Visualizing hierarchies

UNSUPERVISED LEARNING IN PYTHON

Visualizations communicate insight

- "t-SNE" : Creates a 2D map of a dataset
- "Hierarchical clustering"

t-SNE (t-Distributed Stochastic Neighbor Embedding)

t-SNE is a **dimensionality reduction technique** designed for visualizing high-dimensional data in a lower-dimensional space (typically 2D or 3D) while preserving the local structure of the data. It is widely used to explore patterns and clusters in complex datasets.
t-SNE is a powerful non-linear dimensionality reduction technique particularly suited for visualizing high-dimensional datasets. While computationally intensive and sensitive to hyperparameters, it remains a popular choice for uncovering clusters and patterns in complex data.

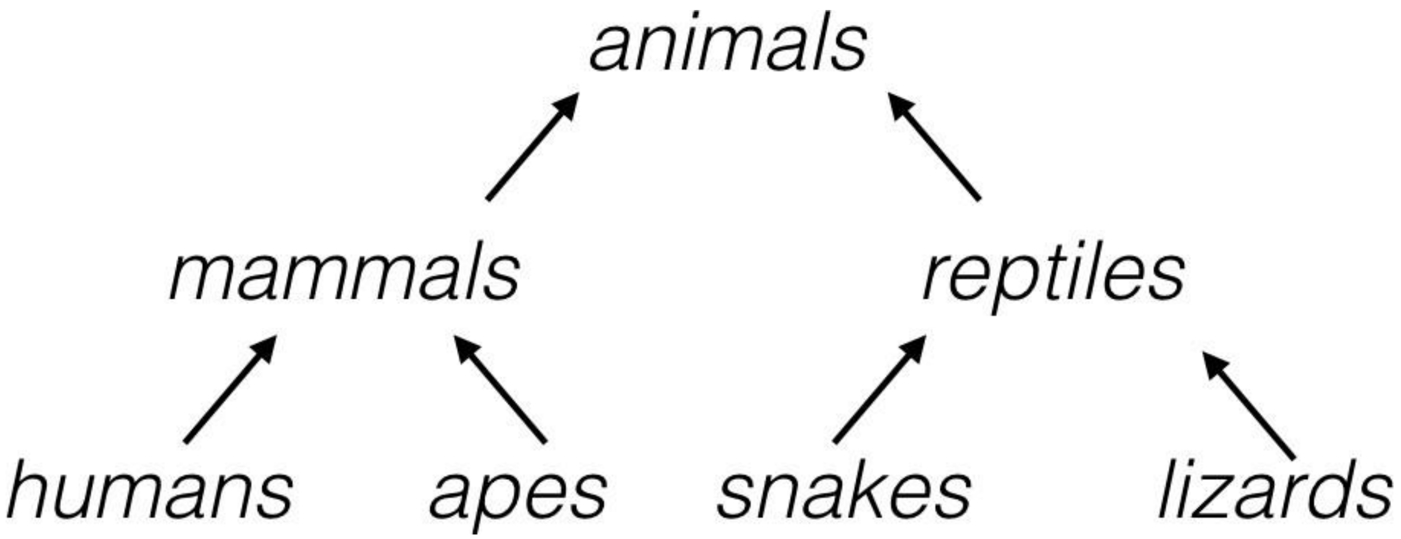
Hierarchical clustering is an unsupervised machine learning algorithm used to group similar data points into clusters. Unlike flat clustering methods (e.g., K-Means), hierarchical clustering creates a tree-like structure called a **dendrogram**, which shows the relationships between clusters at various levels of granularity.

Types of Hierarchical Clustering
1. Agglomerative Clustering (Bottom-Up):
   1. Starts with each data point as its own cluster.
   2. Iteratively merges the two closest clusters until a single cluster is formed (or a stopping criterion is met).

2. Divisive Clustering (Top-Down):
   1. Starts with all data points in one cluster.
   2. Iteratively splits clusters into smaller clusters until each data point is its own cluster (or a stopping criterion is met).

A hierarchy (계층) of groups

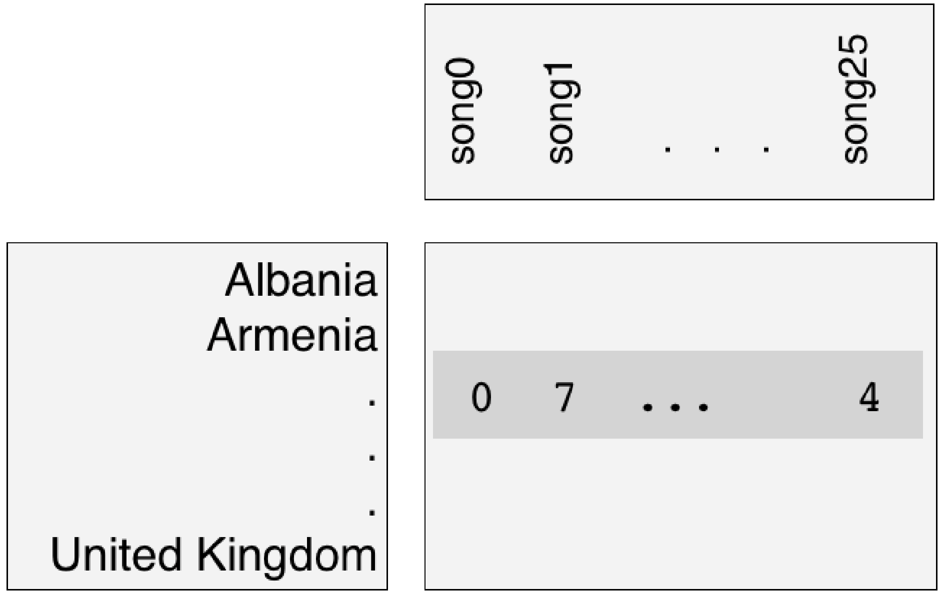
- Groups of living things can form a hierarchy
- Clusters are contained in one another



Eurovision scoring dataset

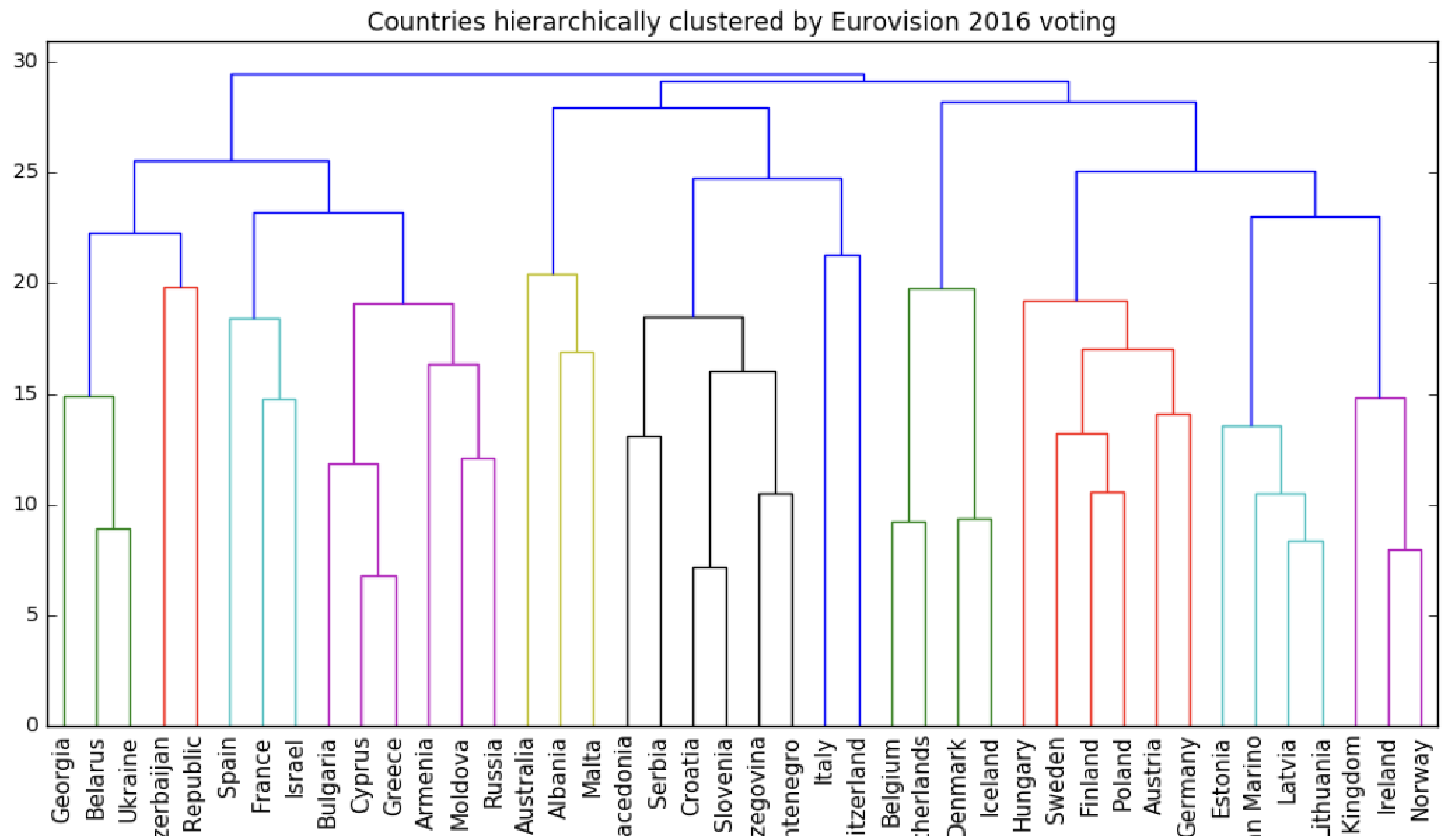
- Countries gave scores to songs performed at the Eurovision 2016
- 2D array of scores
- Rows are countries, columns are songs

유로비전 송 콘테스트(영어: Eurovision Song Contest, 프랑스어: Concours Eurovision de la Chanson)는 유럽방송연맹(European Broadcasting Union) 회원국 시청자 앞에서 노래, 춤 등 자신의 기량을 뽐낸 뒤 순위를 가리는 유럽 최대의 음악 경연 대회이다. 세계에서 가장 시청자 수가 많은 방송중 하나로 2021년에는 1억 8000만명 이상이 시청했다.



[1] https://www.eurovision.tv/page/results

Hierarchical clustering of voting countries



Hierarchical clustering

- Every country begins in a separate cluster
- At each step, the two closest clusters are merged
- Continue until all countries in a single cluster
- This is "agglomerative" hierarchical clustering

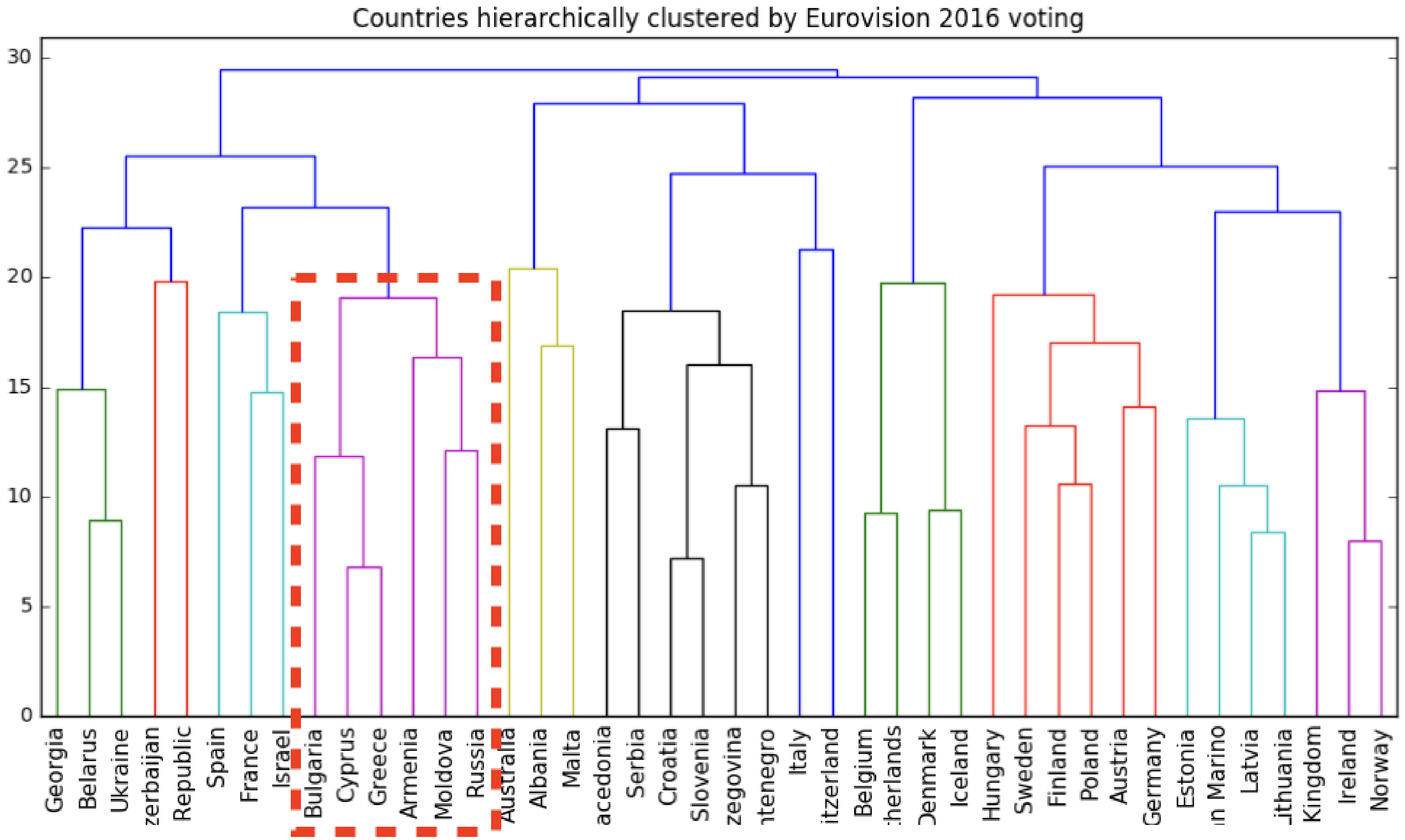
병합 군집agglomerative clustering 알고리즘은 **시작할 때 각 포인트를 하나의 클러스터로 지정**하고, 그다음 **종료 조건을 만족할 때까지 가장 비슷한 두 클러스터를 합침**

종료조건 : 클러스터 갯수, 지정된 갯수의 클러스터가 남을 때까지 비슷한 클러스터를 합침

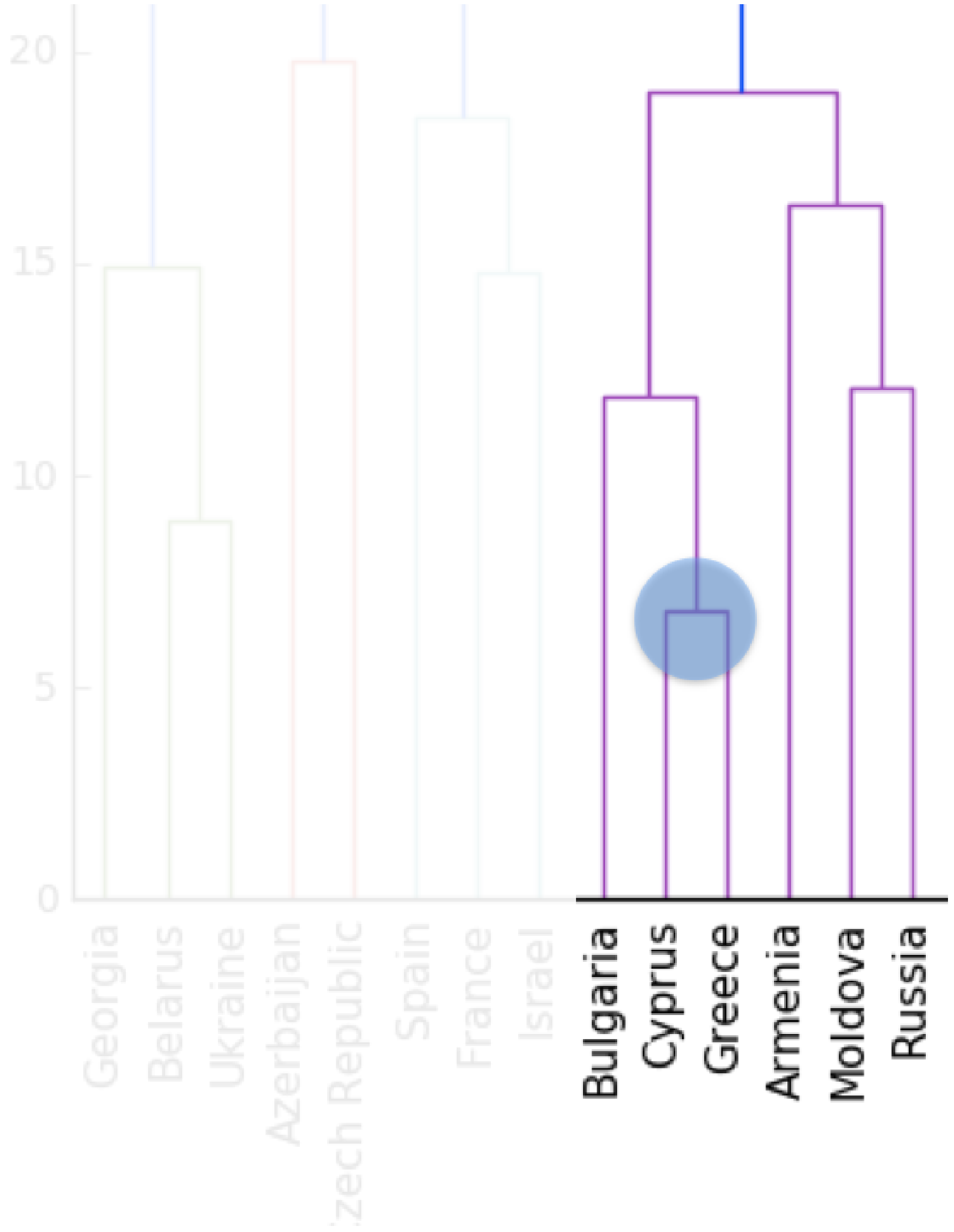
Linkage Methods
To define the distance between clusters, different linkage methods are used:
1. **Single Linkage**:: Distance between the closest pair of points in two clusters.
2. **Complete Linkage**: Distance between the farthest pair of points in two clusters.
3. **Average Linkage**: Average distance between all pairs of points in two clusters.
4. **Centroid Linkage**: Distance between the centroids of two clusters.
5. **Ward's Linkage**: Minimizes the variance within clusters.

The dendrogram of a hierarchical clustering

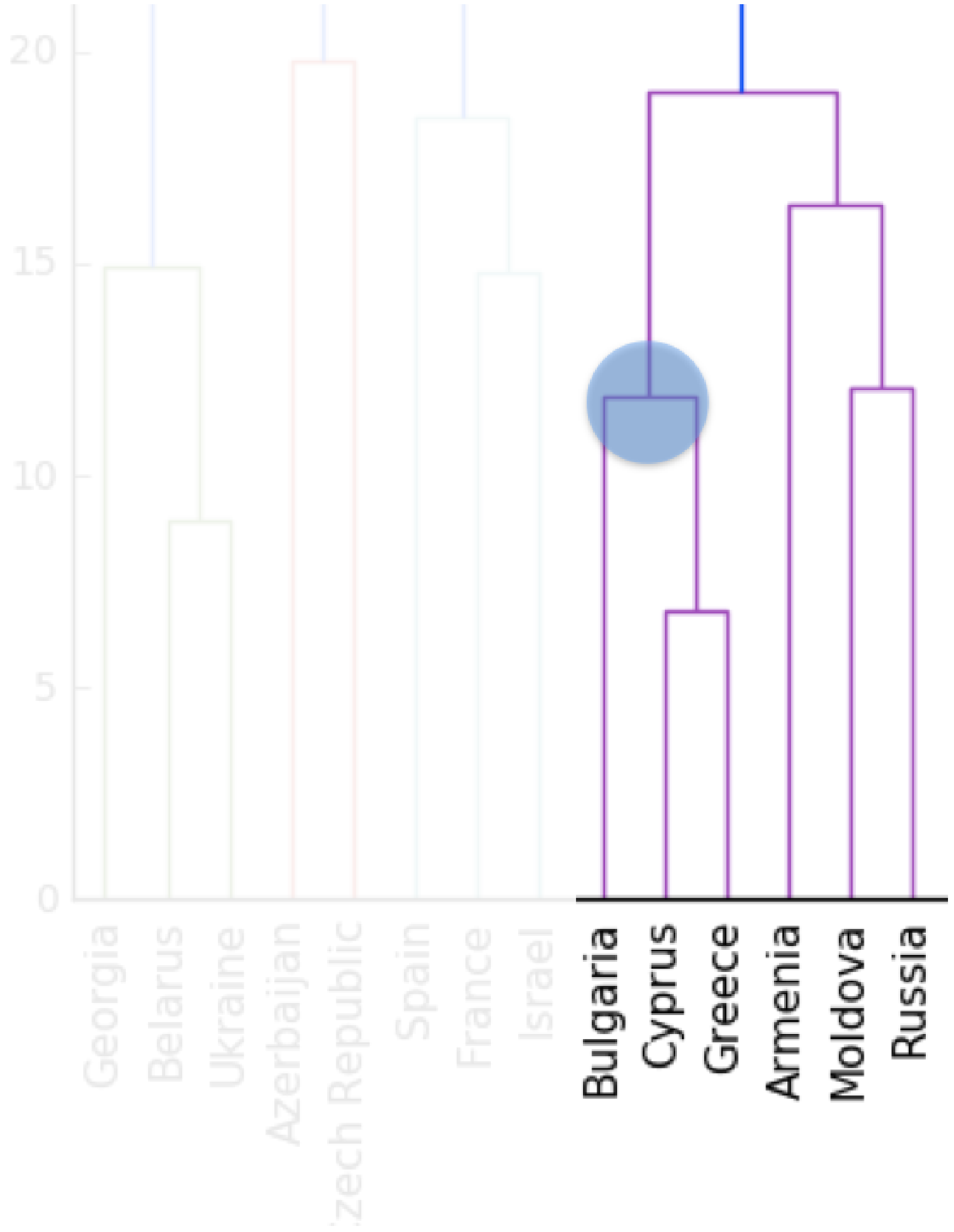
- Read from the bottom up
- Vertical lines represent clusters



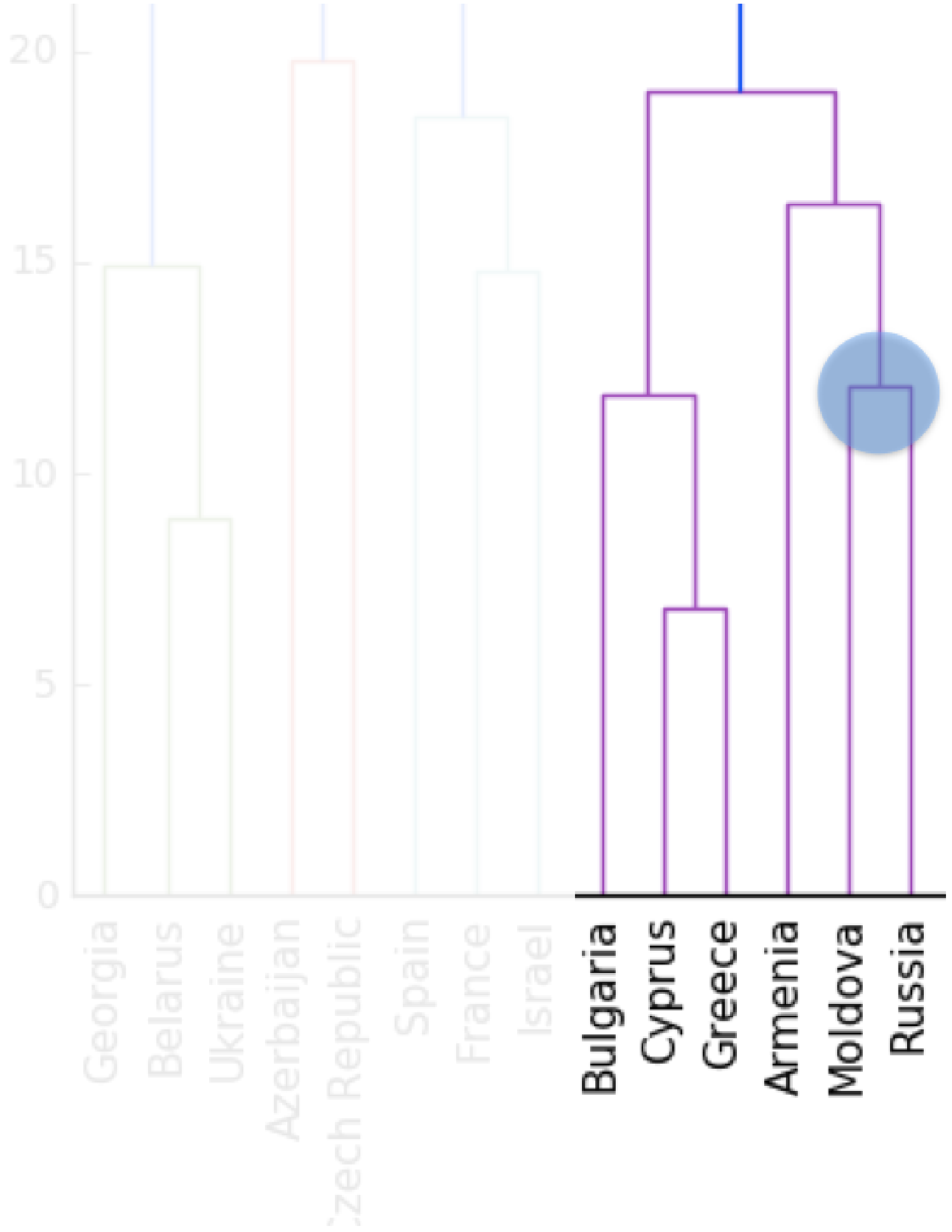
Dendrograms, step-by-step



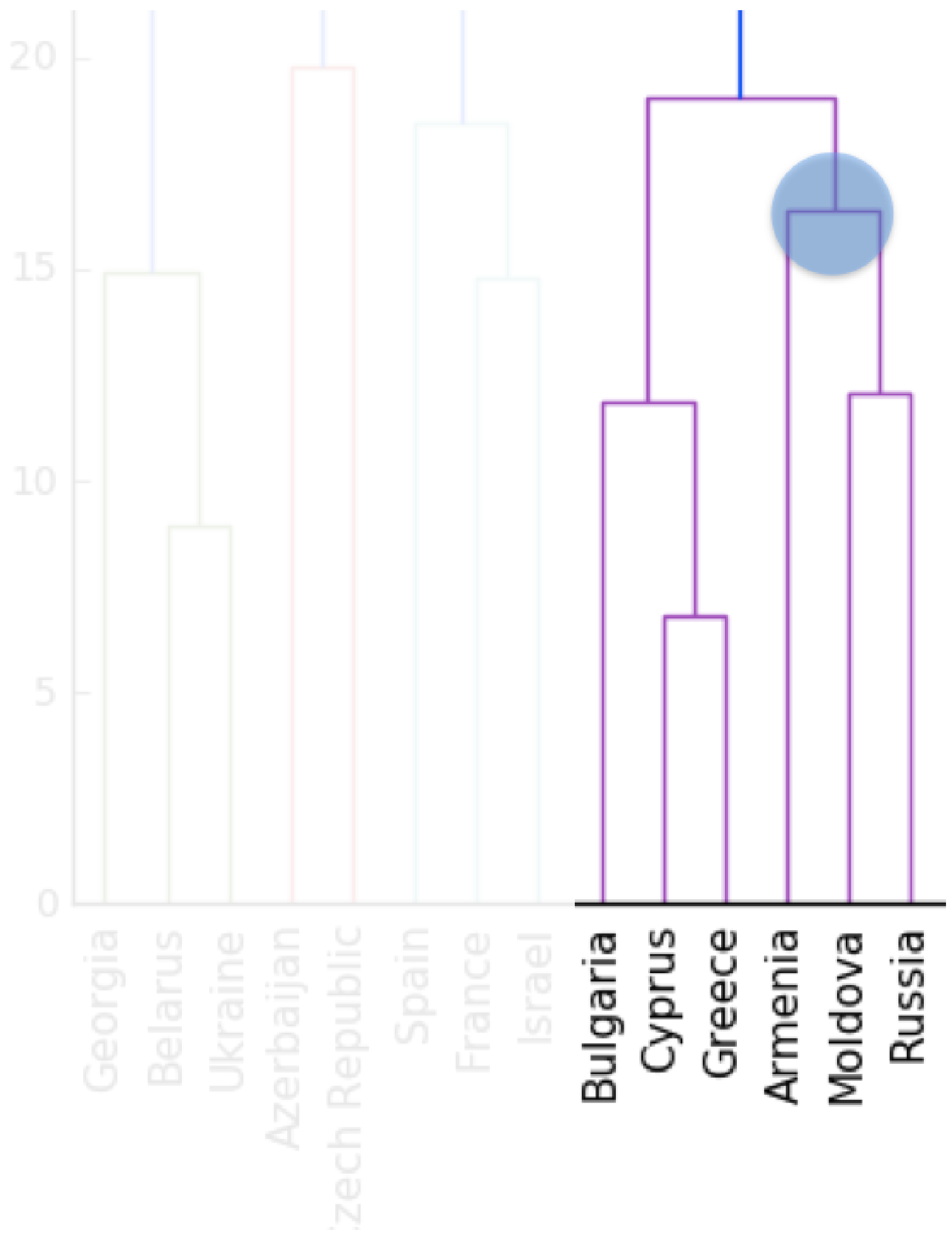
Dendrograms, step-by-step



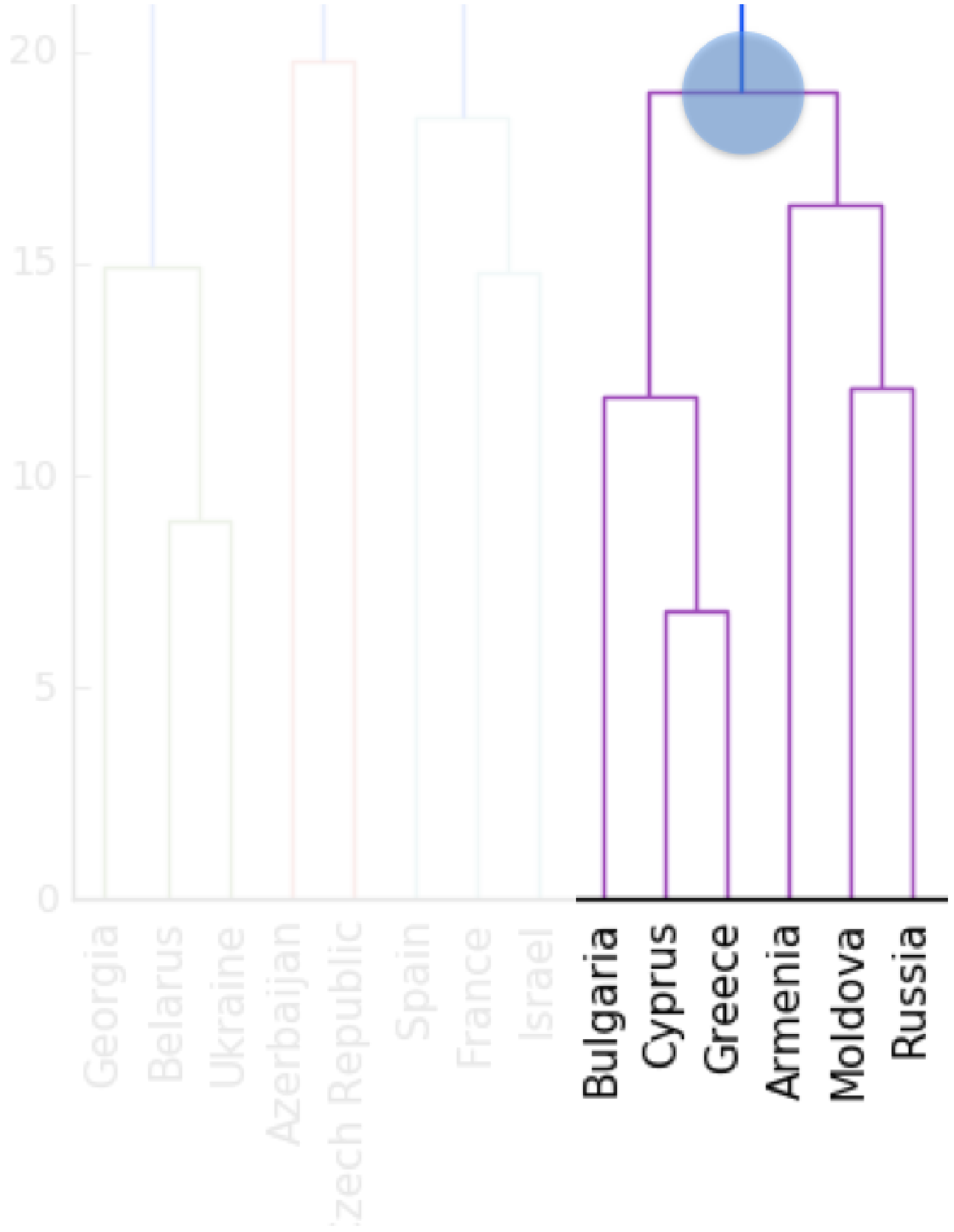
Dendrograms, step-by-step



Dendrograms, step-by-step



Dendrograms, step-by-step



Hierarchical clustering with SciPy

- Given samples (the array of scores), and country_names

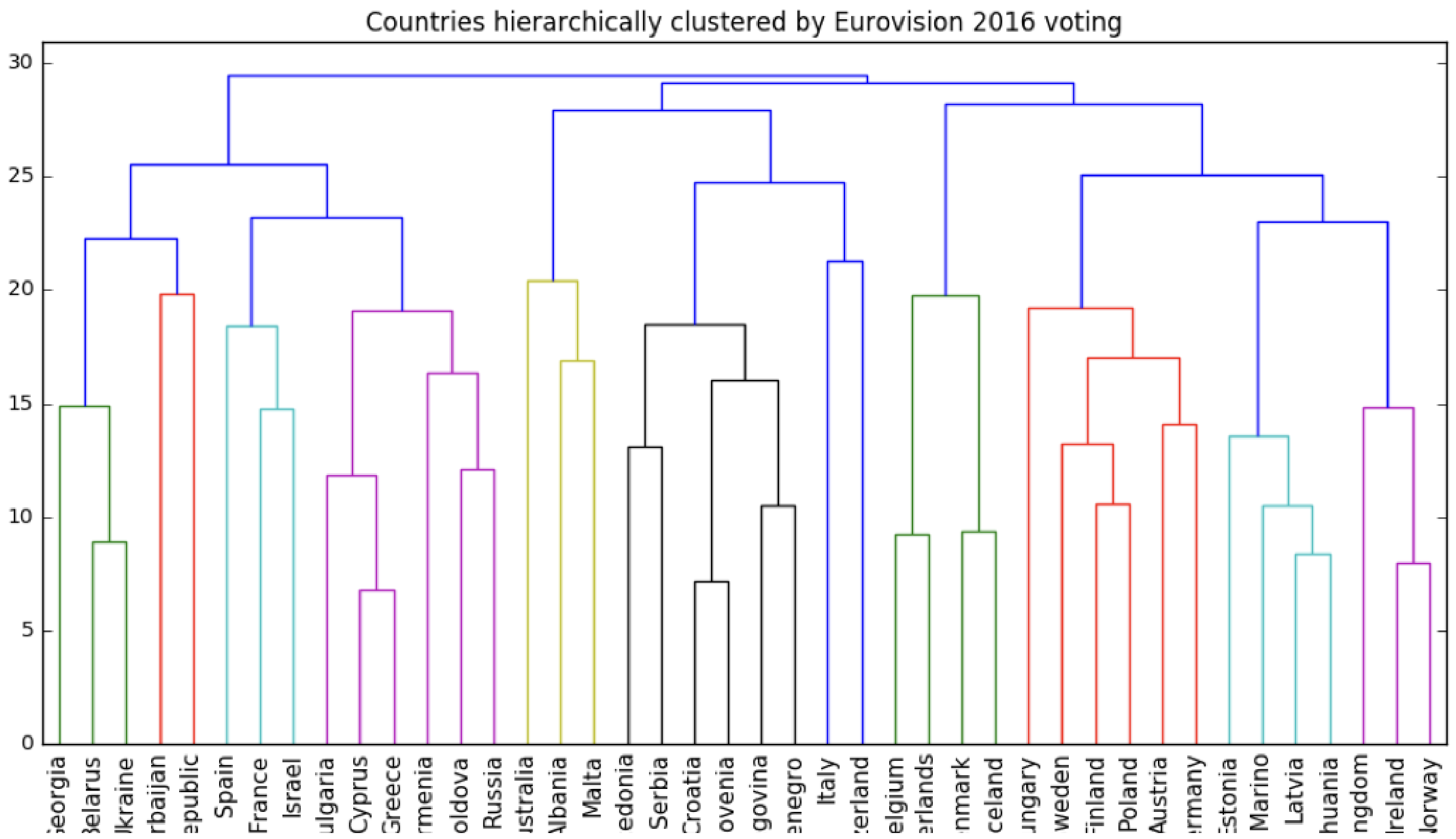
```
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage, dendrogram
mergings = linkage(samples, method='complete')
dendrogram(mergings,
           labels=country_names,
           leaf_rotation=90,
           leaf_font_size=6)
plt.show()
```

Cluster labels in hierarchical clustering

UNSUPERVISED LEARNING IN PYTHON

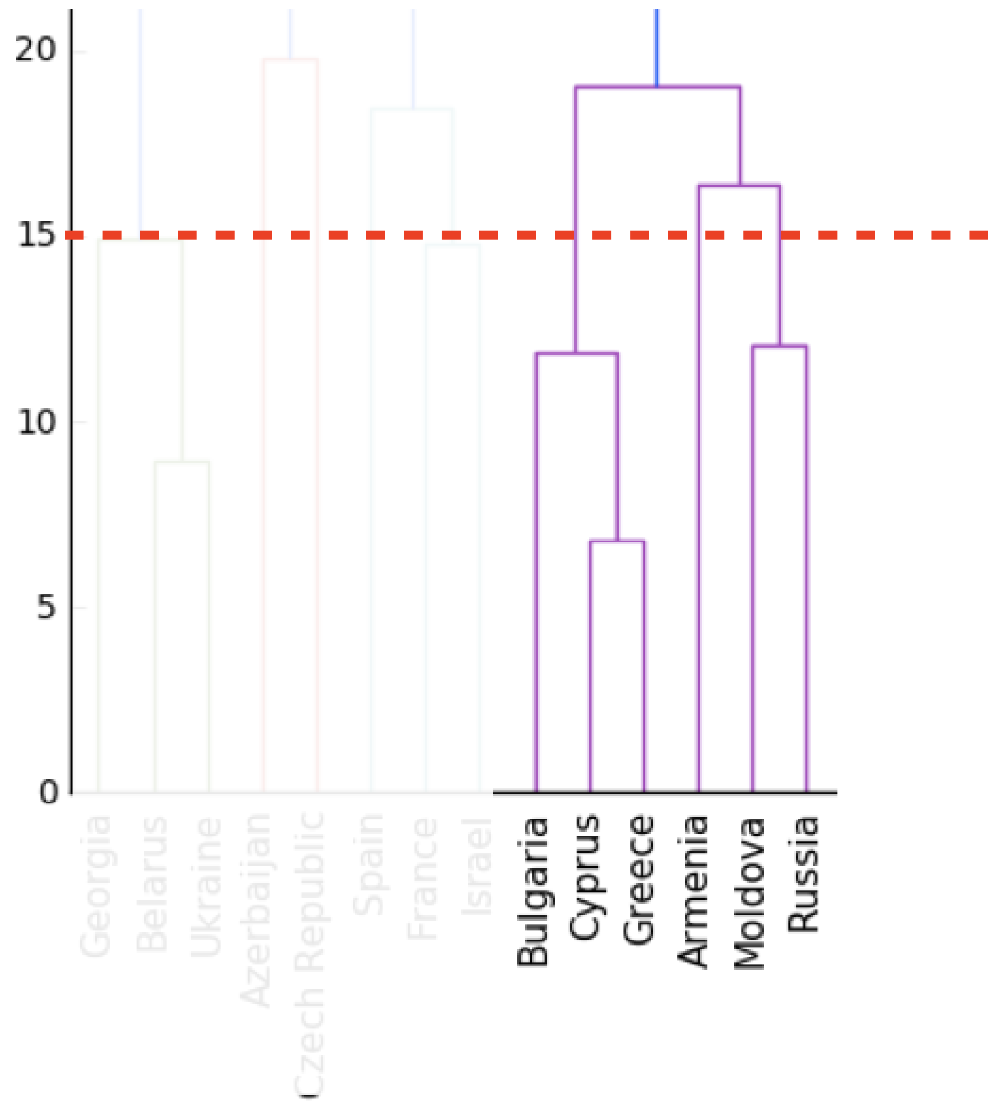
Cluster labels in hierarchical clustering

- Not only a visualization tool!
- Cluster labels at any intermediate stage can be recovered
- For use in e.g. cross-tabulations



Intermediate clusterings & height on dendrogram

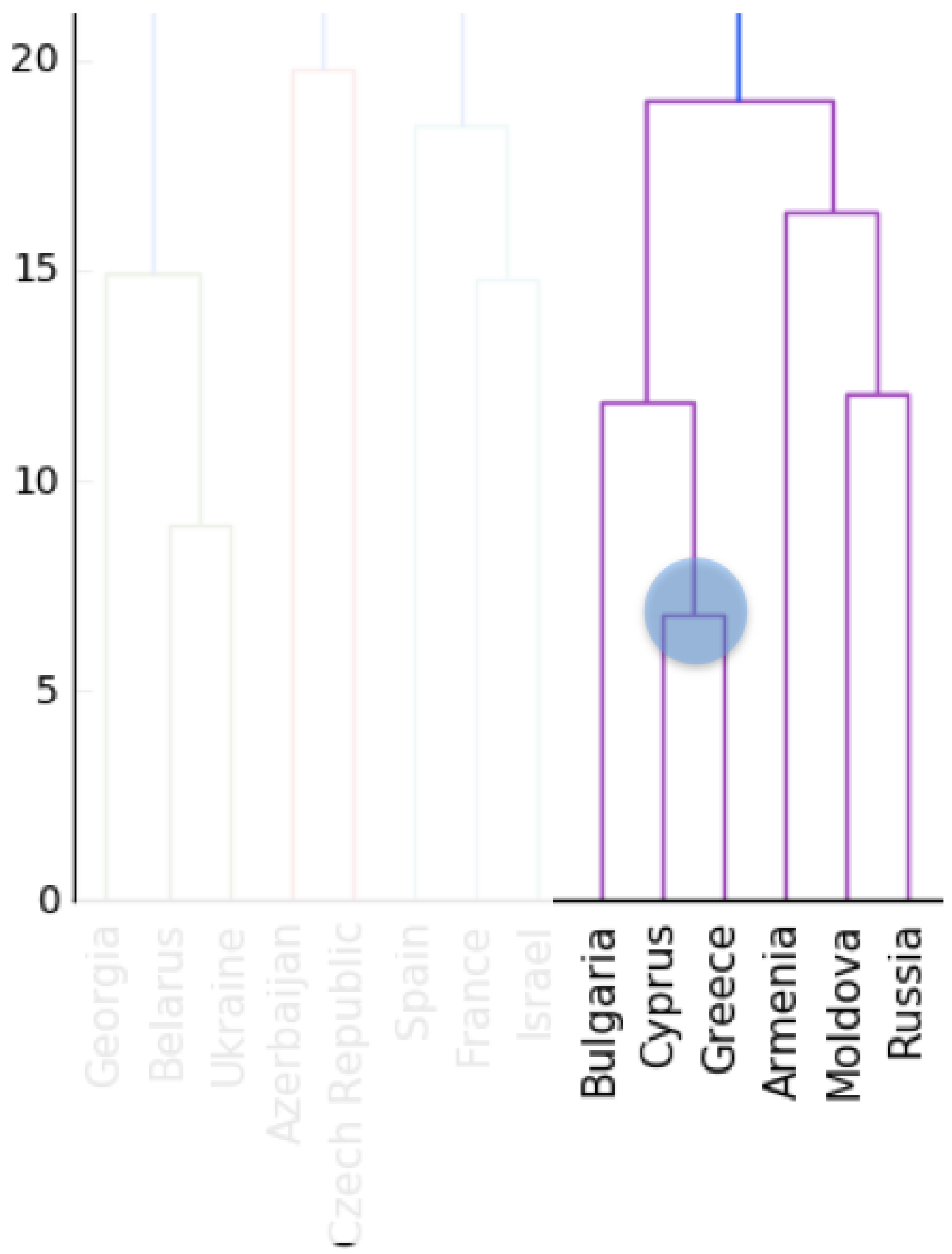
- E.g. at height 15:
  - Bulgaria, Cyprus, Greece are one cluster
  - Russia and Moldova are another
  - Armenia in a cluster on its own



덴드로그램(Dendrogram)은 나무를 나타내는 다이어그램이다.
계층적 군집화에서는 해당 분석에 의해 생성된 클러스터의 배열을 보여준다.

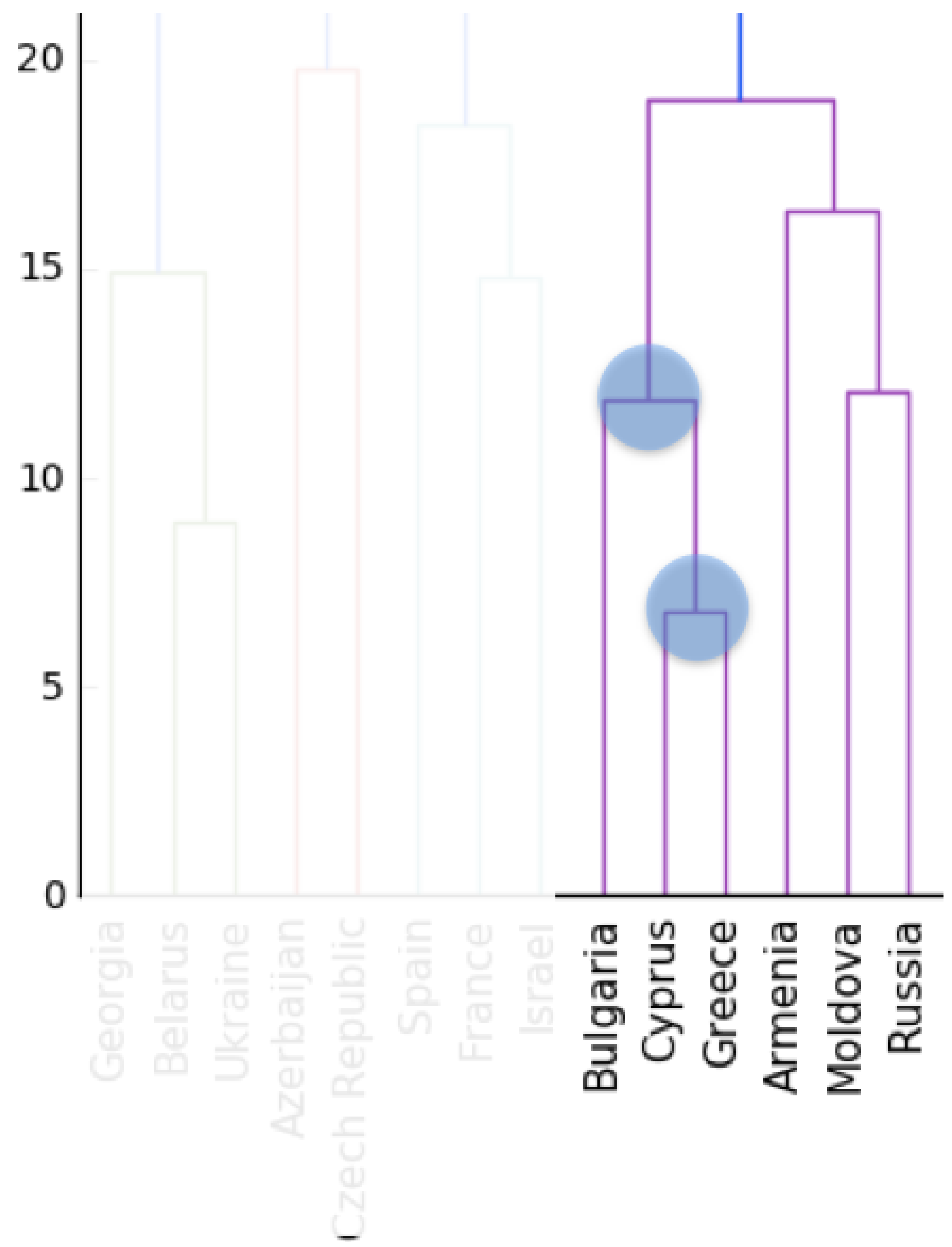
Dendrograms show cluster distances

- Height on dendrogram = distance between merging clusters
- E.g. clusters with only Cyprus and Greece had distance approx. 6



Dendrograms show cluster distances

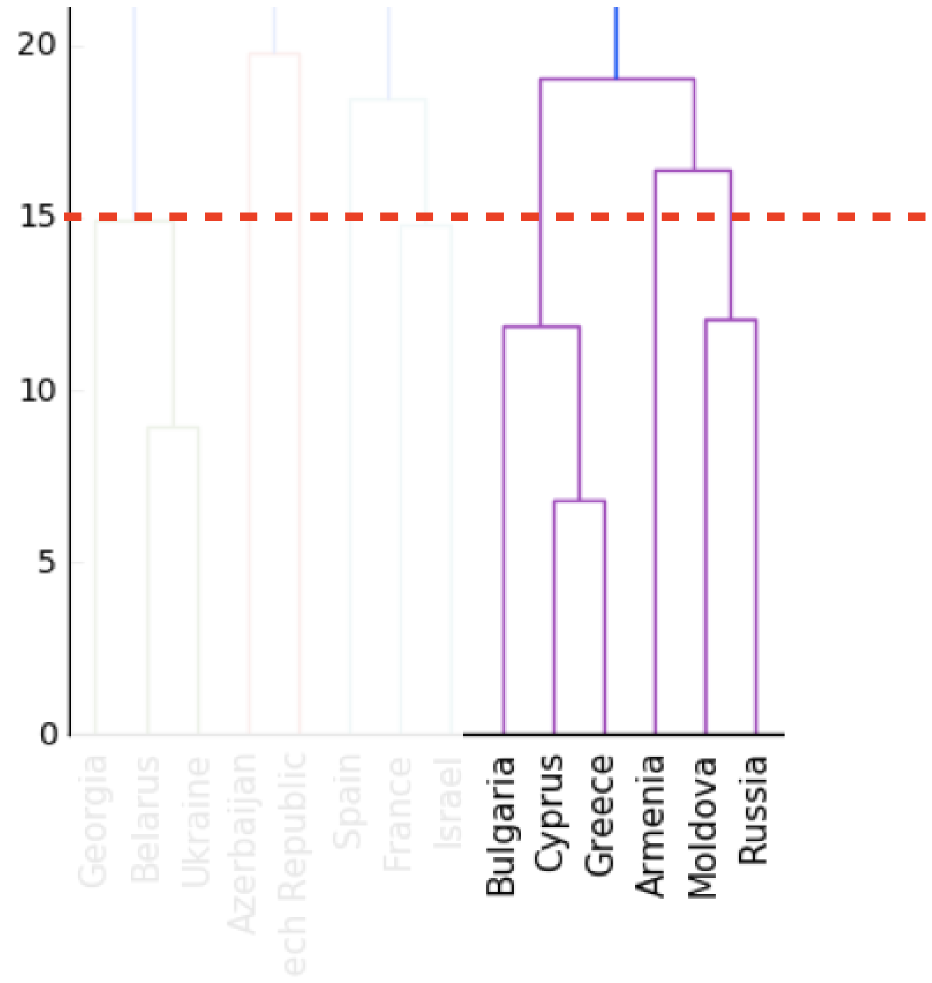
- Height on dendrogram = distance between merging clusters
- E.g. clusters with only Cyprus and Greece had distance approx. 6
- This new cluster distance approx. 12 from cluster with only Bulgaria



Intermediate clusterings & height on dendrogram

- Height on dendrogram specifies max. distance between merging clusters

- Don't merge clusters further apart than this (e.g. 15)



Distance between clusters

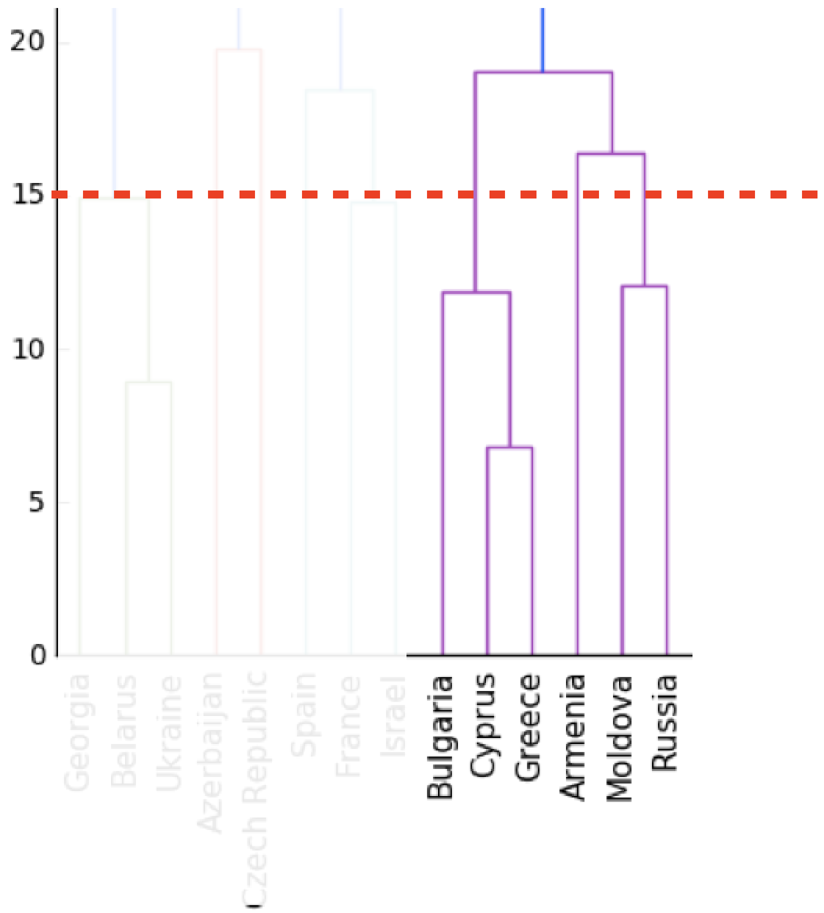
- Defined by a "linkage method"
- In "complete" linkage: distance between clusters is max. distance between their samples

  Specified via method parameter, e.g. linkage(samples, method="complete")
- Different linkage method, different hierarchical clustering!

Complete Linkage: Distance between the farthest pair of points in two clusters.

Extracting cluster labels

- Use the `fcluster()` function
- Returns a NumPy array of cluster labels



Extracting cluster labels using fcluster

```
from scipy.cluster.hierarchy import linkage
mergings = linkage(samples, method='complete')
from scipy.cluster.hierarchy import fcluster
labels = fcluster(mergings, 15, criterion='distance') print(labels)
```

```
[ 9 8 11 20 2 1 17 14 ... ]
```

Aligning cluster labels with country names

Given a list of strings `country_names`:

```
import pandas as pd
pairs = pd.DataFrame({'labels': labels, 'countries': country_names}) print(pairs.sort_values('labels'))
```

```
              countries    labels
5               Belarus         1
40              Ukraine         1
...
36                Spain         5
8              Bulgaria         6
19               Greece         6
10               Cyprus         6
28              Moldova         7
...
```

t-SNE for 2-dimensional maps

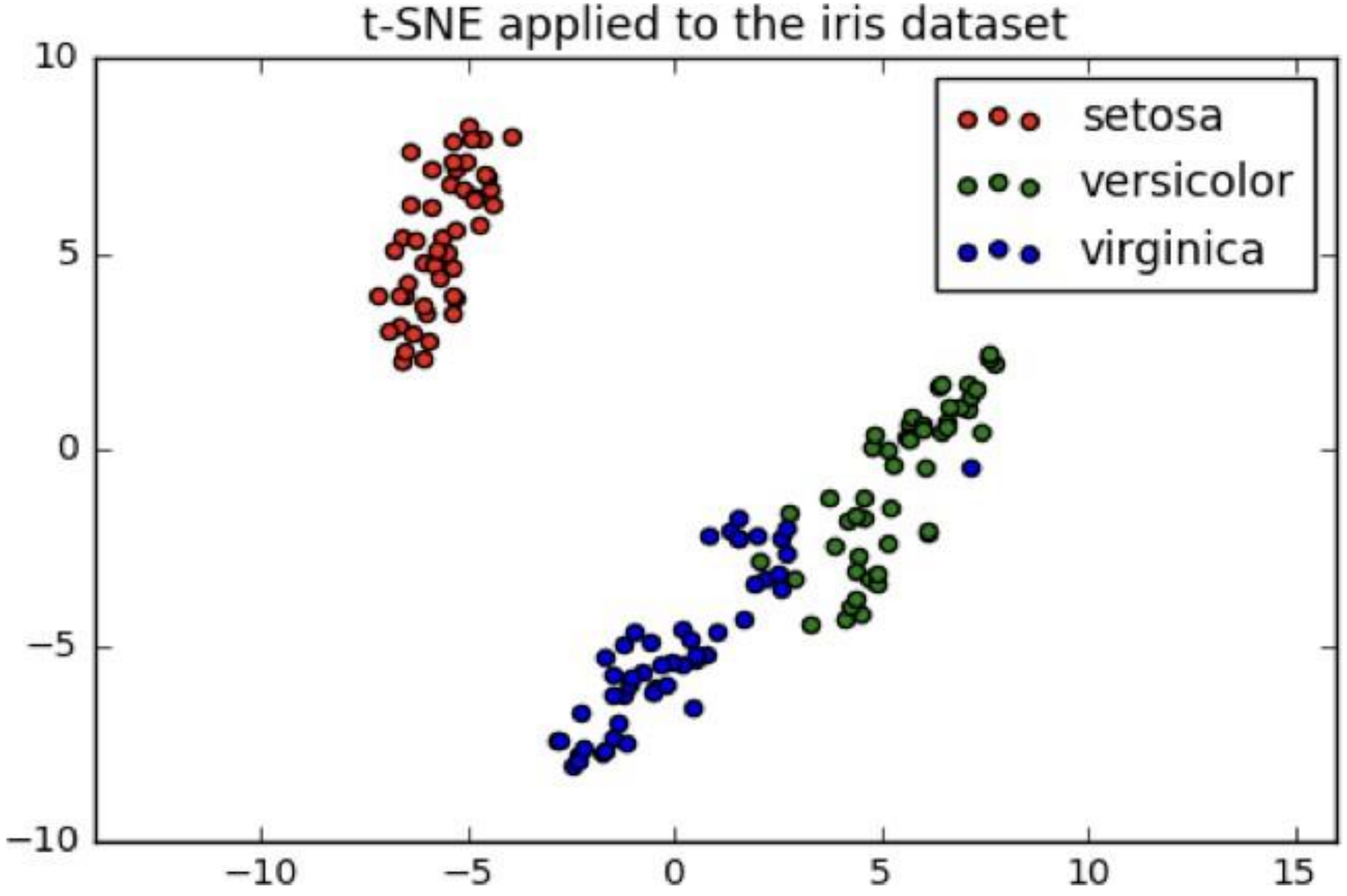
UNSUPERVISED LEARNING IN PYTHON

t-SNE for 2-dimensional maps

- t-SNE = "t-distributed stochastic neighbor embedding"
- Maps samples to 2D space (or 3D)
- Map approximately preserves nearness of samples
- Great for inspecting datasets

t-SNE on the iris dataset

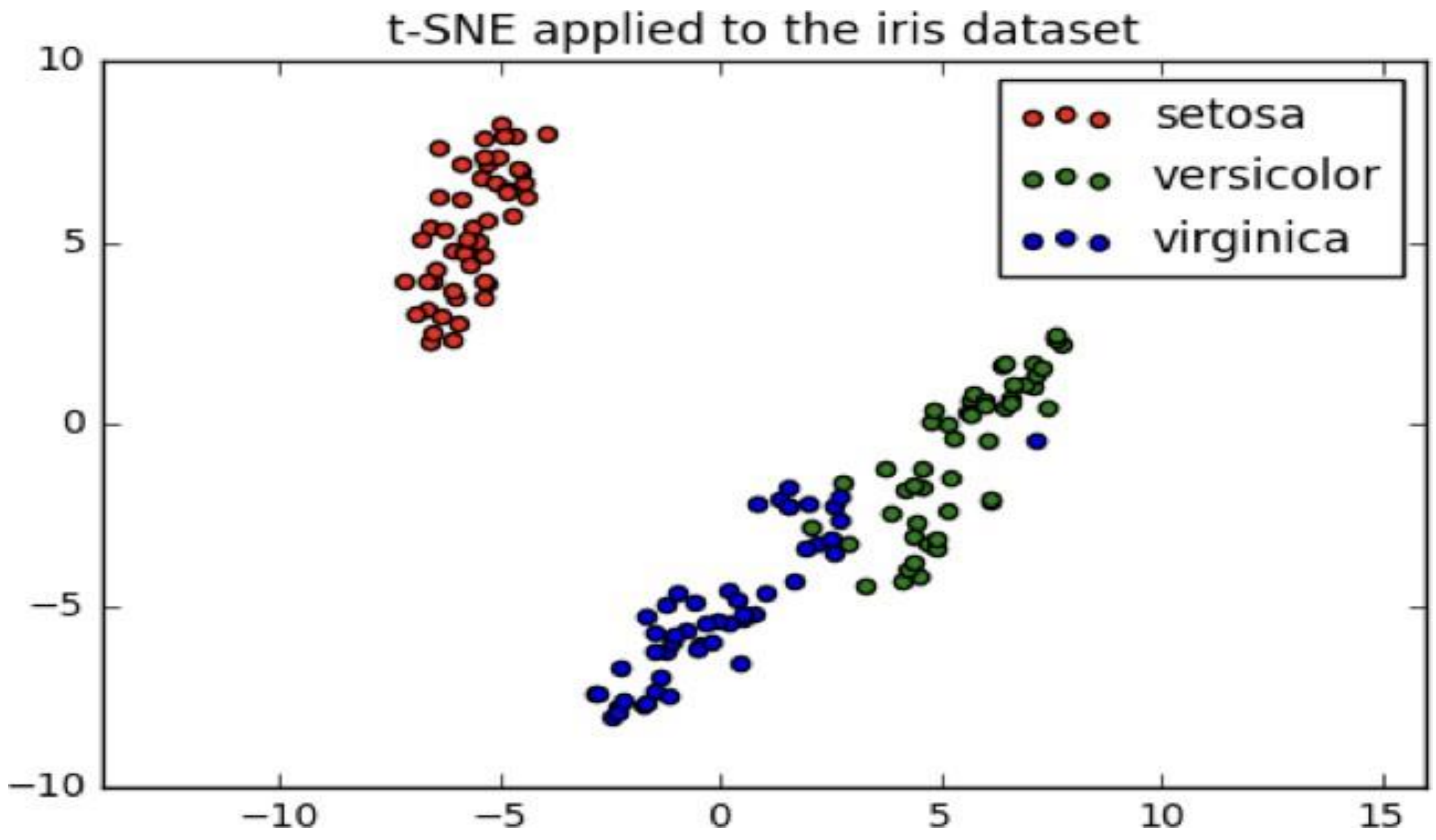
- Iris dataset has 4 measurements, so samples are 4-dimensional
- t-SNE maps samples to 2D space
- t-SNE didn't know that there were different species
- ... yet kept the species mostly separate



Interpreting t-SNE scatter plots

- "versicolor" and "virginica" harder to distinguish from one another

  Consistent with k-means inertia plot: could argue for 2 clusters, or for 3
-



t-SNE in sklearn

- 2D NumPy array `samples`

```
print(samples)
```

```
[[  5.     3.3    1.4    0.2]
 [  5.     3.5    1.3    0.3]
 [  4.9    2.4    3.3    1. ]
 [  6.3    2.8    5.1    1.5]
...
[ 4.9     3.1    1.5    0.1]]
```

- List `species` giving species of labels as number (0, 1, or 2)

```
print(species)
```

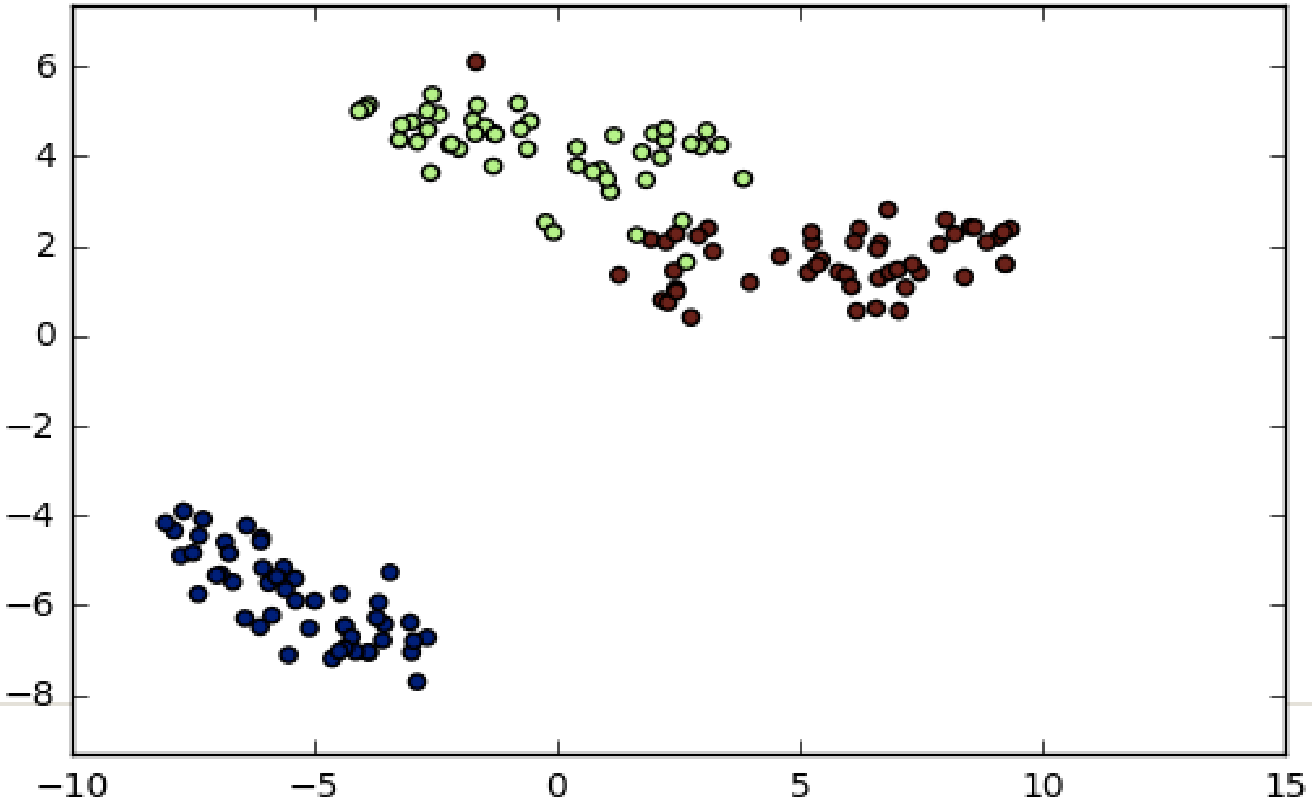
```
[0, 0, 1, 2, ..., 0]
```

t-SNE in sklearn

```
import matplotlib.pyplot as plt
from sklearn.manifold import TSNE
model = TSNE(learning_rate=100) transformed =
model.fit_transform(samples) xs = transformed[:,0]
ys = transformed[:,1] plt.scatter(xs, ys,
c=species) plt.show()
```

The learning rate determines how fast or slow the t-SNE algorithm adjusts the positions of points in the lower-dimensional space during optimization. A learning rate that is too low or too high may lead to suboptimal results.

transformed[:, 0]:The first column of the transformed array, representing the x-coordinates of the 2D projection.
transformed[:, 1]:The second column of the transformed array, representing the y-coordinates of the 2D projection.



t-SNE has only fit_transform()

- Has a `fit_transform()` method
- Simultaneously fits the model and transforms the data
- Has no separate `fit()` or `transform()` methods Can't
- extend the map to include new data samples Must
- start over each time!

t-SNE learning rate

- Choose learning rate for the dataset
- Wrong choice: points bunch together
- Try values between 50 and 200

Different every time

- t-SNE features are different every time
- Piedmont wines, 3 runs, 3 different scatter plots!
- ... however: The wine varieties (=colors) have same position relative to one another