Module 3 – Statistical techniques for data analysis

Summer school in public auditing and accountability
Data mining and analytics: what implications for auditing?
23-27 July 2018

Vincenzo Mauro

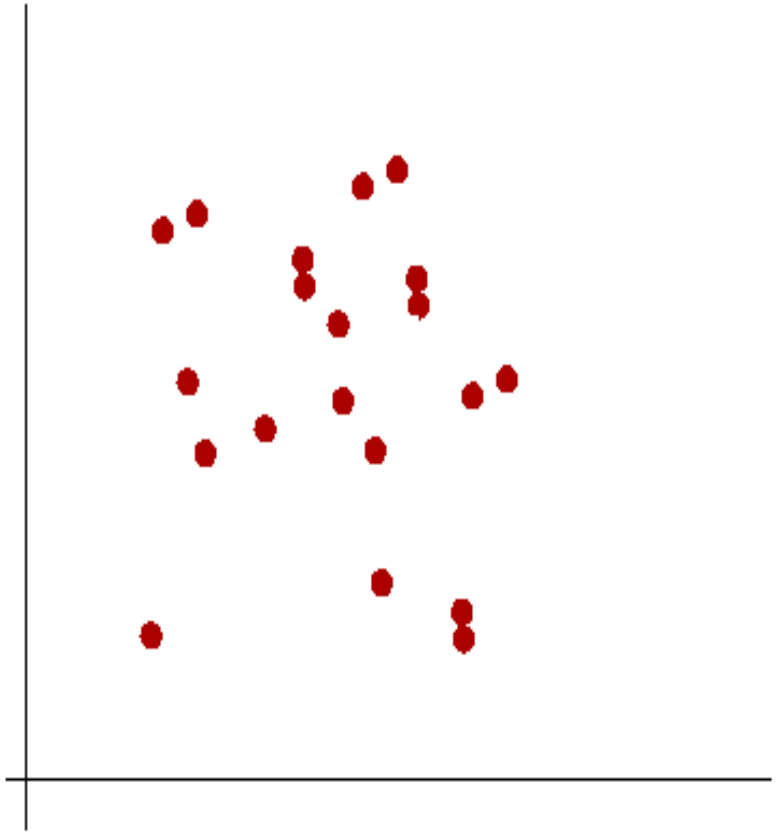
University of Pisa

Outline

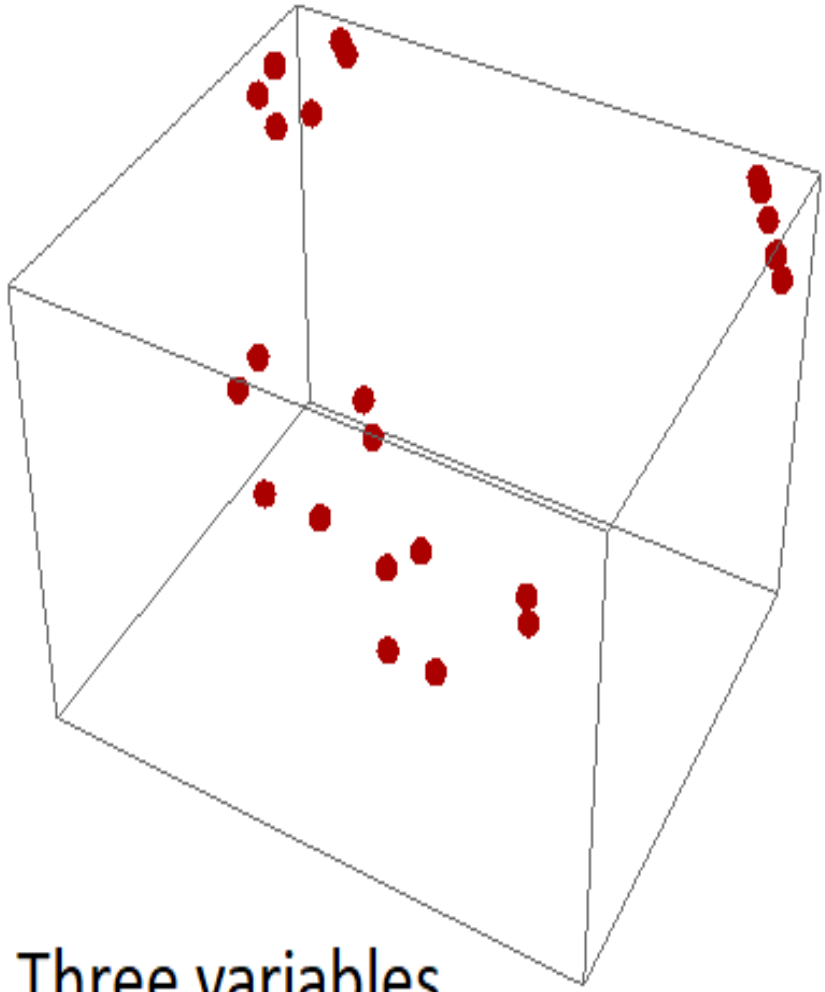
- Principal components analysis
- Cluster analysis

Both using a $n_x p$ data matrix, where the x_{ij} generic entry is the *j*-th achievement for unit *i*

Goal: data reduction



Two variables



Three variables

Both techniques are based on hypotheses, maths, geometrical interpretations, and both of them have many ways to get to the point.

Hard to define/decide what is the “best” solution

PCA: it works (mostly) on variables

Cluster: it works (mostly) on units

The two methods can be combined

Principal Component Analysis (PCA)

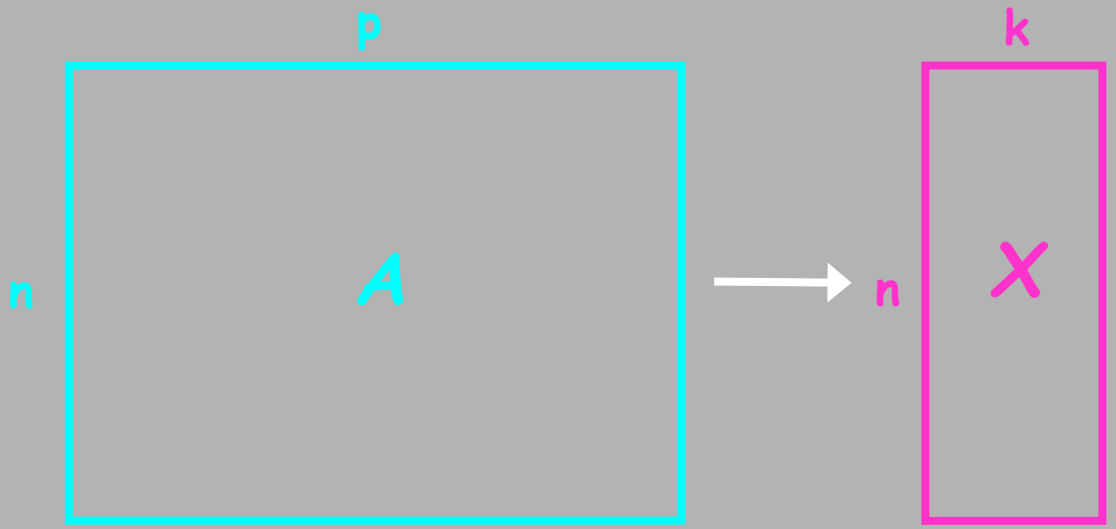
probably the most widely-used and well-known of the “standard” multivariate methods

invented by Pearson (1901) and Hotelling (1933)

(“factor analysis” is very similar to PCA).

Data Reduction

- summarization of data with many (p) variables by a smaller set of (k) derived (synthetic, composite) variables.



Data Reduction

"Residual" variation is information in *A* that is not retained in *X*

balancing act between

- clarity of representation, ease of understanding
- oversimplification: loss of important or relevant information.

Principal Component Analysis (PCA)

takes a data matrix of *n* objects by *p* variables, which may be correlated, and summarizes it by uncorrelated axes (principal components or principal axes) that are linear combinations of the original *p* variables

the first *k* components display as much as possible of the variation among objects.

Geometric Rationale of PCA

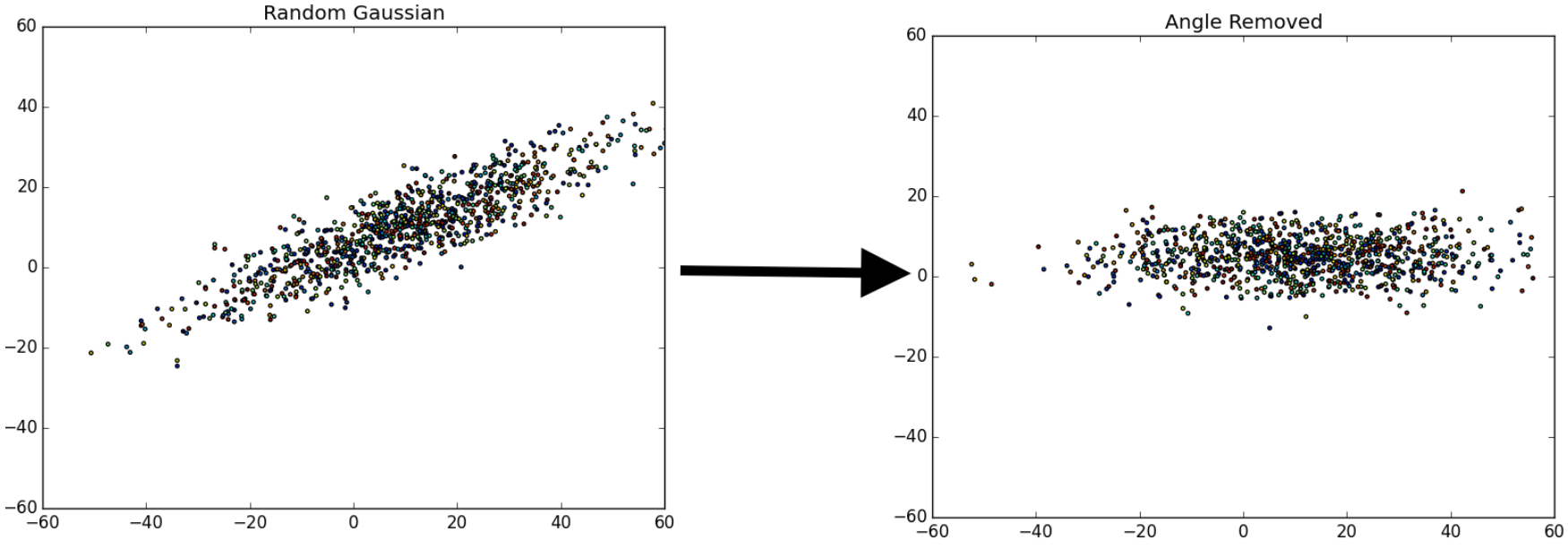
objects are represented as a cloud of *n* points in a multidimensional space with an axis for each of the *p* variables

the variance of each variable is the average squared deviation of its *n* values around the mean of that variable.

$$V_i = \frac{1}{n-1} \sum_{m=1}^{n} \left(X_{im} - \overline{X}_i \right)^2$$

Geometric Rationale of PCA

- degree to which the variables are linearly correlated is represented by their covariances.



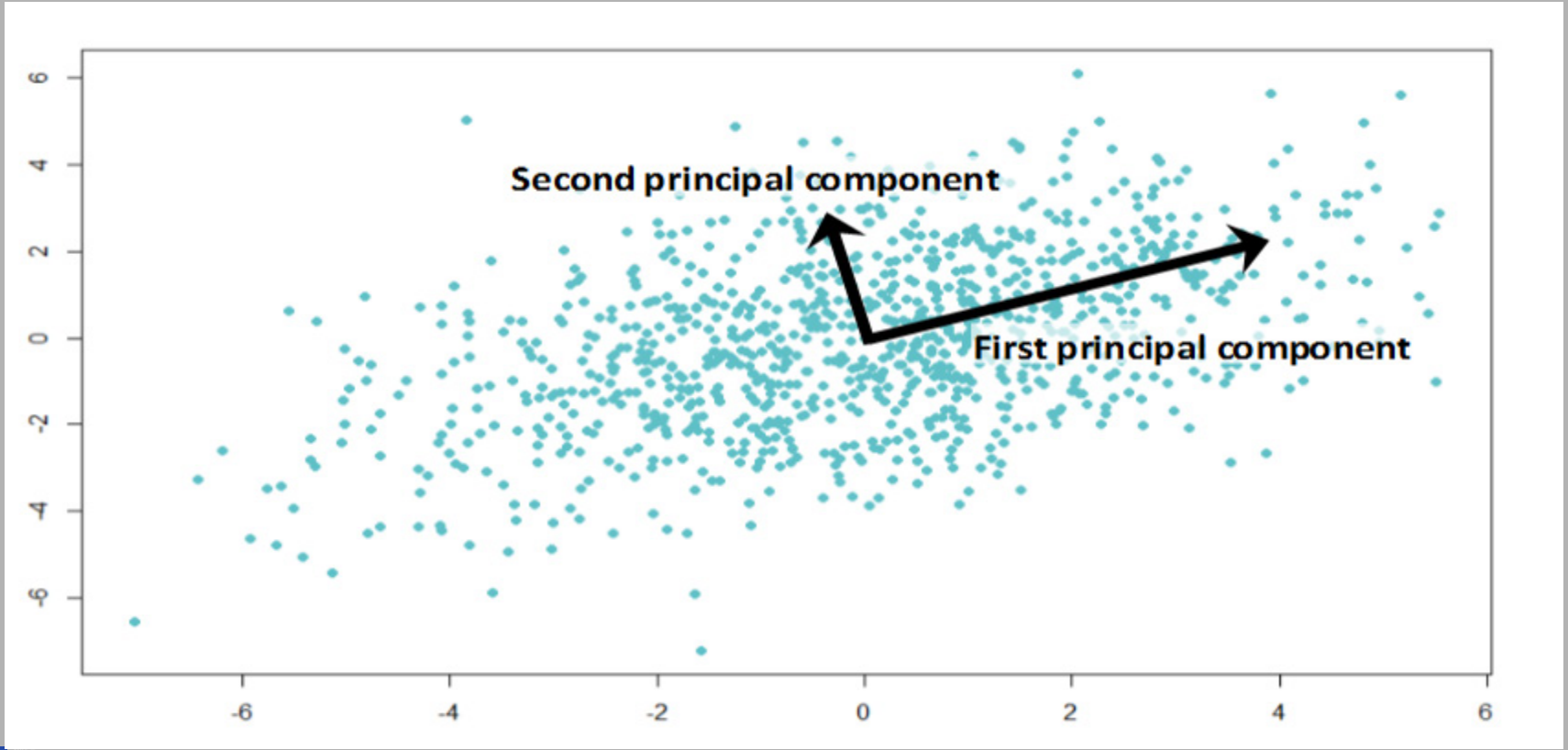
Geometric Rationale of PCA

objective of PCA is to rigidly rotate the axes of this *p*-dimensional space to new positions (principal axes) that have the following properties:

ordered such that principal axis 1 has the highest variance, axis 2 has the next highest variance, ... , and axis *p* has the lowest variance

covariance among each pair of the principal axes is zero (the principal axes are uncorrelated).

Geometric Rationale of PCA



Generalization to *p*-dimensions

if we take the first *k* principal components, they define the *k*-dimensional “hyperplane of best fit” to the point cloud of the total variance of all *p* variables:

PCs 1 to *k* represent the maximum possible proportion of that variance that can be displayed in *k* dimensions

i.e. the squared Euclidean distances among points calculated from their coordinates on PCs 1 to *k* are the best possible representation of their squared Euclidean distances in the full *p* dimensions.

Covariance *vs* Correlation

- using covariances among variables only makes sense if they are measured in the same units
- even then, variables with high variances will dominate the principal components
- these problems are generally avoided by standardizing each variable to unit variance and zero mean.

$$X'_{im} = \frac{\left(X_{im} - \overline{X}_i\right)}{\mathrm{SD}_i}$$

Mean variable *i*

Standard deviation of variable *i*

Covariance *vs* Correlation

covariances between the standardized variables are correlations

after standardization, each variable has a variance of 1

Balancing between the two approaches

The Algebra of PCA

Eigenvector

each eigenvector consists of *p* values which represent the "contribution" of each variable to the principal component axis

eigenvectors are uncorrelated (orthogonal)

their cross-products are zero.

Eigenvectors

	u_1	u_2
X_1	0.7291	-0.6844
X_2	0.6844	0.7291

The Algebra of PCA

coordinates of each object *i* on the k^{th} principal axis, known as the scores on PC *k*, are computed as

$$z_{ki} = u_{1k}x_{1i} + u_{2k}x_{2i} + \cdots + u_{pk}x_{pi}$$

where Z is the *n* x *k* matrix of PC scores, X is the *n* x *p* centered data matrix and U is the *p* x *k* matrix of eigenvectors.

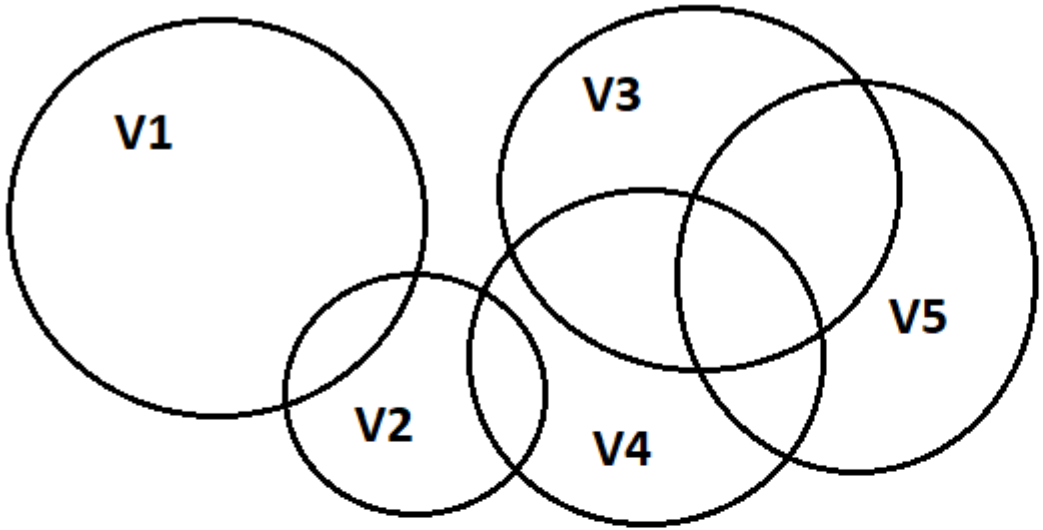
The Algebra of PCA

- variance of the scores on each PC axis is proportional to the corresponding eigenvalue for that axis
- the eigenvalue represents the variance displayed ("explained" or "extracted") by the k^{th} axis
- the sum of the first k eigenvalues is the variance explained by the k-dimensional ordination.

Eigenvalues

Axis	Eigenvalue	% of Variance	Cumulative % of Variance
1	5.855	36.60	36.60
2	3.420	21.38	57.97
3	1.122	7.01	64.98
4	1.116	6.97	71.95
5	0.982	6.14	78.09
6	0.725	4.53	82.62
7	0.563	3.52	86.14
8	0.529	3.31	89.45
9	0.476	2.98	92.42
10	0.375	2.35	94.77

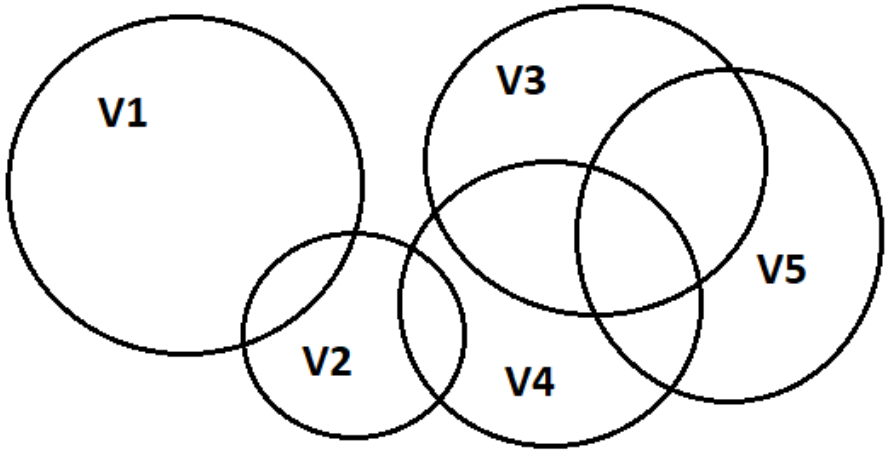
Generalization to *p*-dimensions



Overlapping, correlated, redundant.

I want to reduce dimensionality to 2

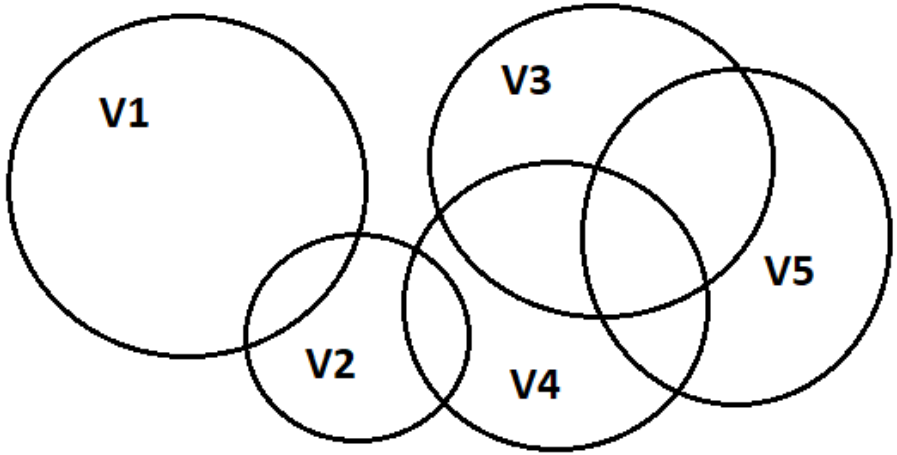
Generalization to *p*-dimensions

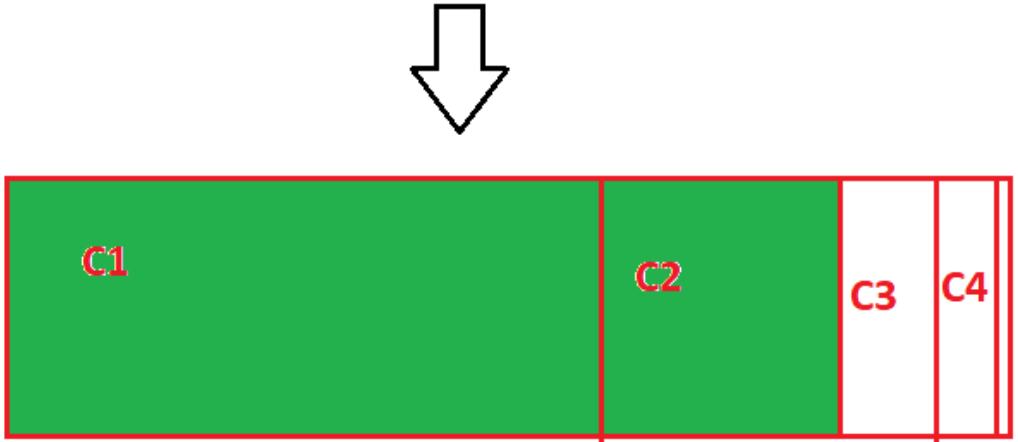


Overlapping, correlated, redundant.

I want to reduce dimensionality to 2

Generalization to *p*-dimensions



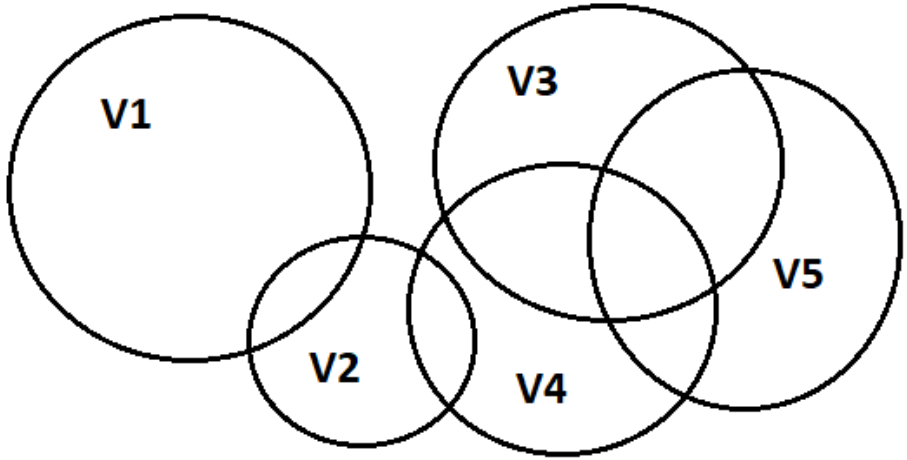


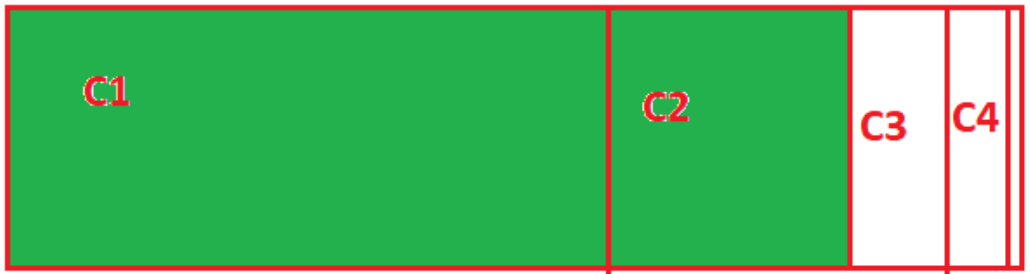
Overlapping, correlated, redundant.

I want to reduce dimensionality to 2

I keep 2 components only
Big reduction, at what cost?

Generalization to *p*-dimensions





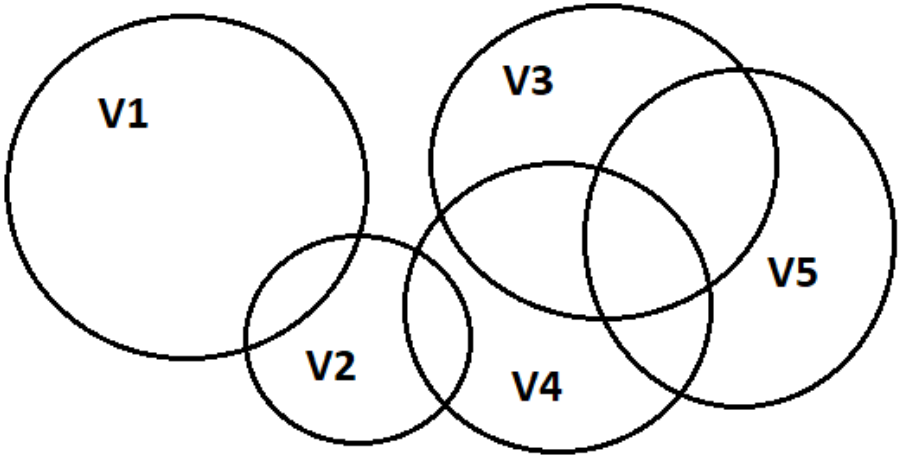
Overlapping, correlated, redundant.

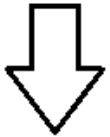
I want to reduce dimensionality to 2

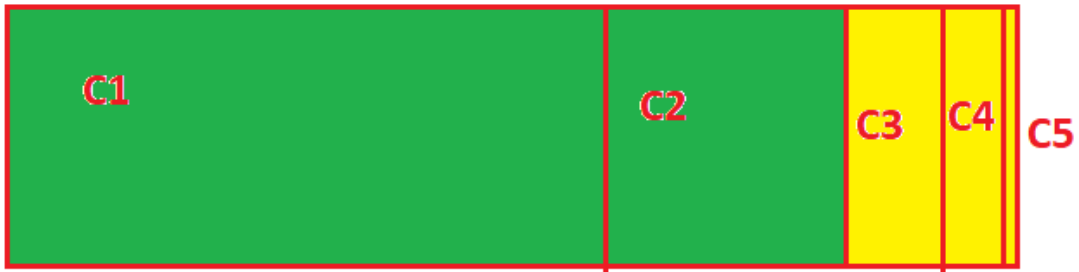
I keep 2 components only
Big reduction, at what cost?

C1 + C2 still mantain 85% of the total variance (information)

Generalization to *p*-dimensions







The part in yellow is the price to pay for the data reduction

Main issue with the components

They can’t be directly interpreted !!

Interpreting components

- correlations between variables and the principal axes are known as loadings
- each element of the eigenvectors represents the contribution of a given variable to a component

	1	2	3
ROE	0.3842	0.0659	-0.1177
ROA	0.2159	0.1696	-0.0578
Asset Turnover	-0.2729	-0.1200	0.3636
Cash Ratio	0.0538	-0.2800	0.2621
ROI	-0.0765	0.3855	-0.1462
ER	0.0248	0.4879	0.2426
ROI	0.0599	0.4568	0.2497
RaROC	0.0789	0.4223	0.2278
Debt ratio	0.0305	0.5587	-0.0276
Earnings per share	-0.3053	0.1226	0.1145
Net present value	-0.3144	0.0402	-0.1067
%W	-0.0886	-0.0654	-0.1171

Interpreting components

- Example using financial ratios
- Component 1: use of assets and control of expenses
- Component 2: availability of cash and capacity to pay debts
- Component3: financial health of the company

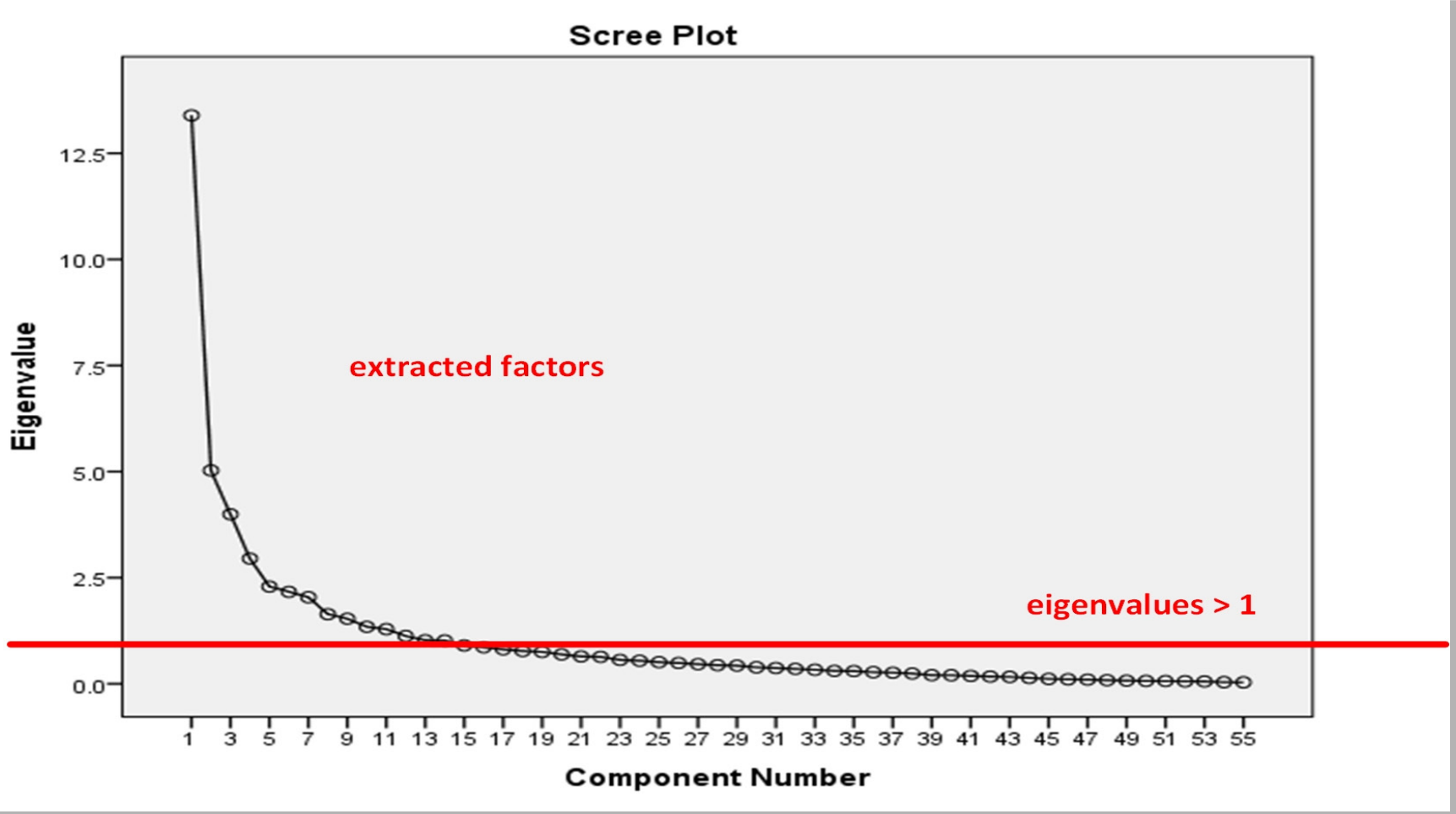
	1	2	3
ROE	0.3842	0.0659	-0.1177
ROA	0.2159	0.1696	-0.0578
Asset Turnover	-0.2729	-0.1200	0.3636
Cash Ratio	0.0538	-0.2800	0.2621
ROI	-0.0765	0.3855	-0.1462
ER	0.0248	0.4879	0.2426
ROI	0.0599	0.4568	0.2497
RaROC	0.0789	0.4223	0.2278
Debt ratio	0.0305	0.5587	-0.0276
Earnings per share	-0.3053	0.1226	0.1145
Net present value	-0.3144	0.0402	-0.1067
%W	-0.0886	-0.0654	-0.1171

How many axes are needed?

several tests and rules have been proposed

a common “rule of thumb” when PCA is based on correlations is that axes with eigenvalues > 1 are worth interpreting

Best graphical solution: k=2 (bidimensional representation)


Scree Plot
Eigenvalue
12.5
10.0
7.5
5.0
2.5
0.0
extracted factors
eigenvalues > 1
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55
Component Number

What are the assumptions of PCA?

assumes relationships among variables are LINEAR

- cloud of points in *p*-dimensional space has linear dimensions that can be effectively summarized by the principal axes

if the structure in the data is NONLINEAR (the cloud of points twists and curves its way through *p*-dimensional space), the principal axes will not be an efficient and informative summary of the data.

Cluster analysis

Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert Tryon in 1939 and used by Cattell beginning in 1943 for trait theory classification in personality psychology.

Its aim is grouping a set of objects in such a way that objects in the same group (called a **cluster**) are in some sense similar

It can be achieved by various algorithms that differ significantly, with the appropriate depending on the specific data set and intended use of the results.

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms

“A group of data objects” (units).

A good clustering method will produce high quality clusters with

- high intra-class similarity
- low inter-class similarity

The quality of a clustering result depends on both the similarity measure used by the method and its implementation.

The quality of a clustering method is also measured by its ability to discover hidden patterns

Cluster models

Cluster models are many, and include: *Connectivity models*: for example, hierarchical clustering builds models based on distance connectivity.

- *Centroid models*: for example, the k-means algorithm represents each cluster by a single mean vector.
- *Distribution models*: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the expectation-maximization algorithm.
- *Density models*: for example, DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- *Subspace models*: in biclustering (also known as co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.

Cluster models

- *Group models*: some algorithms do not provide a refined model for their results and just provide the grouping information.
- *Graph-based models*: a clique, that is, a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques, as in the HCS clustering algorithm.
- *Neural models*: the most well known unsupervised neural network is the self-organizing map and these models can usually be characterized as similar to one or more of the above models, and including subspace models when neural networks implement a form of Principal Component Analysis or Independent Component Analysis.

To make things even more complicated, clusterings can be roughly distinguished as:

- *Hard clustering*: each object belongs to a cluster or not
- *Soft (or fuzzy) clustering*: each object belongs to each cluster to a certain degree (for example, a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- *Strict partitioning clustering*: each object belongs to exactly one cluster
- *Strict partitioning clustering with outliers*: objects can also belong to no cluster, and are considered outliers
- *Overlapping clustering* (also: *alternative clustering*, *multi-view clustering*): objects may belong to more than one cluster; usually involving hard clusters
- *Subspace clustering*: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap

Main algorithms:

- Hierarchical clustering
- K-Means Clustering

Hierarchical clustering

Hierarchical clustering, the idea is that objects are more related to nearby objects than to objects farther away.

The key factor is distance.

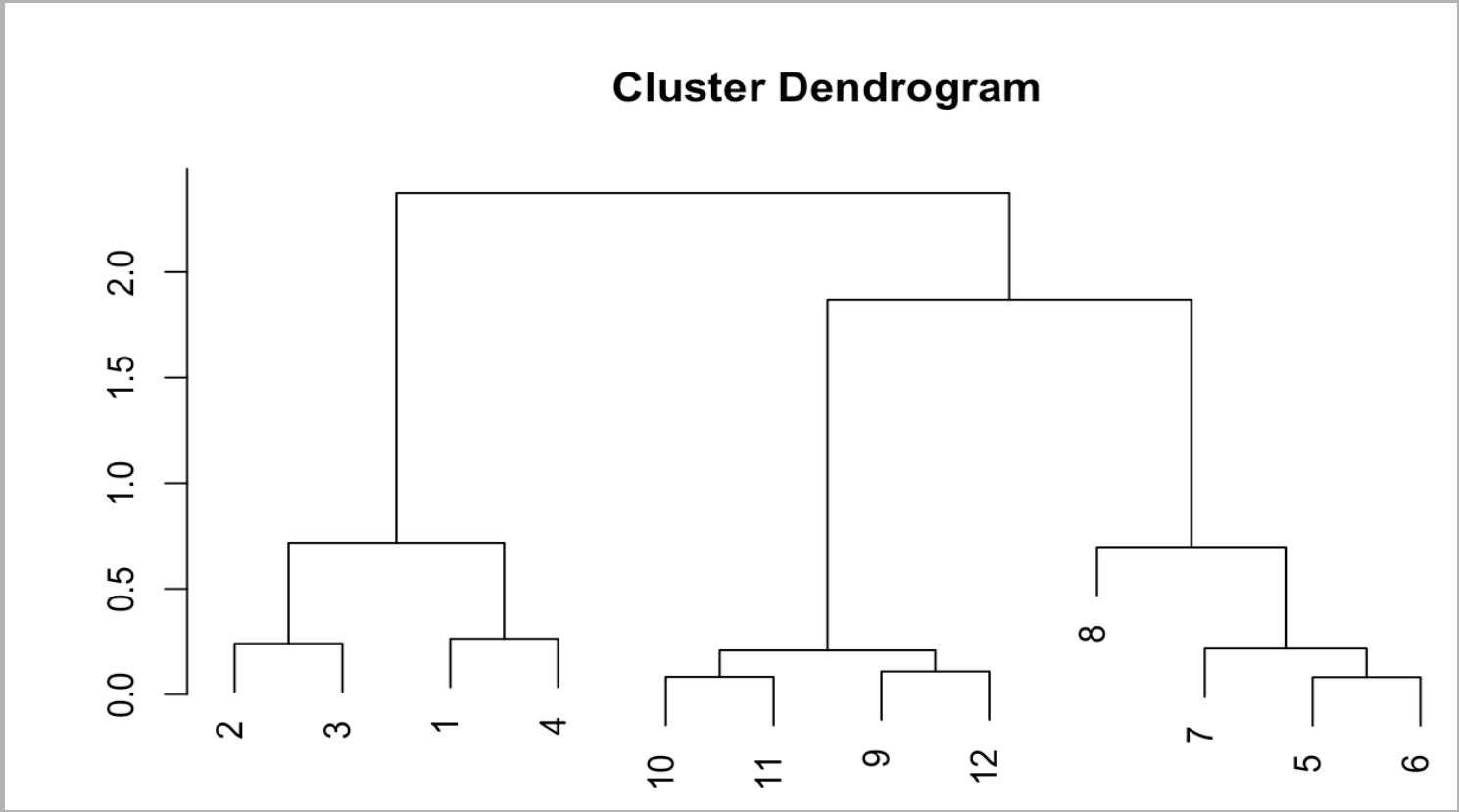
Distances

- Euclidean distance: $||a-b||_2 = \sqrt{(\Sigma(a_i-b_i))}$
- Squared Euclidean distance: $||a-b||_2^2 = \Sigma((a_i-b_i)^2)$
- Manhattan distance: $||a-b||_1 = \Sigma|a_i-b_i|$
- Maximum distance:$||a-b| = max_i|a_i-b_i|$
- Mahalanobis distance: $\sqrt{((a-b)^T S^{-1} (-b))}$

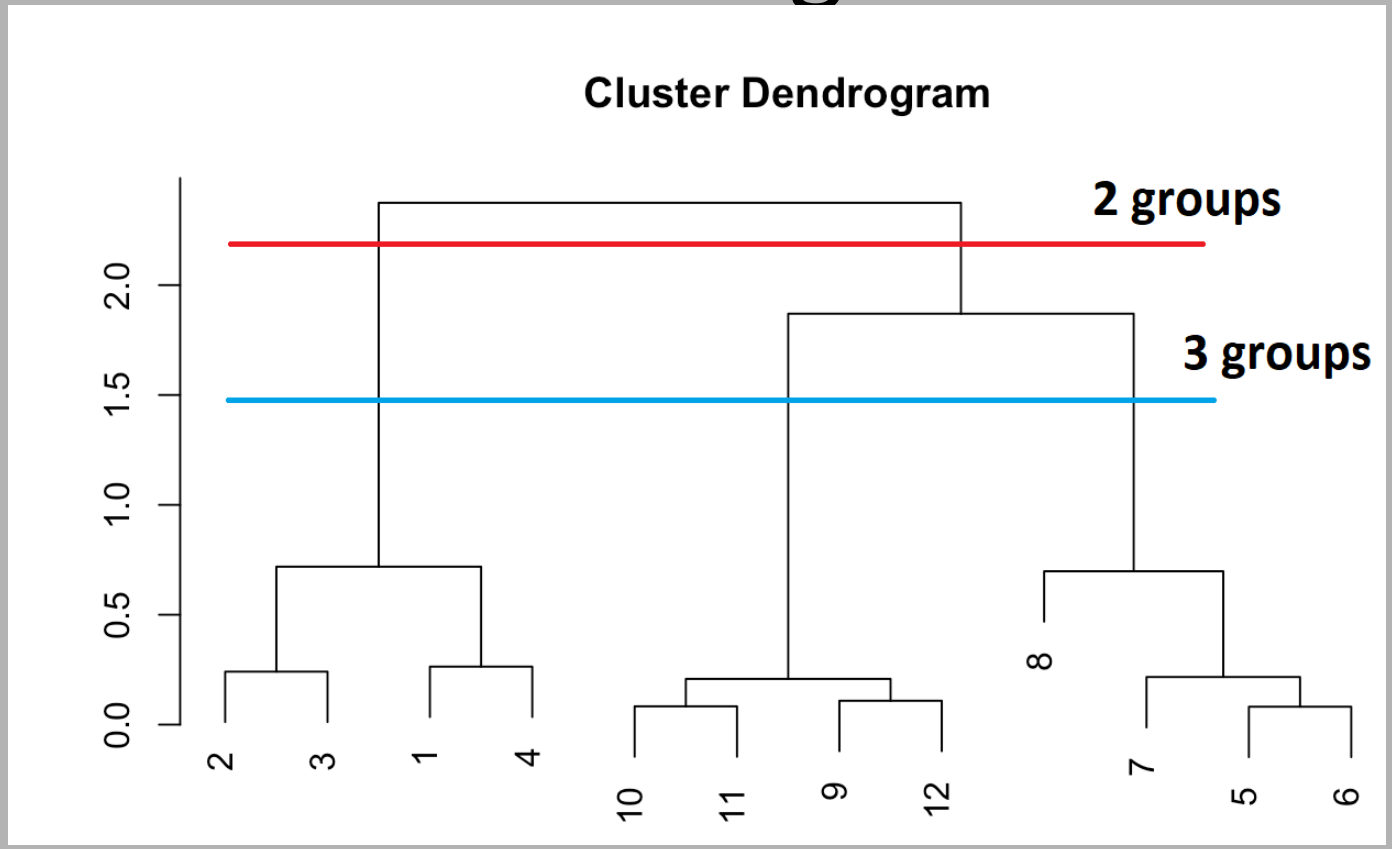
Dendograms

At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances.

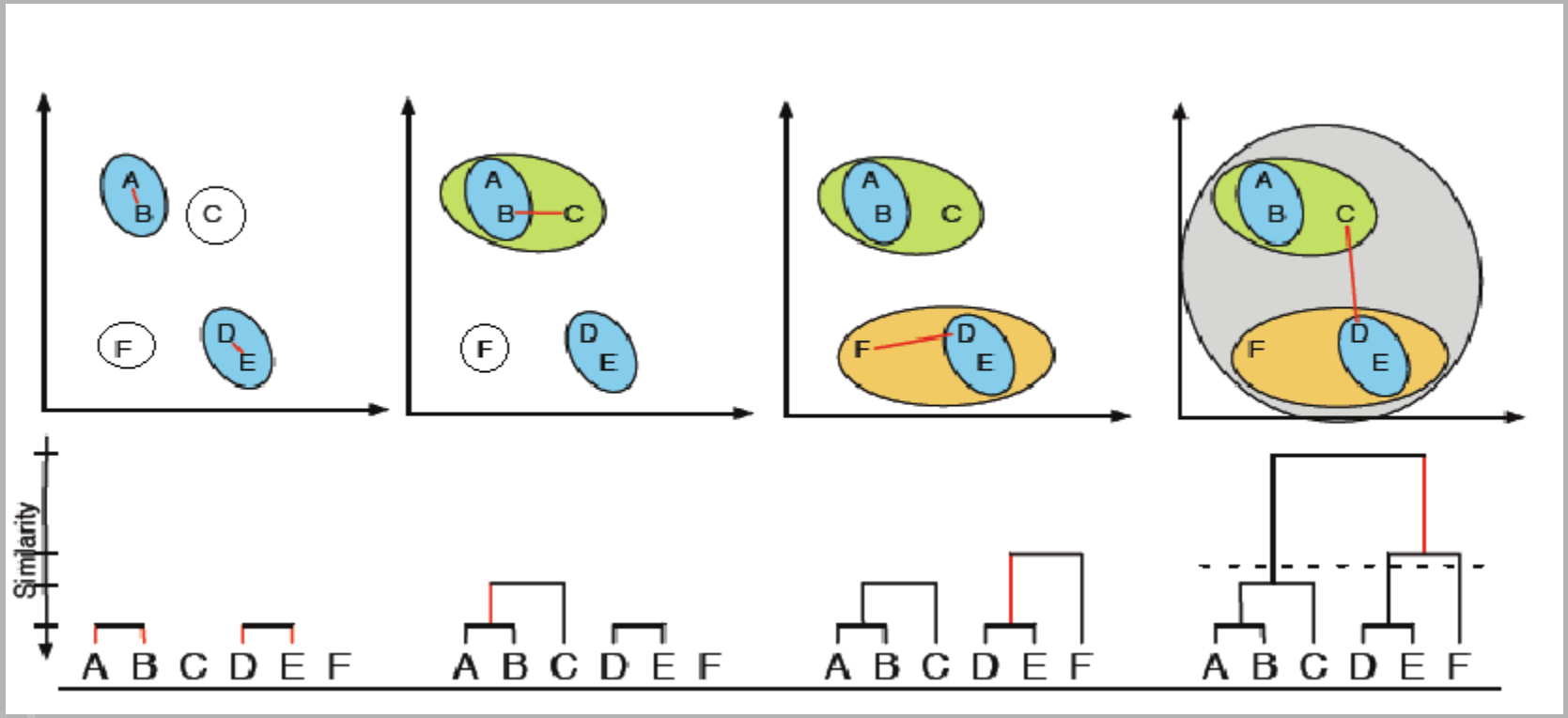
Dendograms



Dendograms


Cluster Dendrogram
2 groups
3 groups
0.0
0.5
1.0
1.5
2.0
2
3
1
4
10
11
9
12
8
7
5
6

Dendograms



Dendograms

In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

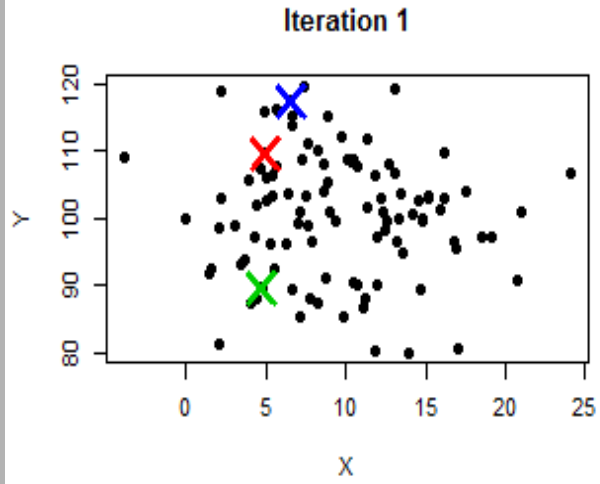
Connectivity-based clustering is a whole family of methods that differ by the way distances are computed.

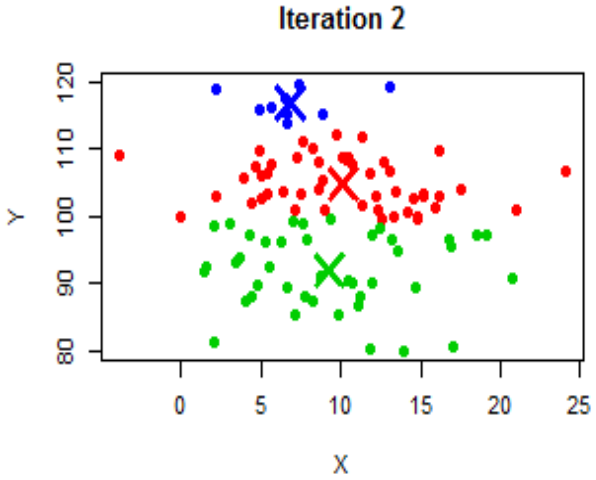
It is hard to define “similar enough”or “good enough”–the answer is typically highly subjective.

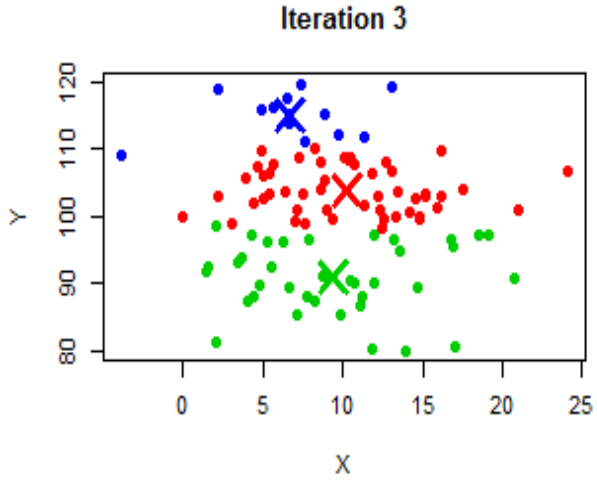
K Means Clustering

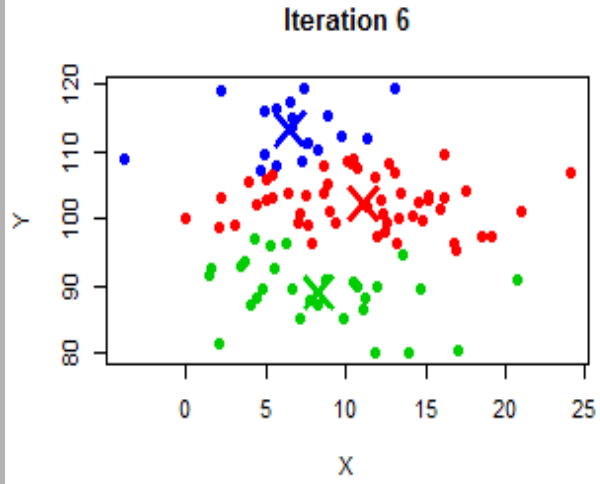
It works in 5 steps :

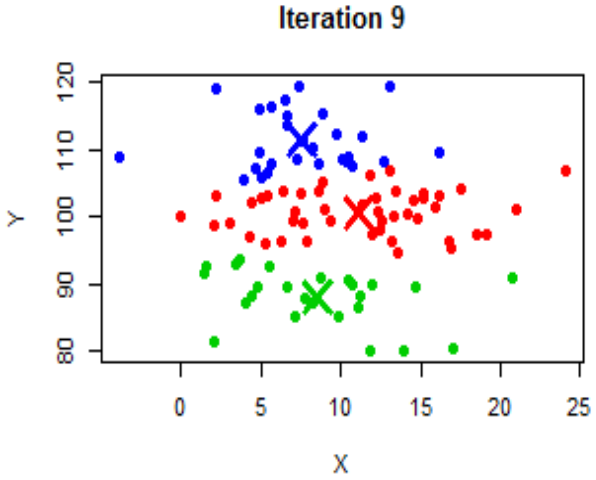
1. Specify the desired number of clusters K
2. Randomly assign each data point to a cluster
3. Compute cluster centroids
4. Re-assign each point to the closest cluster centroid :
5. Re-compute cluster centroids : Now, re-computing the centroids for the clusters.
6. Repeat steps 4 and 5 until no improvements are possible : Similarly, we'll repeat the 4^{th} and 5^{th} steps until we'll reach global optima. When there is no further switching of data points between two clusters , the algorithm terminates.

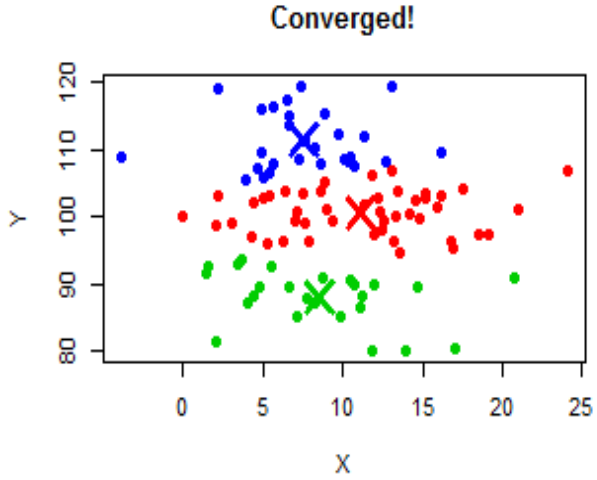

Iteration 1
Y
X
80
90
100
110
120
0
5
10
15
20
25


Iteration 2
Y
X
80
90
100
110
120
0
5
10
15
20
25


Iteration 3
Y
X
80
90
100
110
120
0
5
10
15
20
25


Iteration 6
Y
X
80
90
100
110
120
0
5
10
15
20
25


Iteration 9
Y
X
80
90
100
110
120
0
5
10
15
20
25


Converged!
Y
X
80
90
100
110
120
0
5
10
15
20
25

A brief comparison

- Hierarchical clustering can't handle big data well but K Means clustering can. This is because the complexity of K Means is linear i.e. O(n) while that of hierarchical clustering is quadratic i.e. $O(n^2)$.

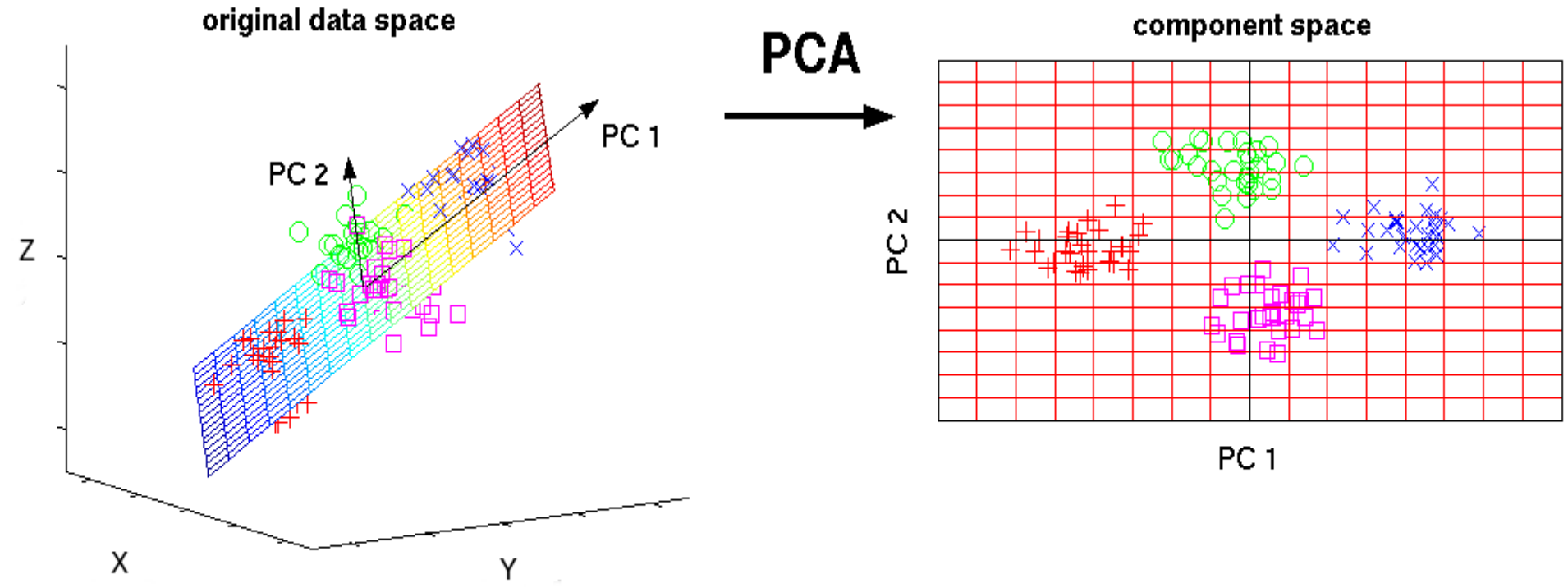
- In K Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.

- K Means is found to work better when the shape of the clusters is hyper spherical

- K Means clustering requires prior knowledge of the number of clusters you want to divide your data into. But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram

For big data, the so-called “curse of dimensionality” (when the dimensionality increases, the volume of the space increases so fast that the available data become sparse) led to new methods like CLIQUE and SUBCLU.

Combining the two methods



Orthogonal projection on the component space

Clustering on the component space

Thank you for your attention !