ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Ενότητα 4 – Unsupervised Learning

Μέθοδοι Μηχανικής Μάθησης στη Χρηματοοικονομική

Αθανάσιος Σάκκας, Επ. Καθηγητής, ΟΠΑ

Unsupervised Learning

- Στο Unsupervised Learning δεν προσπαθούμε να προβλέψουμε τίποτα.
- Ο στόχος είναι να ομαδοποιήσουμε δεδομένα για να βελτιώσουμε την κατανόησή μας για το περιβάλλον.

Παράδειγμα: Ομαδοποίηση (Clustering) πελατών

- Ας υποθέσουμε ότι είστε τράπεζα και έχετε εκατοντάδες χιλιάδες πελάτες και 100 χαρακτηριστικά που περιγράφουν το καθένα.

- Οι αλγόριθμοι unsupervised learning μπορούν να χρησιμοποιηθούν για να χωρίσετε τους πελάτες σας σε ομάδες, ώστε να μπορείτε να προβλέψετε τις ανάγκες τους και να επικοινωνήσετε μαζί τους πιο αποτελεσματικά.

k-means algorithm

Τα βήματα που απαιτούνται για την ομαδοποίηση

A. Scaling

Πριν χρησιμοποιήσετε αλγόριθμους ML (συμπεριλαμβανομένων αυτών για unsupervised learning), είναι σημαντικό να κάνετε scaling τις τιμές των χαρακτηριστικών έτσι ώστε να είναι συγκρίσιμες.

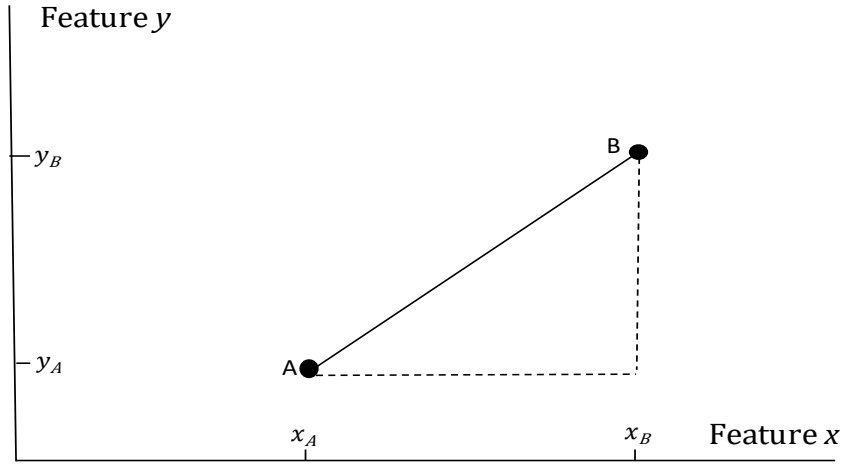
Δύο τρόποι

α. Z-score scaling: $\text{Value} \rightarrow \frac{\text{Value}-\text{Mean}}{\text{SD}}$

β. Min-Max scaling: $\text{Value} \rightarrow \frac{\text{Value}-\text{Minimum}}{\text{Maximum}-\text{Minimum}}$

Β. Μέτρο απόστασης

- Το απλούστερο μέτρο απόστασης είναι το μέτρο της Ευκλείδειας απόστασης.
- Απόσταση (Distance) = $\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$



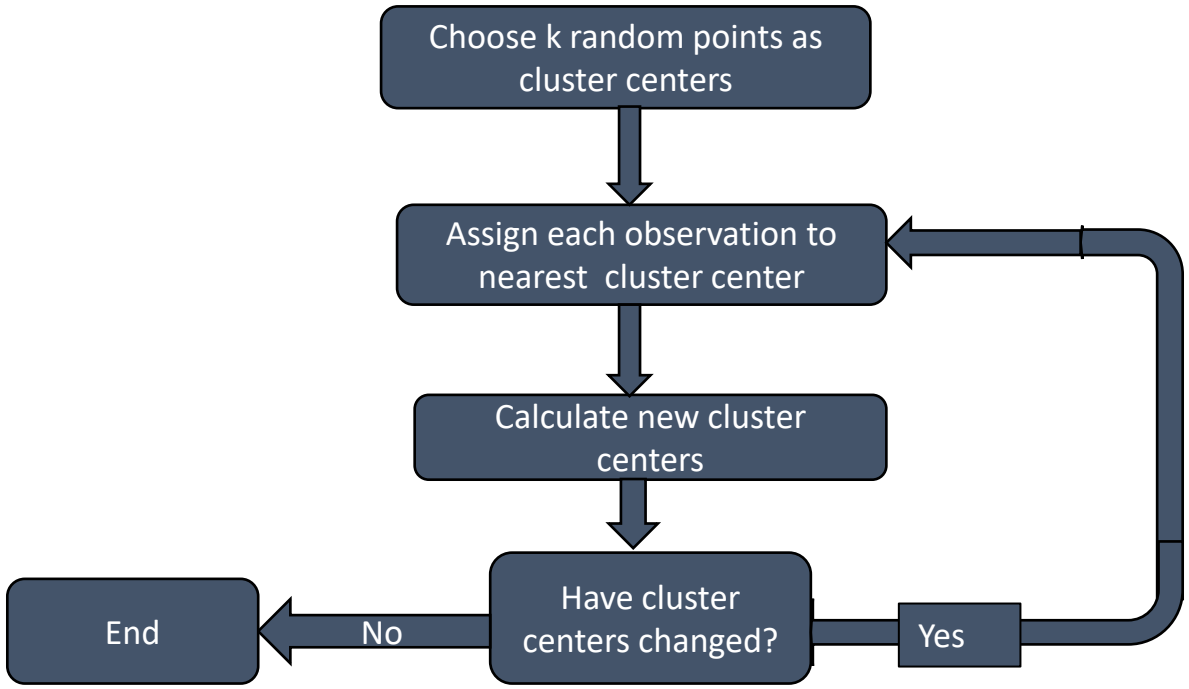
- Γενικά όταν υπάρχουν m χαρακτηριστικά η απόσταση μεταξύ P και Q είναι $\sqrt{\sum_{j=1}^{m}(v_{pj} - v_{qj})^2}$, όπου v_{pj} και v_{qj} οι τιμές values του jth χαρακτηριστικών P and Q

Γ. Cluster Centers

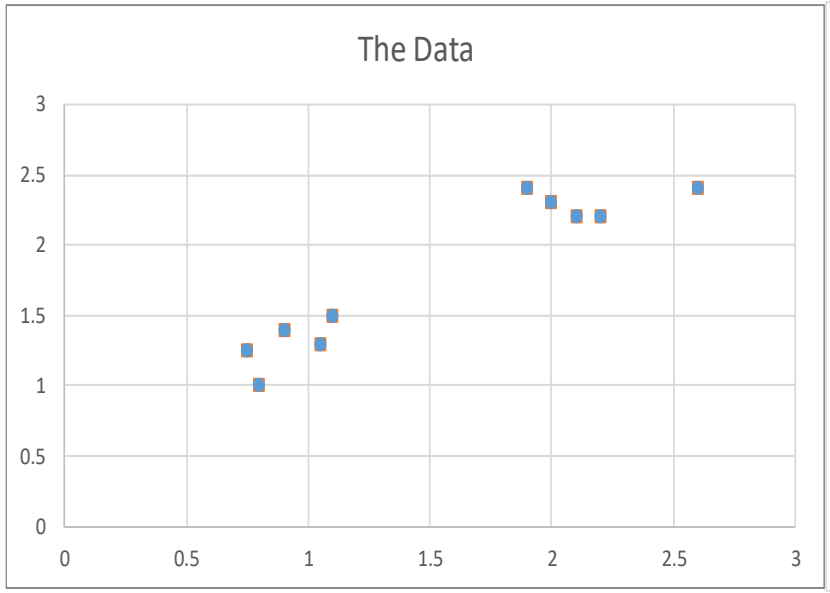
- Το κέντρο μια ομάδας (μερικές φορές ονομάζεται κεντροειδές - centroid) προσδιορίζεται με τον μέσο όρο των τιμών κάθε χαρακτηριστικού για όλα τα σημεία της ομάδας.
- Παράδειγμα:

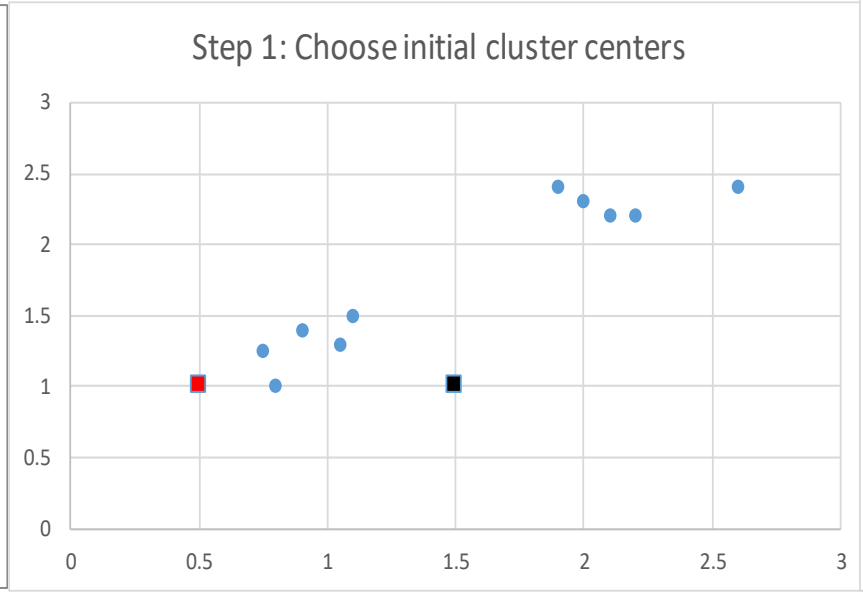
Observ.	Feature 1	Feature 2	Feature 3	Feature 4	**Distance to center**
1	1.00	1.00	0.40	0.25	**0.145**
2	0.80	1.20	0.25	0.40	**0.258**
3	0.82	1.05	0.35	0.50	**0.206**
4	1.10	0.80	0.21	0.23	**0.303**
5	0.85	0.90	0.37	0.27	**0.137**
Center	**0.914**	**0.990**	**0.316**	**0.330**	

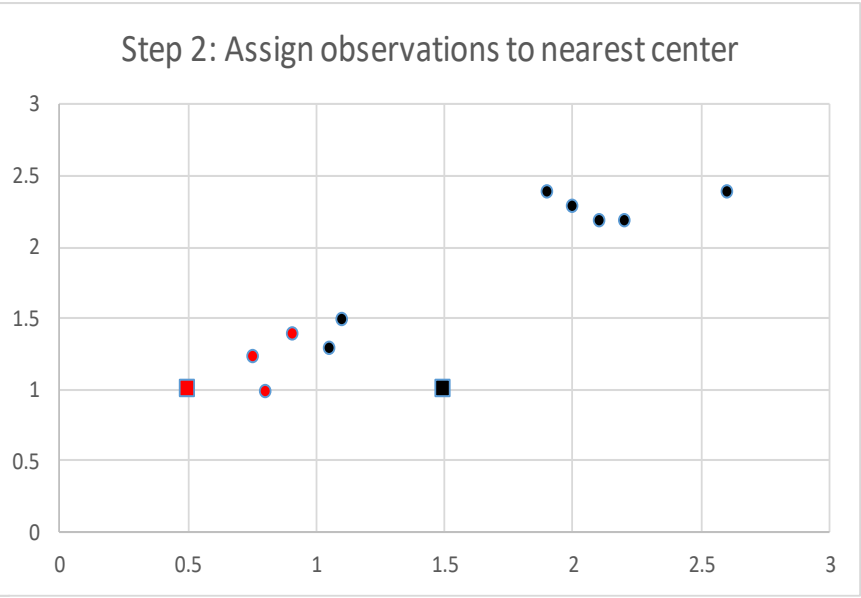
Ο αλγόριθμος k-means algorithm για να βρείτε k clusters

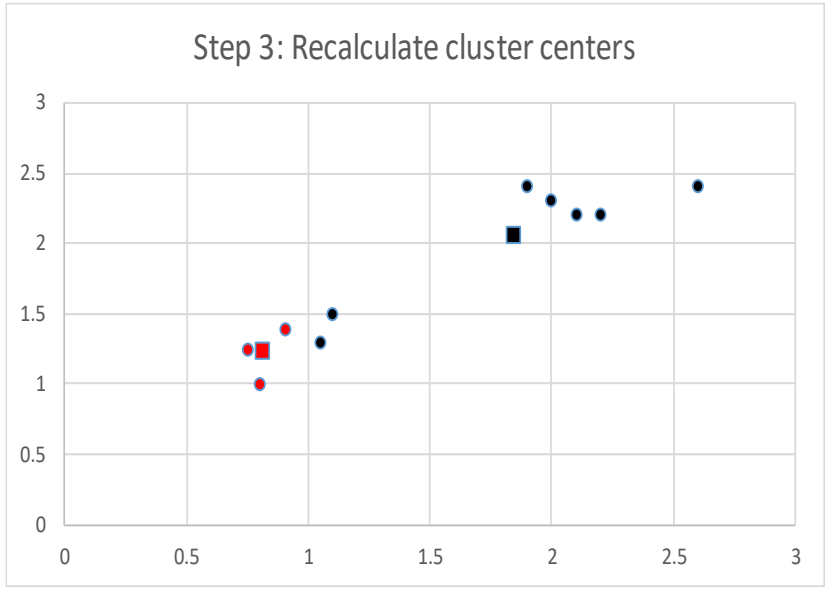


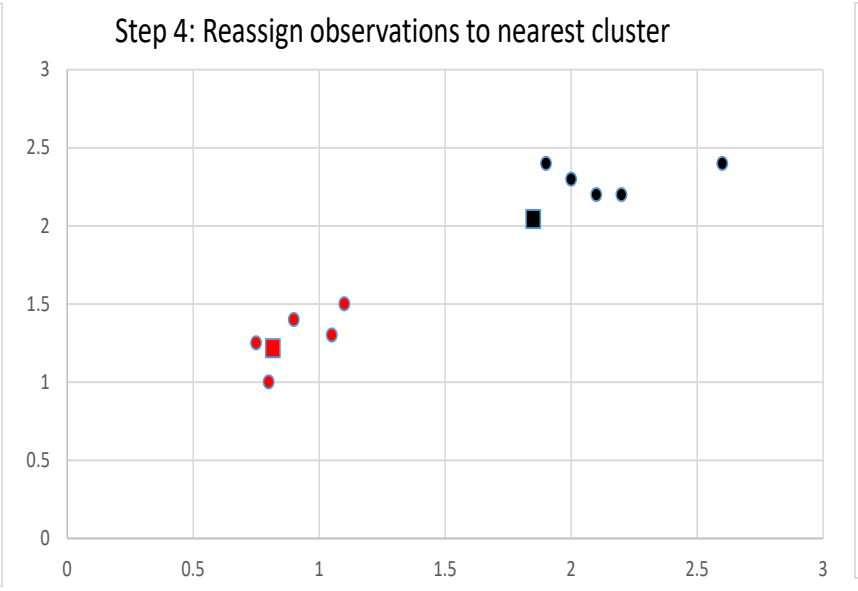
Ένα απλό παράδειγμα

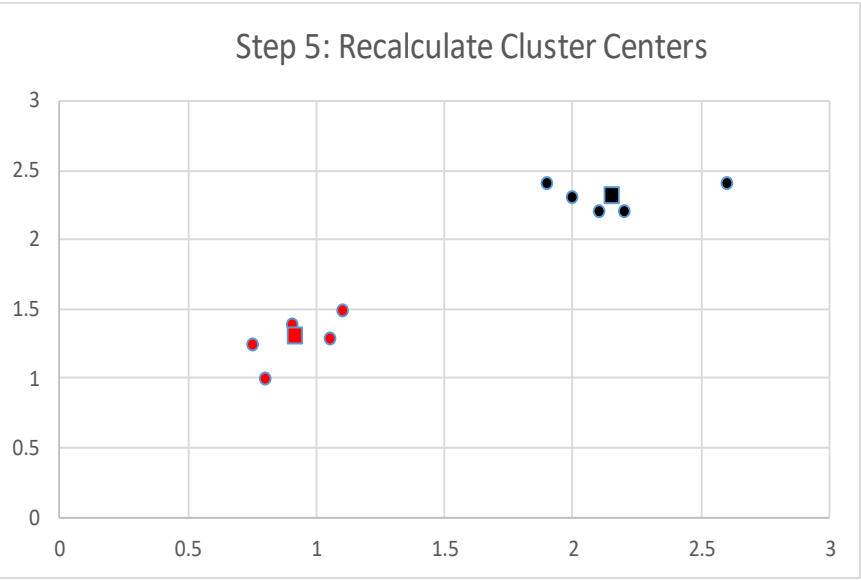












Inertia (Αδράνεια)

- Για κάθε δεδομένο k, ο στόχος είναι να ελαχιστοποιηθεί η αδράνεια, η οποία ορίζεται ως το άθροισμα των τετραγώνων εντός της ομαδας (cluster):

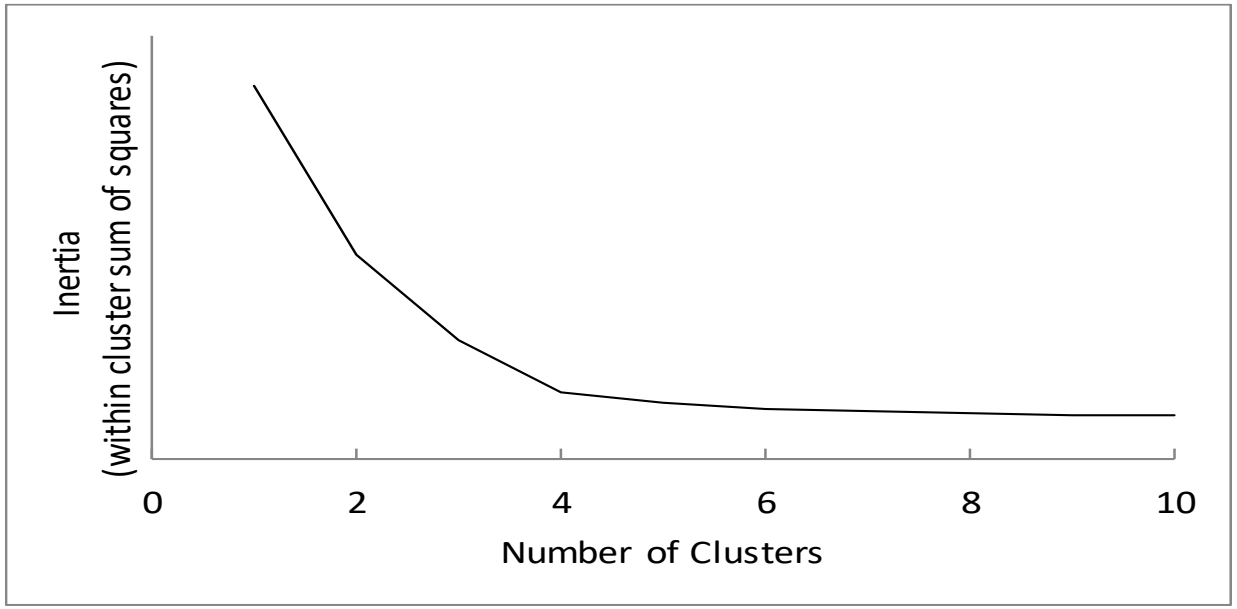
$$\text{Inertia} = \sum_{i=1}^{n} d_i^2$$

όπου d_i είναι η απόσταση παρατήρησης i από το cluster center του

- Στην πράξη χρησιμοποιούμε τον αλγόριθμο k-means με πολλά διαφορετικά σημεία εκκίνησης και επιλέγουμε το αποτέλεσμα που έχει τη μικρότερη αδράνεια

Επιλέγοντας το k

- **Η προσέγγιση elbow:**



Σε αυτό το παράδειγμα προτείνεται k=4

- **Η μέθοδος silhouette:**

  Για κάθε παρατήρηση i υπολογίζω το $a(i)$, τη μέση απόσταση από άλλες παρατηρήσεις στο cluster, και το $b(i)$, τη μέση απόσταση από τις παρατηρήσεις στο πλησιέστερο άλλο cluster. Το silhouette score για την παρατήρηση i, $s(i)$, ορίζεται ως

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]}$$

  Επιλέξτε τον αριθμό των clusters που μεγιστοποιεί το μέσο silhouette score σε όλες τις παρατηρήσεις.

Curse of Dimensionality

- Το μέτρο της Ευκλείδειας απόστασης αυξάνεται όσο αυξάνεται ο αριθμός των χαρακτηριστικών.
- Αυτό αναφέρεται ως curse of dimensionality
- Εξετάστε δύο παρατηρήσεις που έχουν τιμές για το χαρακτηριστικό j ίσες με x_j and y_j. Ένα εναλλακτικό μέτρο απόστασης που βρίσκεται πάντα μεταξύ 0 και 2 είναι

$$1 - \frac{\sum_{j=1}^{m} x_j y_j}{\sqrt{\sum_{j=1}^{m} x_j^2 \sum_{j=1}^{m} y_j^2}}$$

Εφαρμογή: Κίνδυνος χώρας (Country Risk)

Στόχος είναι η ομαδοποίηση (clustering) των χωρών ανάλογα με την επικινδυνότητά τους για ξένες επενδύσεις χρησιμοποιώντας δεδομένα του 2019. Ανοίξτε τα αρχεία

Country_risk_2019_data.csv, 4.1 K-means_elbow.ipynb , 4.2 K-means_silhouette.ipynb

Μέτρα κινδύνου χώρας

- Πραγματικός ρυθμός ανάπτυξης του ΑΕΠ (ΔΝΤ) - **GDP real growth rate** (IMF)
- Δείκτης διαφθοράς (Διεθνής Διαφάνεια)- **Corruption index** (Transparency international)
- Δείκτης Ειρήνης (Ινστιτούτο Οικονομικών και Ειρήνης) - **Peace index** (Institute for Economics and Peace)
- Δείκτης νομικού κινδύνου (Σύλλογος δικαιωμάτων ιδιοκτησίας) - **Legal Risk Index** (Property Rights Association)

❖Συλλέχθηκαν δεδομένα για 121 χώρες. Χρησιμοποιήθηκε **κλίμακα Z-score.**

Μέρος των δεδομένων

Country	Corruption Index	Peace Index	Legal Risk Index	Real GDP growth rate (% per yr)
Albania	35	1.821	4.546	2.983
Algeria	35	2.219	4.435	2.553
Argentina	45	1.989	5.087	−3.061
Armenia	42	2.294	4.812	6.000
Australia	77	1.419	8.363	1.713
Austria	77	1.291	8.089	1.605

In [5]:
```
# load raw data
DATA_FOLDER = './'
raw = pd.read_csv(os.path.join(DATA_FOLDER, 'Country_risk_2019_data.csv'))

# check the raw data
print("Size of the dataset (row, col): ", raw.shape)
print("\nFirst 5 rows\n", raw.head(n=5))
print("\nFirst 5 rows and 5 columns\n",raw .iloc[:5 , :5])
```

```
Size of the dataset (row, col):  (121, 6)

First 5 rows
      Country Abbrev  Corruption  Peace  Legal  GDP Growth
0     Albania     AL          35  1.821  4.546       2.983
1     Algeria     DZ          35  2.219  4.435       2.553
2   Argentina     AR          45  1.989  5.087      -3.061
3     Armenia     AM          42  2.294  4.812       6.000
4   Australia     AU          77  1.419  8.363       1.713

First 5 rows and 5 columns
      Country Abbrev  Corruption  Peace  Legal
0     Albania     AL          35  1.821  4.546
1     Algeria     DZ          35  2.219  4.435
2   Argentina     AR          45  1.989  5.087
3     Armenia     AM          42  2.294  4.812
4   Australia     AU          77  1.419  8.363
```

Συνοπτικά στατιστικά στοιχεία (Summary statistics) και Πίνακας συσχετίσεων (correlation matrix)

```
In [6]: # print summary statistics
        print("\nSummary statistics\n", raw.describe())
        print("\nCorrelation matrix\n", raw.corr())
```

```
Summary statistics
        Corruption       Peace       Legal  GDP Growth
count  121.000000  121.000000  121.000000  121.000000
mean    46.842975    2.001017    5.752529    2.657529
std     18.702499    0.461485    1.373932    2.563741
min     15.000000    1.072000    2.671000   -9.459000
25%     33.000000    1.699000    4.785000    1.249000
50%     41.000000    1.939000    5.455000    2.600000
75%     60.000000    2.294000    6.488000    4.000000
max     87.000000    3.369000    8.712000    7.800000

Correlation matrix
             Corruption     Peace     Legal  GDP Growth
Corruption    1.000000 -0.705002  0.938512   -0.123545
Peace        -0.705002  1.000000 -0.662233   -0.004428
Legal         0.938512 -0.662233  1.000000   -0.150369
GDP Growth   -0.123545 -0.004428 -0.150369    1.000000
```

- Ο Δείκτης διαφθοράς (**Corruption index) και ο Δείκτης νομικού κινδύνου (Legal Risk) είναι υψηλά συσχετιζόμενοι (correlation = 0.939).**
- **Για τον λόγο αυτόν στον αλγόριθμο θα χρησιμοποιήσουμε τους παρακάτω δείκτες:** GDP growth rate, Peace index, Legal risk index
- Ας δούμε πρώτα τα δεδομένα μετά το scaling στην επόμενη διαφάνεια προτού προχωρήσουμε στον αλγόριθμο.

Δεδομένα μετά την κλίμακα Z-score

Country	Corruption Index	Peace Index	Legal Risk Index	Real GDP growth rate (% per yr)
Albania	−0.633	−0.390	−0.878	0.127
Algeria	−0.633	0.472	−0.959	−0.041
Argentina	−0.099	−0.026	−0.484	−2.231
Armenia	−0.259	0.635	−0.685	1.304
Australia	1.612	−1.261	1.900	−0.368

```
In [9]: #K means cluster
        #Pick features & normalization
        #Since Corruption and Legal are highly correlated, we drop the Courrup
        X = raw[['Peace', 'Legal', 'GDP Growth']]
        X = (X - X.mean()) / X.std()
        print(X.head(5))
```

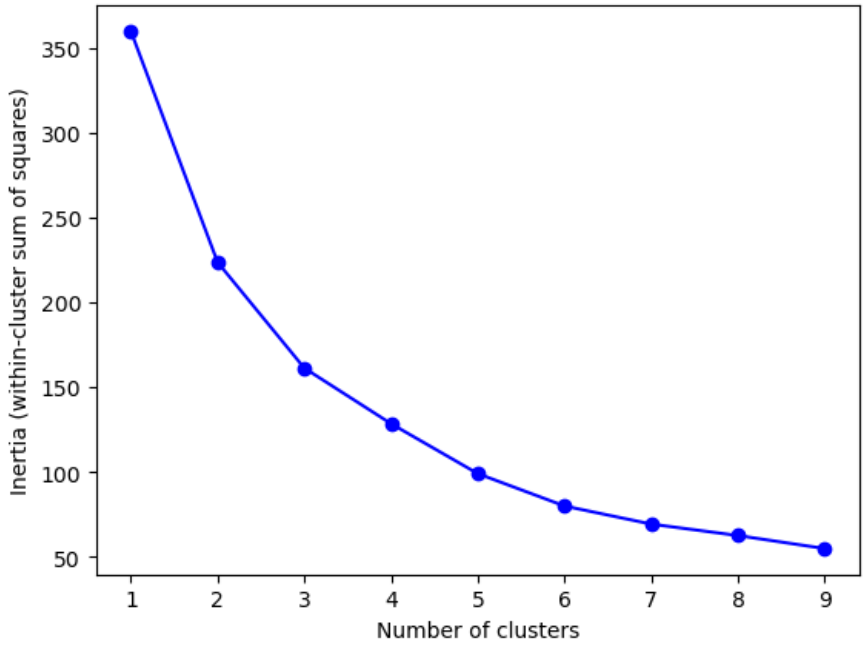
```
      Peace     Legal  GDP Growth
0 -0.390081 -0.878158    0.126952
1  0.472352 -0.958948   -0.040772
2 -0.026039 -0.484397   -2.230541
3  0.634871 -0.684553    1.303747
4 -1.261182  1.900001   -0.368418
```

I. Προσέγγιση elbow

Πώς το συνολικό άθροισμα τετραγώνων εντός του cluster μειώνεται καθώς το k αυξάνεται όταν χρησιμοποιείται ο αλγόριθμος k-means

In [10]:
```
#Perform elbow method
# https://stackoverflow.com/questions/41540751/sklearn-kmeans-equivalent-of-elbow-method
Ks = range(1, 10)
inertia = [KMeans(i).fit(X).inertia_ for i in Ks]

fig = plt.figure()
plt.plot(Ks, inertia, '-bo')
plt.xlabel('Number of clusters')
plt.ylabel('Inertia (within-cluster sum of squares)')
plt.show()
```

Inertia
(Αδράνεια)



Με βάση το διάγραμμα προτείνεται k=3, άρα 3 clusters.

Cluster centers

	Peace index	Legal index	GDP
High risk	1.23	−0.83	−1.08
Moderate risk	-0.85	1.02	-0.24
Low risk	0.23	-0.54	0.65

Σημειώστε ότι οι υψηλές τιμές για τον peace index είναι κακές ενώ οι υψηλές τιμές για τον legal risk index είναι καλές.

In [11]:
```
#K means with k=3
k = 3
kmeans = KMeans(n_clusters=k, random_state=1)
kmeans.fit(X)

# print inertia & cluster center
print("inertia for k=3 is", kmeans.inertia_)
print("cluster centers: ", kmeans.cluster_centers_)

# take a quick look at the result
y = kmeans.labels_
print("cluster labels: ", y)
```

```
inertia for k=3 is 161.1333871005255
cluster centers:  [[ 1.22506036 -0.83385901 -1.07842464]
 [-0.85097477  1.02149992 -0.23897931]
 [ 0.23006626 -0.54045468  0.65506397]]
cluster labels:  [2 2 0 2 1 1 2 2 2 1 2 2 2 1 0 2 0 2 1 0 1 2 2 1 2 1 1 0 1 2 0 2 2 1 2 1 1
 2 2 1 2 2 2 2 1 1 2 2 0 1 2 1 1 1 1 2 2 1 1 1 0 0 1 2 2 1 2 2 1 0 2 2 2 2
 2 1 1 0 0 1 1 0 2 0 2 2 1 1 1 1 0 2 0 2 2 2 1 1 1 0 1 2 1 1 1 2 2 2 0 2 0
 2 0 1 1 1 1 2 0 2 0]
```

II. Μέθοδος silhouette

Number of clusters	Average silhouette score
2	0.351
3	0.356
4	0.337
5	0.344
6	0.348
7	**0.360**
8	0.315
9	0.332

```
In [19]: # Silhouette Analysis
range_n_clusters=[2,3,4,5,6,7,8,9,10]
for n_clusters in range_n_clusters:
    clusterer=KMeans(n_clusters=n_clusters, random_state=1)
    cluster_labels=clusterer.fit_predict(X)
    silhouette_avg=silhouette_score(X,cluster_labels)
    print("For n_clusters=", n_clusters,
          "The average silhouette_score is :", silhouette_avg)
```

```
For n_clusters= 2 The average silhouette_score is : 0.3509139523852161
For n_clusters= 3 The average silhouette_score is : 0.3558522334350506
For n_clusters= 4 The average silhouette_score is : 0.3372449209416129
For n_clusters= 5 The average silhouette_score is : 0.34438420977393375
For n_clusters= 6 The average silhouette_score is : 0.34875382122984605
For n_clusters= 7 The average silhouette_score is : 0.3603542108728006
For n_clusters= 8 The average silhouette_score is : 0.3394917368960437
For n_clusters= 9 The average silhouette_score is : 0.3152647236003266
For n_clusters= 10 The average silhouette_score is : 0.3090796538425007
```

Επιλέγουμε 7 clusters επειδή εκεί μεγιστοποιείται το μέσο silhouette score (Average silhouette score).

Cluster centers

	Peace index	Legal index	GDP
(High)1	0.21	0.53	-0.59
2	0.25	-0.53	1.35
3	-1.26	1.79	-0.53
4	1.85	-1.05	-0.15
5	0.71	-0.96	-3.44
6	0.05	-0.69	-0.02
(Low)7	-0.85	0.53	0.20

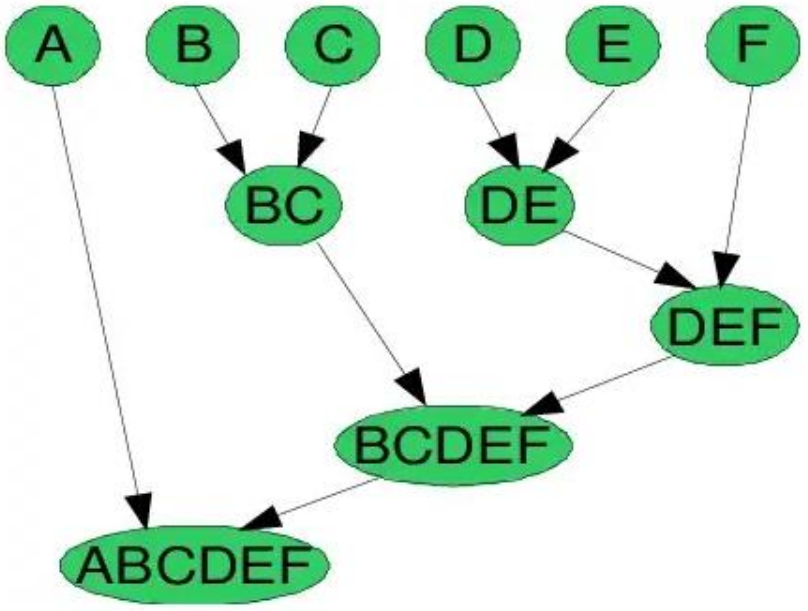
```
]: #K means with k=7
   k = 7
   kmeans = KMeans(n_clusters=k, random_state=1)
   kmeans.fit(X)

   # print cluster center
   print("cluster centers: ", kmeans.cluster_centers_)

   # take a quick look at the result
   y = kmeans.labels_
   print("cluster labels: ", y)
```

```
cluster centers:  [[ 0.20555045  0.52556914 -0.59243165]
 [ 0.25388748 -0.52823312  1.34995125]
 [-1.26104619  1.7919167  -0.52568349]
 [ 1.84989234 -1.05242316 -0.14505601]
 [ 0.70529573 -0.95894794 -3.43893096]
 [ 0.04538919 -0.6852538  -0.01515842]
 [-0.84739893  0.52786017  0.19939546]]
cluster labels:  [5 5 4 1 2 2 5 0 1 2 1 5 5 6 0 6 3 3 2 3 6 1 3 6 5 6 6 3 2 1 5 1 5 6 1 2 0
 5 1 2 1 5 5 5 6 2 1 1 4 6 0 0 0 2 0 5 1 0 0 6 3 5 6 1 5 6 3 1 6 3 5 5 5 5
 1 2 2 4 3 2 0 3 6 5 5 1 6 6 6 6 3 1 0 1 5 1 2 6 6 0 6 5 2 2 6 1 5 5 0 5 3
 1 3 0 2 0 0 1 3 5 4]
```

Hierarchical Clustering

Hierarchical Clustering (Agglomerative)

- Ξεκινήστε με κάθε παρατήρηση που θεωρείται ως ένα ατομικό cluster.
- Συνδυάστε τα δύο πιο κοντινά clusters.
- Συνεχίστε έως ότου όλες οι παρατηρήσεις έχουν συνδυαστεί σε ένα ενιαίο cluster.



Agglomerative Hierarchical Clustering Technique

The basic algorithm of Agglomerative is the following

- Compute the *connectivity matrix* .
- Let each data point be a cluster.
- Repeat: Merge the two closest clusters and update the proximity matrix.
- Until only a single cluster remains

- Για τα μέτρα εγγύτητας των clusters μπορείτε να συμβουλευτείτε το AgglomerativeClustering documentation της Python

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

linkage : *{'ward', 'complete', 'average', 'single'}, default='ward'*

Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

- 'ward' minimizes the variance of the clusters being merged.
- 'average' uses the average of the distances of each observation of the two sets.
- 'complete' or 'maximum' linkage uses the maximum distances between all observations of the two sets.
- 'single' uses the minimum of the distances between all observations of the two sets.

Εφαρμογή: Κίνδύνος χώρας (όταν όλα τα χαρακτηριστικά χρησιμοποιούνται

- Εφαρμογή του Hierarchical Clustering με βάση τη **μέση απόσταση**

Ανοίξτε τα αρχεία
Country_risk_2019_data.csv, *4.3 hierarchical_clustering_averagemethod.ipynb*

- Εφαρμογή του Hierarchical Clustering με βάση τη **μέθοδο Ward.**

Ανοίξτε τα αρχεία
Country_risk_2019_data.csv, *4.4 hierarchical_clustering_wardmethod.ipynb*

Principal Components Analysis (PCA)

Principal Components Analysis (PCA)

- Αυτή είναι μια άλλη προσέγγιση για τη μείωση του αριθμού των μεταβλητών.
- Το PCA αντικαθιστά ένα σύνολο n μεταβλητών με n παράγοντες έτσι ώστε:
  - Οποιαδήποτε παρατήρηση στις αρχικές μεταβλητές είναι ένας γραμμικός συνδυασμός των n παραγόντων.
  - Οι n παράγοντες είναι ασύνδετοι.
  - Η ποσότητα ενός συγκεκριμένου παράγοντα σε μια συγκεκριμένη παρατήρηση είναι ο factor score.
  - Η σημασία ενός συγκεκριμένου παράγοντα μετριέται με την τυπική απόκλιση της βαθμολογίας του παράγοντα στις παρατηρήσεις.
- Η ιδέα είναι να βρούμε μερικές μεταβλητές που αντιπροσωπεύουν ένα υψηλό ποσοστό της διακύμανσης στις παρατηρήσεις.

Εφαρμογή: Κίνδύνος χώρας (όταν όλα τα χαρακτηριστικά χρησιμοποιούνται

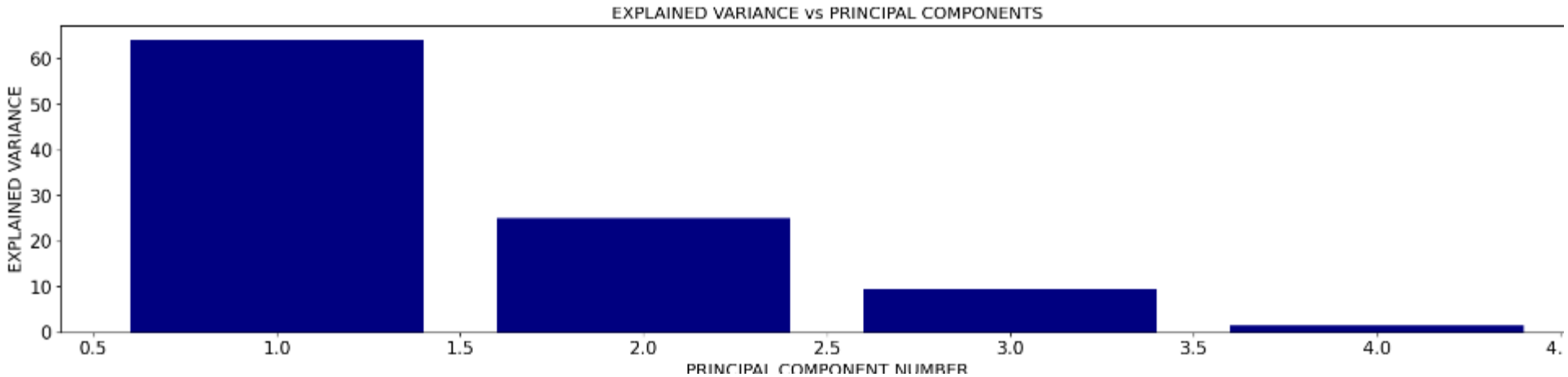
- Ανοίξτε το αρχεία *Country_risk_2019_data.csv , 4.5 PCA example.ipynb*
- Factor Loadings

	PC1	PC2	PC3	PC4
Corruption index	- 0.602	−0.015	0.328	0.728
Peace index	0.524	0.201	0.825	0.065
Legal risk index	-0.594	0.022	0.425	−0.683
GDP Growth rate	0.103	−0.979	0.174	−0.013

```
[9]: array([[-0.60188457,  0.52361257, -0.59403386,  0.10338576],
            [-0.01512074,  0.20084677,  0.02192676, -0.97926051],
            [ 0.32824929,  0.82539386,  0.42519586,  0.17374078],
            [ 0.72784526,  0.06492634, -0.68253313, -0.01320492]])
```

- Ο πρώτος PCA factor έχει περίπου τις ίδιες σταθμίσεις στους Corruption, Peace, Legal Risk indices.

```
In [10]: # Percentage of variation explained by successive eigenvectors/PCS
         model.explained_variance_ratio_.round(2)

Out[10]: array([0.64, 0.25, 0.09, 0.01])

Out[11]: <BarContainer object of 4 artists>
```



- Ο πρώτος PCA factor εξηγεί το 64% της διακύμανσης.
- Ο δεύτερος PCA factor εξηγεί το 25% της διακύμανσης.
- Ο τρίτος PCA factor εξηγεί το 9% της διακύμανσης.
- Ο τέταρτος PCA factor εξηγεί το 1% της διακύμανσης.