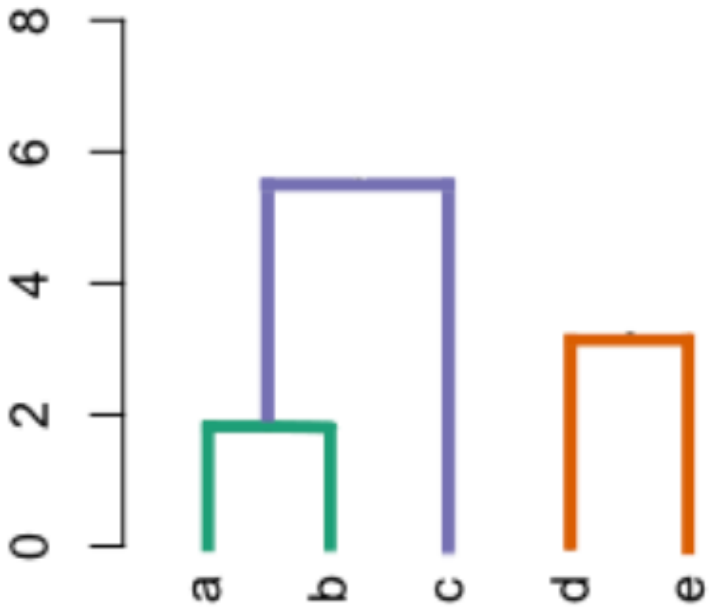




Slide courtesy Paul Fannon

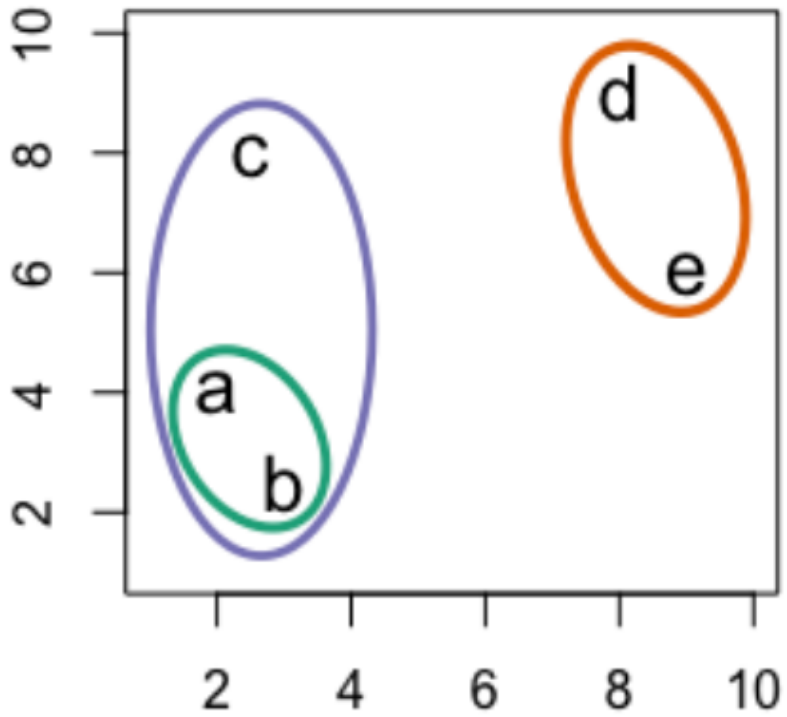
Hierarchical clustering

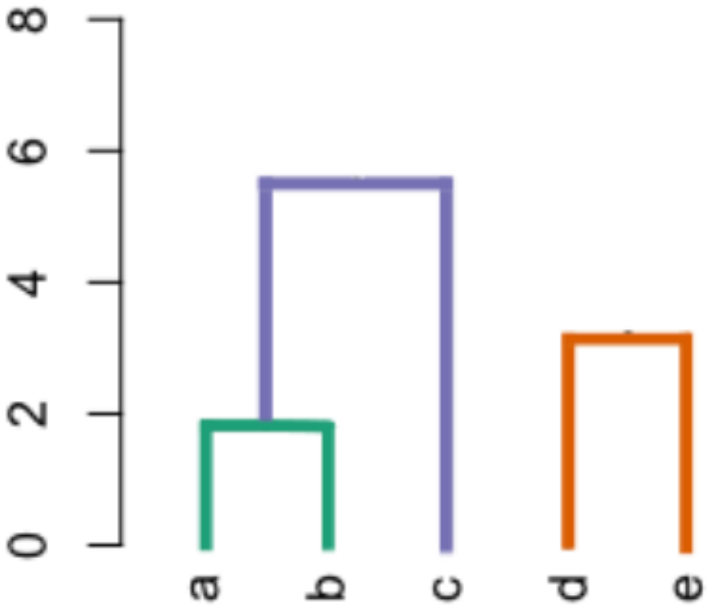




Slide courtesy Paul Fannon

Hierarchical clustering





Slide courtesy Paul Fannon

Hierarchical clustering

Algorithm 10.2 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Dissimilarity measures

- Euclidean distance

- Correlation based distance

Dissimilarity measures

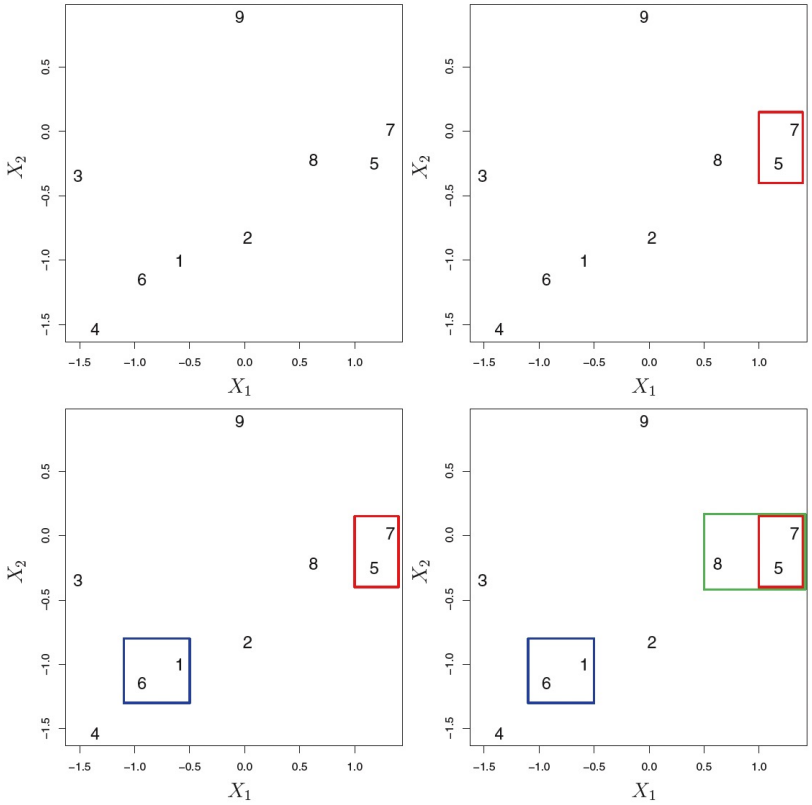


FIGURE 10.11. *An illustration of the first few steps of the hierarchical clustering algorithm, using the data from Figure 10.10, with complete linkage and Euclidean distance.* Top Left: *initially, there are nine distinct clusters,* $\{1\}, \{2\}, \ldots, \{9\}$. Top Right: *the two clusters that are closest together,* $\{5\}$ *and* $\{7\}$, *are fused into a single cluster.* Bottom Left: *the two clusters that are closest together,* $\{6\}$ *and* $\{1\}$, *are fused into a single cluster.* Bottom Right: *the two clusters that are closest together using* complete linkage, $\{8\}$ *and the cluster* $\{5, 7\}$, *are fused into a single cluster.*

Dissimilarity measures

For instance, consider an online retailer interested in clustering shoppers based on their past shopping histories. The goal is to identify subgroups of *similar* shoppers, so that shoppers within each subgroup can be shown items and advertisements that are particularly likely to interest them. Suppose the data takes the form of a matrix where the rows are the shoppers and the columns are the items available for purchase; the elements of the data matrix indicate the number of times a given shopper has purchased a given item (i.e. a 0 if the shopper has never purchased this item, a 1 if the shopper has purchased it once, etc.) What type of dissimilarity measure should be used to cluster the shoppers? If Euclidean distance is used, then shoppers who have bought very few items overall (i.e. infrequent users of the online shopping site) will be clustered together. This may not be desirable. On the other hand, if correlation-based distance is used, then shoppers with similar preferences (e.g. shoppers who have bought items A and B but

Dissimilarity measures

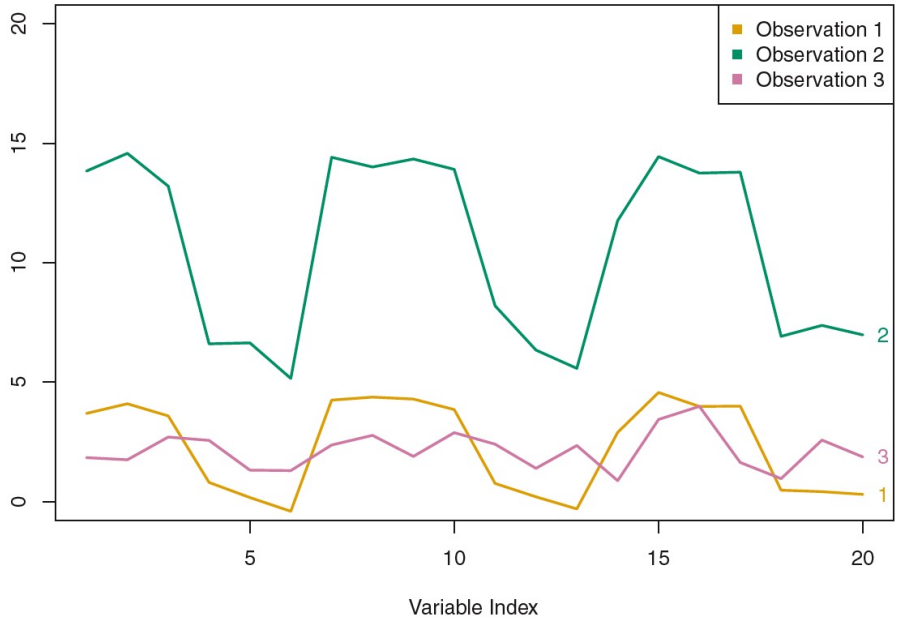


FIGURE 10.13. *Three observations with measurements on 20 variables are shown. Observations 1 and 3 have similar values for each variable and so there is a small Euclidean distance between them. But they are very weakly correlated, so they have a large correlation-based distance. On the other hand, observations 1 and 2 have quite different values for each variable, and so there is a large Euclidean distance between them. But they are highly correlated, so there is a small correlation-based distance between them.*

Dissimilarity measures

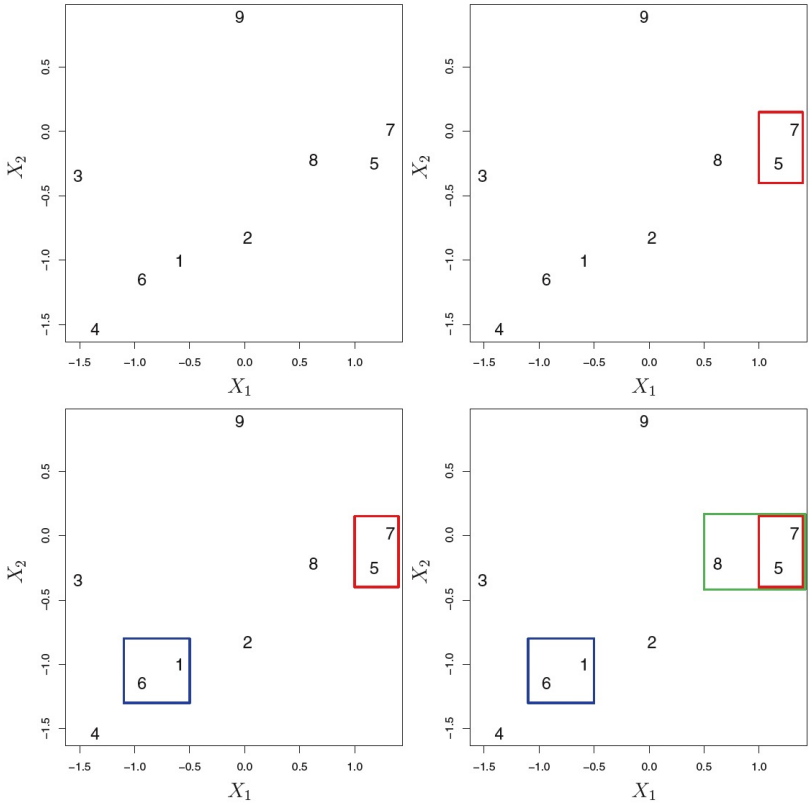


FIGURE 10.11. *An illustration of the first few steps of the hierarchical clustering algorithm, using the data from Figure 10.10, with complete linkage and Euclidean distance.* Top Left: *initially, there are nine distinct clusters,* $\{1\}, \{2\}, \ldots, \{9\}$. Top Right: *the two clusters that are closest together,* $\{5\}$ *and* $\{7\}$, *are fused into a single cluster.* Bottom Left: *the two clusters that are closest together,* $\{6\}$ *and* $\{1\}$, *are fused into a single cluster.* Bottom Right: *the two clusters that are closest together using* complete linkage, $\{8\}$ *and the cluster* $\{5, 7\}$, *are fused into a single cluster.*

Dissimilarity measures

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

TABLE 10.2. *A summary of the four most commonly-used types of linkage in hierarchical clustering.*

Scale variables/features

In addition to carefully selecting the dissimilarity measure used, one must also consider whether or not the variables should be scaled to have standard deviation one before the dissimilarity between the observations is computed. To illustrate this point, we continue with the online shopping example just described. Some items may be purchased more frequently than others; for instance, a shopper might buy ten pairs of socks a year, but a computer very rarely. High-frequency purchases like socks therefore tend to have a much larger effect on the inter-shopper dissimilarities, and hence on the clustering ultimately obtained, than rare purchases like computers. This may not be desirable. If the variables are scaled to have standard deviation one before the inter-observation dissimilarities are computed, then each variable will in effect be given equal importance in the hierarchical clustering performed. We might also want to scale the variables to have standard deviation one if they are measured on different scales; otherwise, the choice of units (e.g. centimeters versus kilometers) for a particular variable will greatly affect the dissimilarity measure obtained. It should come as no surprise that whether or not it is a good decision to scale the variables before computing the dissimilarity measure depends on the application at hand. An example is shown in Figure 10.14. We note that the issue of whether or not to scale the variables before performing clustering applies to K-means clustering as well.

Comparing k means to Hierarchical

Advantage of k means over hierarchical:

- Faster computationally.
- Does not require choice of linkage.

Advantage of hierarchical over k means

- Does not require number of groups to be known in advance.
- No random seeds so more reproducible.
- Does not assume spherical shape.

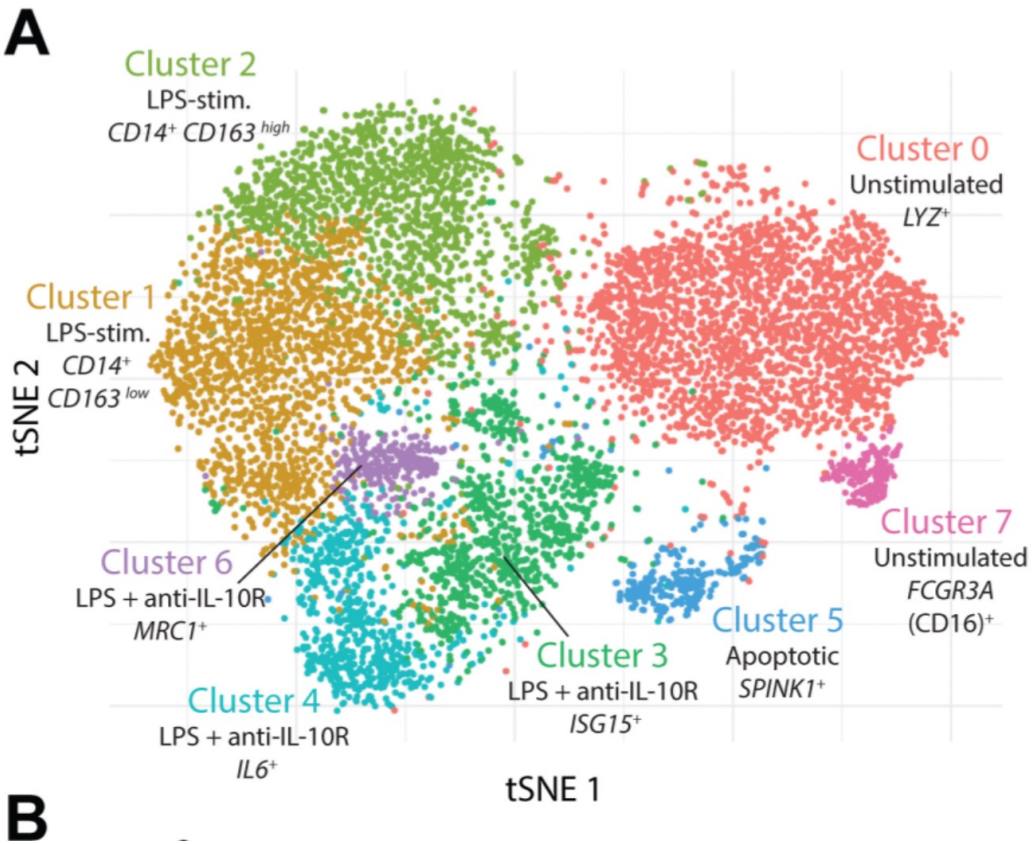
Slide courtesy Paul Fannon

Decisions

In order to perform clustering, some decisions must be made.

- Should the observations or features first be standardized in some way? For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
  - Where should we cut the dendrogram in order to obtain clusters?
- In the case of K-means clustering, how many clusters should we look for in the data?

- Validate clusters in external dataset
- Hypothesis generation



Interactive explanations

- https://mlu-explain.github.io