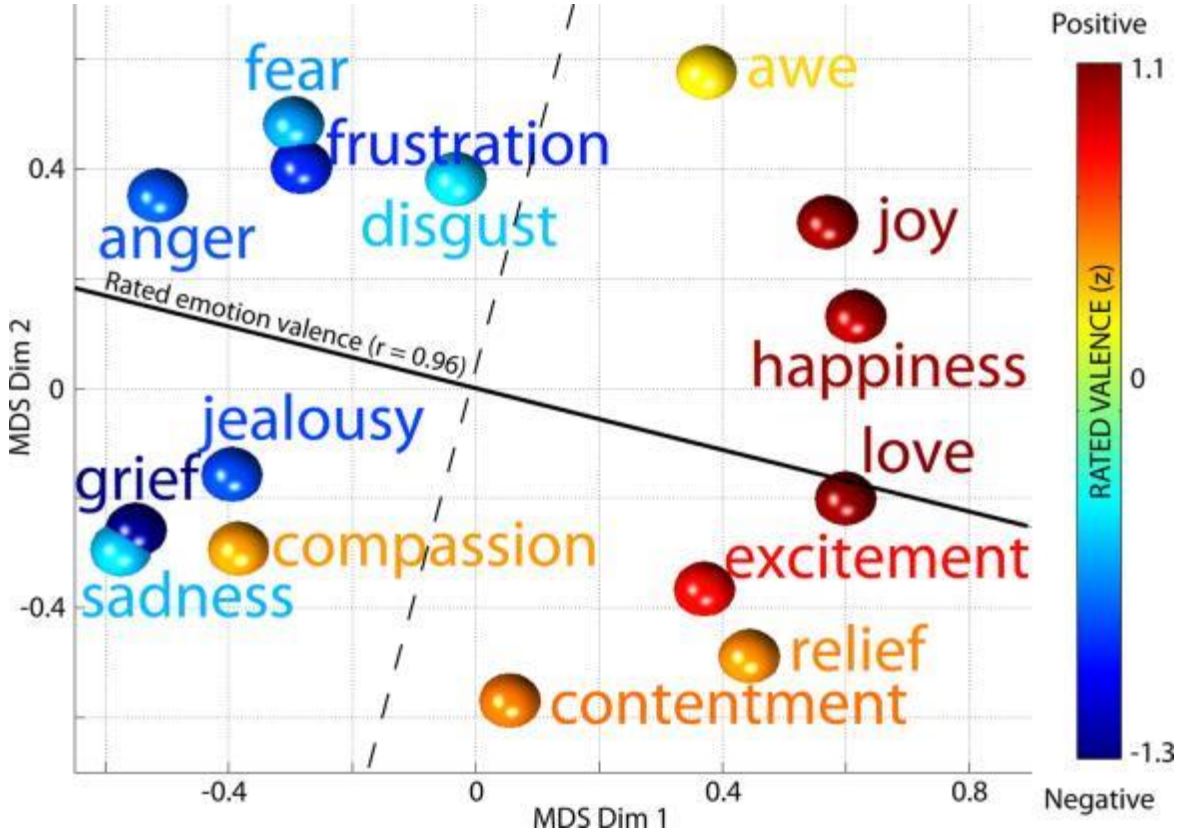
MDS II



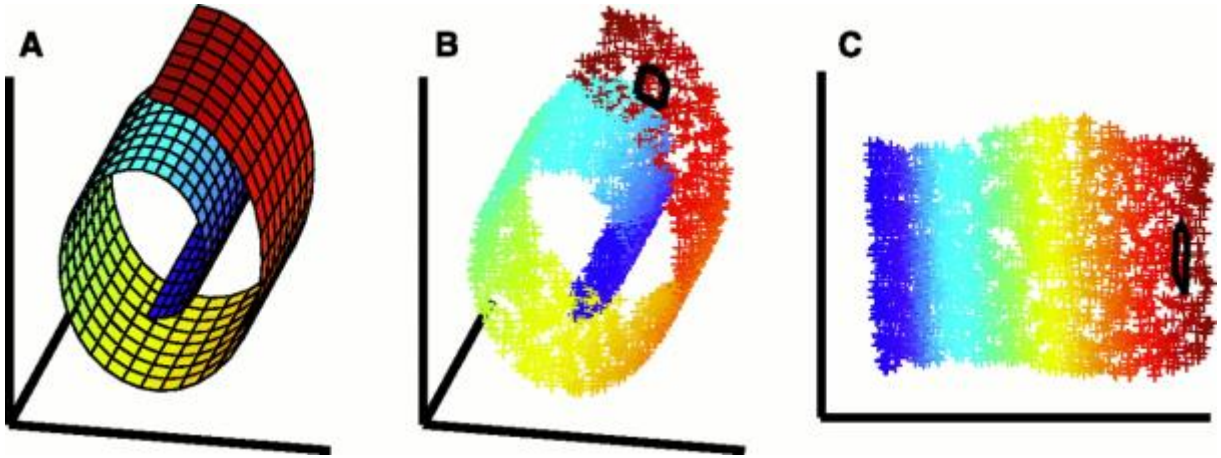
Ref:
http://journal.frontiersin.org/article/10.3389/neuro.09.061.2009/

Figure 8. Multi-dimensional scaling of median time weights of broadband IMs for each emotion. Similarities between median IM weights in the 15 emotion periods, drawn from log-spectral decompositions for each subject of brain source ICs only, as represented in the best-fitting two-dimensional space by non-metric multi-dimensional scaling (MDS). Colors of the balls represent the mean behavioral ratings of (positive or negative) valence of the 15 emotion terms by a separate subject cohort. The solid line shows the best-fit regression direction (r = 0.96) predicting mean rated valence for each emotion term from its location in the 2-D MDS space solely based on IM weights after neglecting compassion (see text). The dashed line orthogonal to this cleanly separates positive-valence emotions terms (warm color balls) from negative-valence terms (cool color balls).

Menggian LU

What is MDS? – An Overview

- Find a mapping such that nearby points on the roll (B) are also adjacent in 2D (c) Keeping **"distances"** between points similar



- The "multi" part means the maps not necessarily to be 1 or 2-D

What MDS does?

- **Problem:** high-dimensional data often lies on an intrinsically low-dimensional manifold
- **Goal:** given high-dimensional data, find a low-dimensional coordinates of the data describing where the points lie on the manifold
- **Usage:**
  - Data visualization
  - Pattern recognition
  - Smooth interpolation

How to do MDS?

- The input for MDS is proximities, often proximity matrix.
- MDS looks for low-dimensional spatial configuration to represent the given similarity of dissimilarity
- Thus first step: *deriving proximities from your raw data*.
  1. Direct:
     - A numerical (dis)similarity value to each pair
     - A ranking of the pairs w.r.t. their similarity
  2. Indirect:
     - From confusion data
     - From Correlation matrices

Measure the "distance"

Euclidean distance:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$

Use dist() in R

What if variables are on different scales?

- Normalization or Standardization

```
R> dist(scale(measure[, c("chest", "waist", "hips")],
+        center = FALSE))

      1    2    3    4    5    6    7    8    9   10   11
2  0.17
3  0.15 0.08
4  0.22 0.07 0.14
5  0.11 0.15 0.09 0.22
6  0.29 0.16 0.16 0.19 0.21
7  0.32 0.16 0.20 0.13 0.28 0.14
8  0.23 0.11 0.11 0.12 0.19 0.16 0.13
```

Measure the "distance" (Cont'd)

Euclidean distance: $d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2}$

Manhattan distance: $d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ip} - x_{jp}|$

Maximum distance: $d(i,j) = max_{k=1}^{p} |x_{ik} - x_{jk}|$

L2, L1, L-infinity

Less common used one:
Minkowski distance: (The p norm) Lp范数

$$d(i,j) = (|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + ... + |x_{ip} - x_{jp}|^q)^{\frac{1}{q}}$$

Canberra: $d(i,j) = \sum_{p=1}^{p=P} \frac{|x_{ip} - x_{jp}|}{|x_{ip} + x_{jp}|}$

Use `dist()` in R

To scale or not to scale…

Some scenarios:

- Variables measured with different units (lb, ft, sec, …);
  Scale! Because (a) variable with largest range will have most weight; (b) distance depends on scale, scaling gives every variable equal weight

$$d(i,j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_p(x_{ip} - x_{jp})^2}$$

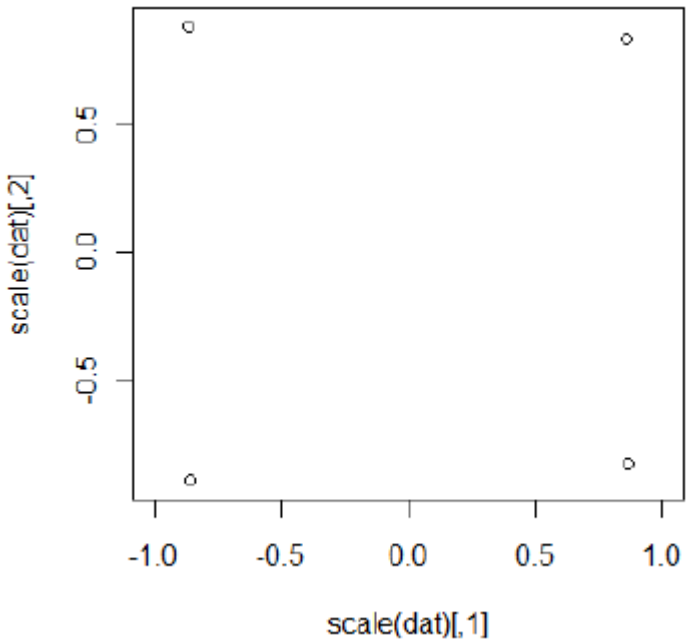
- Variables measured with the same units (lat, long…);
  Don't! Because information lost after scaling

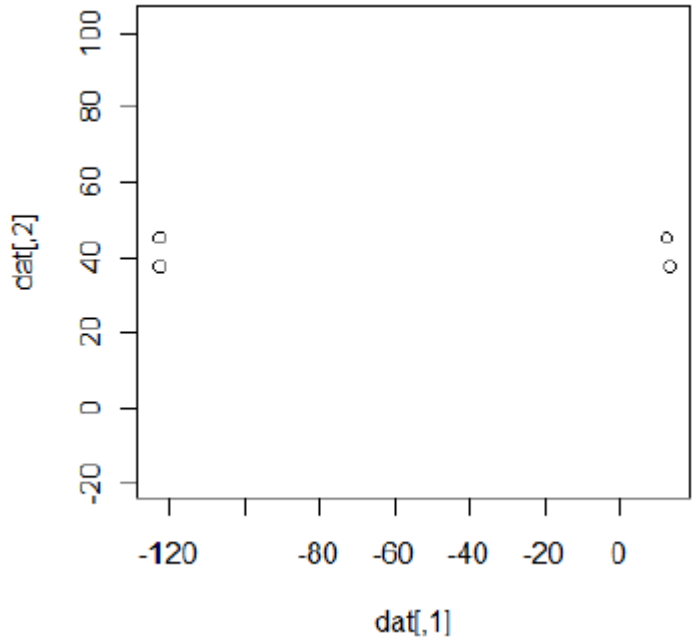
- Most often, you'd better scale

同量纲：不标准化，例如经度维度数据；
不同量纲：标准化。

Object	x1	x2
A	13.3	38.0
B	12.4	45.4
C	-122.7	45.6
D	-122.4	37.7

Object	Long.	Lat.
Palermo	13.3	38.0
Venice	12.4	45.4
Portland	-122.7	45.6
San Francisco	-122.4	37.7





Measure the "distance" (Cont'd)

Between variables?

Correlation measurement of "distance" or association:

调整dissimilarity定义是的满足距离的四大要求

$$\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

Dissimilarity(X,Y) = constant* (1-corr(X,Y))

- Often data context determines distance measurement, based on empirical or domain knowledge.

Similarity and Dissimilarity Matrix

If you recall PCA,

$$\underset{(n\times p)}{\mathbf{X}} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \xrightarrow{\text{PCA}} Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1q} \\ \cdots & \cdots & \cdots & y_{2q} \\ \cdots & \cdots & \cdots & \cdots \\ y_{n1} & y_{n1} & \cdots & y_{nq} \end{bmatrix}$$

Now we only have proximity matrix represents the similarity/dissimilarity between rows or columns of **X**, then how can you find **X** or **Y**? – MDS

Most R package works with either similarity or dissimilarity, be careful! – Convert to the right matrix!

Some examples: d_{ij} = constant – s_{ij} ; $d_{ij} = 1/s_{ij}$ – constant;

$d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$ **Another one in HW3**

Classical MDS

- Input: Distance matrix

  This is arc length? On a Sphere, we have a^2+b^2≥c^2, and it's distance: a+b≥c

- Example: given Euclidean distances among selected European cities

eurodist gives the road distances

$\{\delta_{ij}\}$

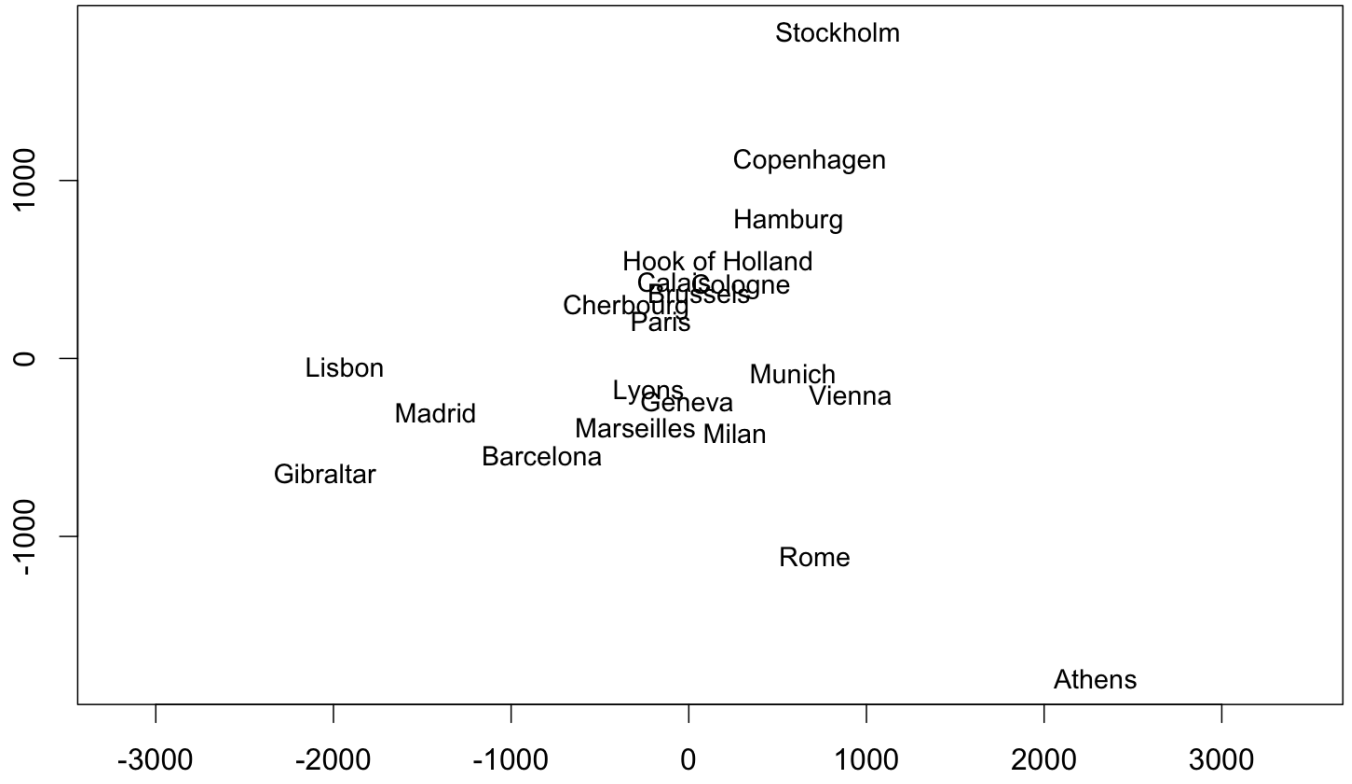
```
> eurodist
```

	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne	Copenhagen	Geneva	Gibraltar
Barcelona	3313								
Brussels	2963	1318							
Calais	3175	1326	204						
Cherbourg	3339	1294	583	460					
Cologne	2762	1498	206	409	785				
Copenhagen	3276	2218	966	1136	1545	760			
Geneva	2610	803	677	747	853	1662	1418		
Gibraltar	4485	1172	2256	2224	2047	2436	3196	1975	
Hamburg	2977	2018	597	714	1115	460	460	1118	2897
Hook of Holland	3030	1490	172	330	731	269	269	895	2428
Lisbon	4532	1305	2084	2052	1827	2290	2971	1936	676
Lyons	2753	645	690	739	789	714	1458	158	1817
Madrid	3949	636	1558	1550	1347	1764	2498	1439	698
Marseilles	2865	521	1011	1059	1101	1035	1778	425	1693
Milan	2282	1014	925	1077	1209	911	1537	328	2185
Munich	2179	1365	747	977	1160	583	1104	591	2565
Paris	3000	1033	285	280	340	465	1176	513	1971
Rome	817	1460	1511	1662	1794	1497	2050	995	2631
Stockholm	3927	2868	1616	1786	2196	1403	650	2068	3886
Vienna	1991	1802	1175	1381	1588	937	1455	1019	2974

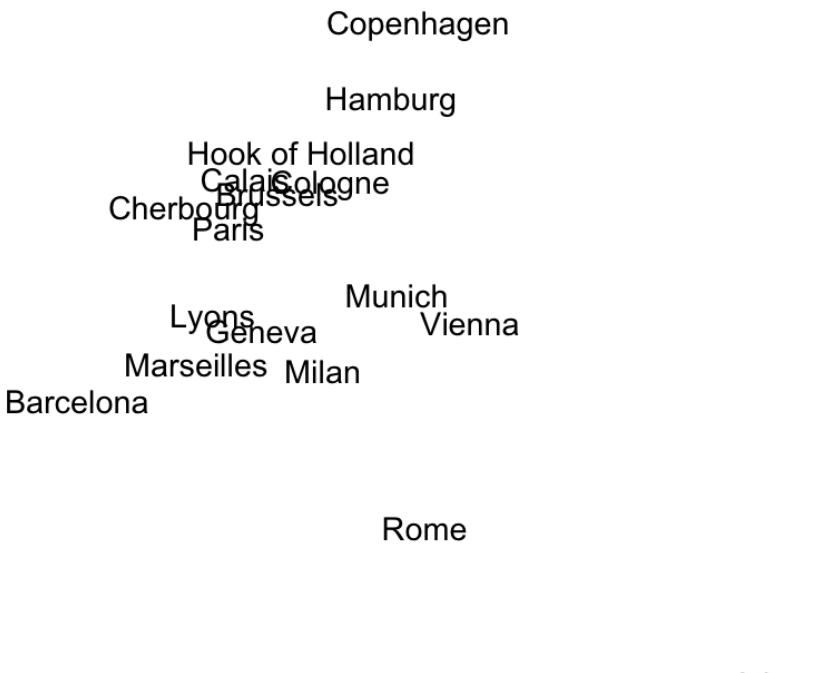
R: cmdscale()

$$\{d_{ij}\} \approx (rescaled)\{\delta_{ij}\}$$

On a Sphere, we have a^2+b^2≥c^2



Invariant under reflection, translation and rotation



Interpretation MDS

- The configuration can be reflected without changing the inter-point distances
- The inter-point distances are not affected if we change the origin
- The set of points (or axes) can be rotated without affecting the inter-point distances

❖ The interpretation of MDS must be invariant under reflection, translation and rotation.

Example: Country Dissimilarities

Kruskal and Wish (1978)

Data from a political science survey: values are average pairwise dissimilarities of countries from a questionnaire given to political science students.

	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

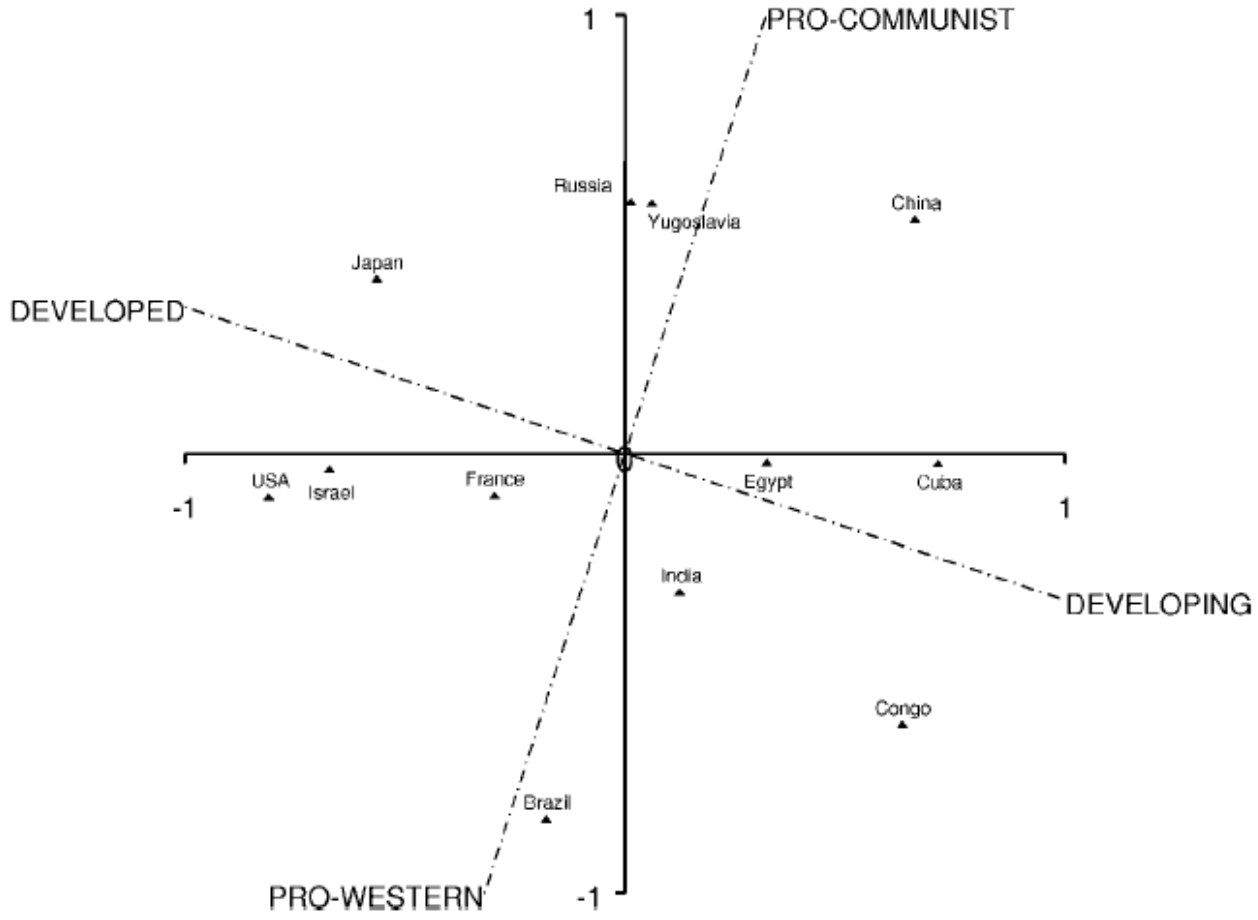
Same Scaling in 1 to 9

Ordinal MDS 做秩相关

- Dissimilarity rating: 1 (very different) to 9 (very similar)

Ordinal MDS

Ordinal MDS – Countries' Dissimilarities



Cautionary remarks:

1. Averaged similarities under implicit assumption of no difference among students' subject rating.
   → Individual Scaling or Three-Way Scaling (*Borg and Groenen, 2005* or *Kruskal and Wish, 1978*)

2. Finding the best interpretable axes (rotation) is not the end.
   → Clustering analysis (further)

Some Theory: Classical MDS

- Objective: $\{d_{ij}\} \approx (rescaled)\{\delta_{ij}\}$
- Input: Euclidean distances between n objects in p dimensions
- Output: Coordinates of points invariant to rotation, shift and reflection

- Two steps:
  1. Compute inner product matrix, B, from distance $D = \{d_{ij}\}$
  2. Compute positions from B

$$d_{ij}^2 = (x_i - x_j)^T (x_i - x_j) = x_i^T x_i + x_j^T x_j - 2x_i^T x_j$$

Let coordinates be $x_i \ (i = 1, \ldots, n)$, where $x_i = (x_{i1}, \ldots, x_{ip})^T$

$$d_{ij}^2 = \sum_{k=1}^{p} (x_{ik} - x_{jk})^2, \text{assuming } \bar{x} = 0$$

$$B = \mathbf{X}\mathbf{X}^T, \text{with } b_{ij} = \sum_{k=1}^{p} x_{ik} x_{jk} = x_i^T x_j$$

B: Inner product matrix

$$d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$$

Centering of coordinate matrix X:

$$\sum_{i=1}^{n} b_{ij} = 0$$

$$\sum_{i=1}^{n} b_{ij} = 0 \text{ and } \boxed{d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}}$$

$$\sum_i d_{ij}^2 = \sum_i (b_{ii} + b_{jj} - 2b_{ij}) \Leftrightarrow \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 = \frac{1}{n}\sum_{i=1}^{n} b_{ii} + b_{jj}$$

$$\sum_i d_{ij}^2 = \sum_i (b_{ii} + b_{jj} - 2b_{ij}) \Leftrightarrow \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 = \frac{1}{n}\sum_{j=1}^{n} b_{jj} + b_{ii}$$

$$\sum_{i,j} d_{ij}^2 = \sum_{i,j} (b_{ii} + b_{jj} - 2b_{ij}) \Leftrightarrow \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 = \frac{2}{n}\sum_{i=1}^{n} b_{ii}$$

$$b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i\bullet}^2 - d_{\bullet j}^2 + d_{\bullet\bullet}^2)$$

$B = \mathbf{X}\mathbf{X}^T$

B is a symmetric and positive definite n-by-n matrix

B can be diagonalized: $B = V \Lambda V^T$

$\Lambda = diag(\lambda_1, ..., \lambda_p)$, with $\lambda_1 \geq \lambda_2 ... \geq \lambda_p$ (eigenvalues)

Columns of V are eigenvectors

Some eigenvalues will be zero;

Drop them: $B = V_1 \Lambda_1 V_1^T$

$\mathbf{X} = V_1 \Lambda_1^{\frac{1}{2}}$

"Take square root" of matrix B: $\mathbf{X} = V_1 \Lambda_1^{\frac{1}{2}}$

X contains the point configuration in R^p

Does this remind you of PCA?

MDS vs. PCA

	MDS	PCA
Data	N x N Distance Matrix	P x P Covariance Matrix
Approach	Spectral decomposition	
Eigenvector/Eigenvalue	Same nonzero eigenvalues with a constant factor	
Results	The same	
Interpretations	▪ Dimension reduction without coordinates ▪ Preserve distances between data points ▪ Normally target 2 or 3d representation	▪ Dimension reduction with coordinates ▪ Preserve covariance of data ▪ Feature extractions, reconstruction
Computation	▪ $O((n+d)D^2)$	▪ $O((D+d)n^2)$

MDS: Analysis and Visualization Using R

- Metric MDS in R

  `cmdscale()` most popular function

  `wcmdscale()` weighted classical MDS

  `pco()` in ecodist

  `pco()` in labdsv

  `pcoa()` in ape

  Package: smacof

Example with **smacof**

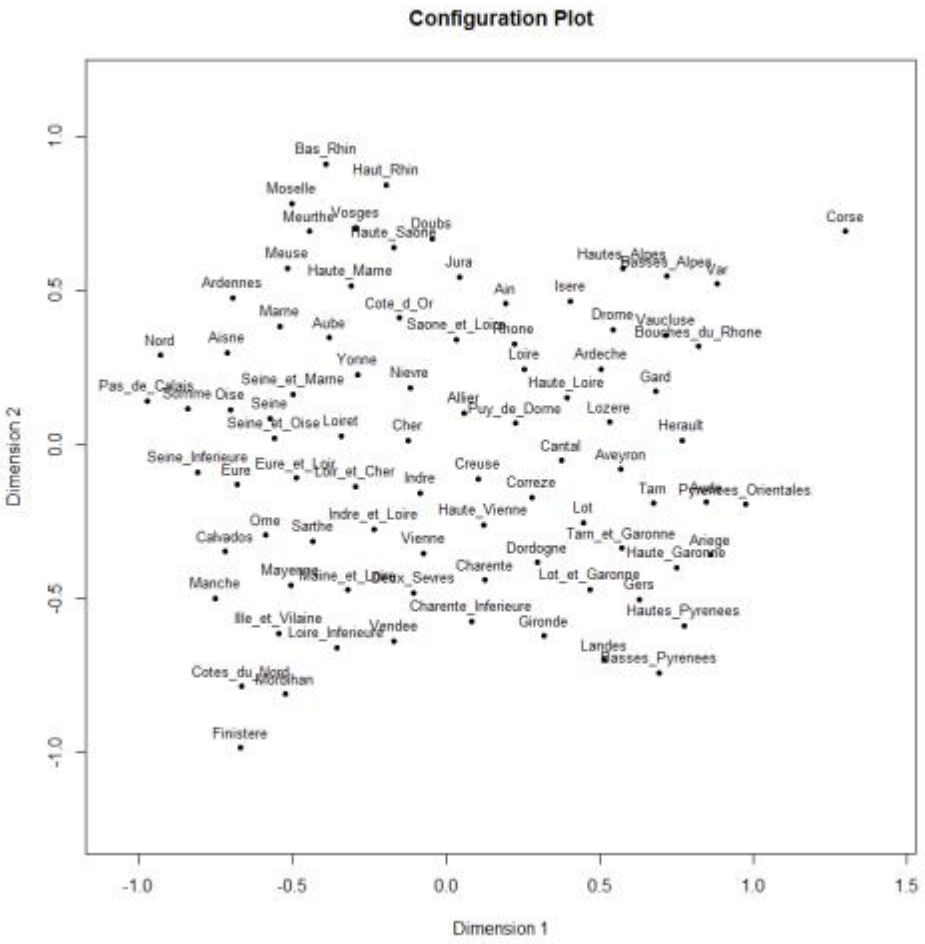
$$d_{ij}(X) \approx \delta_{ij}$$

$$d_{ij}(X) = \sqrt{\sum_{l=1}^{P} (x_{il} - x_{jl})^2}$$

- The elements of X are *configurations* of the objects.
- The *configurations* represent the coordinates in the configuration plot.

- Distances between French department centroids in 1830

- Distances between French department centroids in 1830



Packages used: `smacof` `RgoogleMaps`

cran.r-project.org

Classical MDS – Goodness of fit

- Goodness of fit (GOF) by reducing p-dimension to m-dimension

$$GOF = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \qquad \text{in cmdscale()} \quad GOF = \frac{\sum_{i=1}^{m} |\lambda_i|}{\sum_{i=1}^{p} |\lambda_i|} \quad \text{is used}$$

- Minimize the differences between $\{d_{ij}\}$ and $\{\delta_{ij}\}$

Notes: Classical MDS is good for Euclidean input data, quite fast in terms of computation