多次元尺度構成法

「MDSを使って使って使い倒す！
MDS入門から非対称MDS実習まで」

2010年3月27日～28日
日本行動計量学会 第13回春の合宿セミナー A2コース

中山 厚穂（長崎大学）
横山 暁（慶應義塾大学）

3月27日（土）多次元尺度構成法

- 午前　10:00～12:00　MDSとは何か
  - MDSの扱うデータとモデルの対応
    - Rでの実行可能なパッケージなどの説明
    - 計量MDSと非計量MDS(Rで実行するには…)
- 午後1　13:00～15:00　MDS「R」による分析方法の紹介
  - 午前の復習(非計量MDSとは)
  - 非計量MDSを実行する際に注意点などの説明
  - 非計量MDS「R」による分析の実行
- 午後2　15:30～17:30 MDS　事例による実習
  - 個人差MDSと多次元展開法についての説明
  - 「R」による分析方法の説明

3月28日（日）非対称多次元尺度構成法

- 午前　10:00～12:00　非対称MDSとは何か
  - 非対称MDSについての説明
    - 個人差モデルについても簡単に説明
    - 時間があれば，「R」による分析方法の紹介
- 午後　13:00～15:00　非対称MDS　事例による実習
  - 非対称MDSの「R」による分析方法の紹介
  - 2日間のQ&A

3 月27日（土）午前(10:00～12:00)

多次元尺度構成法(MDS)とは

多次元尺度構成法とは

- MDS(Multi Dimensional Scaling)
  - データに潜む対象間の関係を知りたい
  - 対象を「点」として「多次元空間」内に表現
  - 対象間の「類似度」の大小関係を点の間の「距離」で表現

  - 計量MDS
    - Torgersonの方法
      - 入力データ：比例尺度，間隔尺度

  - 非計量MDS
    - Shepard-Kruskalのアプローチ
      - 入力データ：順序尺度で可

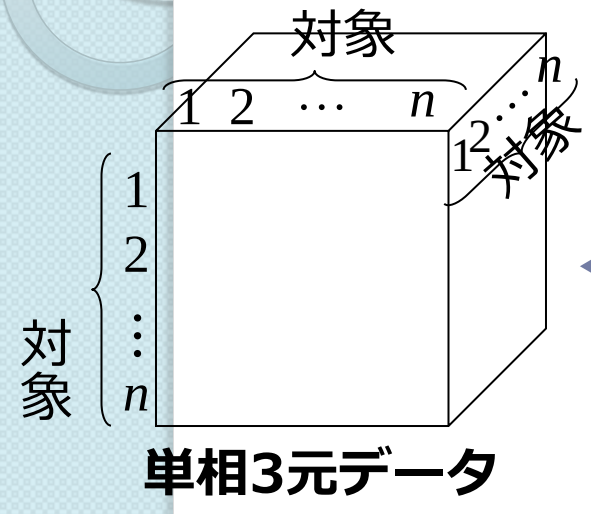
Rで実行するには…

- 計量MDS
  - R のパッケージstats
    - 計量MDSの関数cmdscale()
- 非計量MDS
  - RのパッケージMASS
    - 非計量MDSの関数isoMDS()とsammon()
- Rのパッケージsmacof (De Leeuw & Mair, 2009)
  - 計量と非計量多次元尺度構成法を実行可能
  - 個人差を考慮した分析を実行(INDSCAL)
  - 矩形行列のための分析(unfolding)
  - MULTIWAY PACKAGE(De Leeuw, on the horizon)
    - CANDECOMP and TUCKER

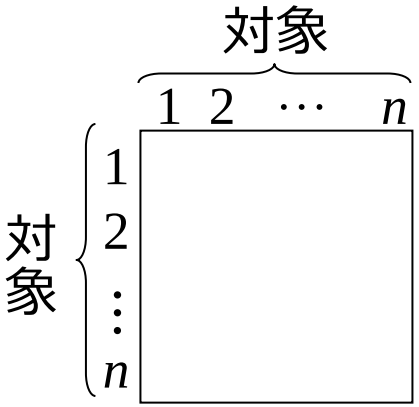
MDSにおいて用いられるデータ

- 相(mode)と元(way)という概念により整理
- *M*個の相と*N*個の元をもつデータ
  - *M*相N元データ (*M*≦*N*)
- データの相とは1組の対象を意味
  - 1つの相をもつデータを単相データ
  - 2つの相を持つデータを2相データ
  - *N*個の相を持つデータを*N*相データ
- 元の数は相がいくつ組み合わされているかにより決定

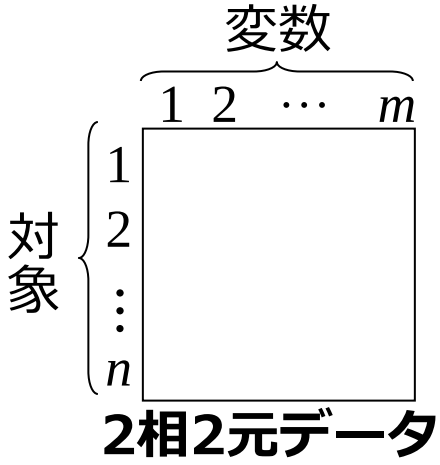
MDSにおいて用いられるデータ
相(mode)と元(way)という概念により整理



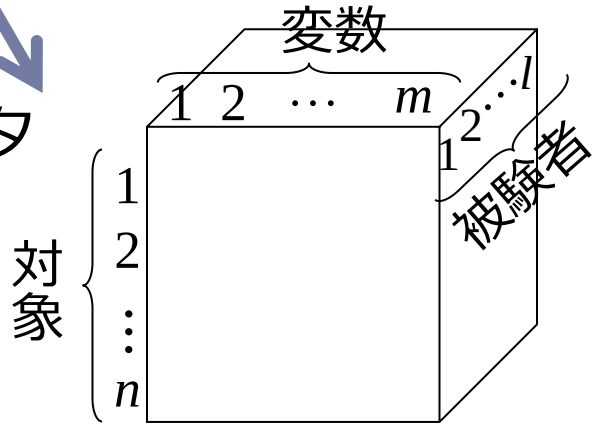
単相3元データ



単相2元データ



2相2元データ

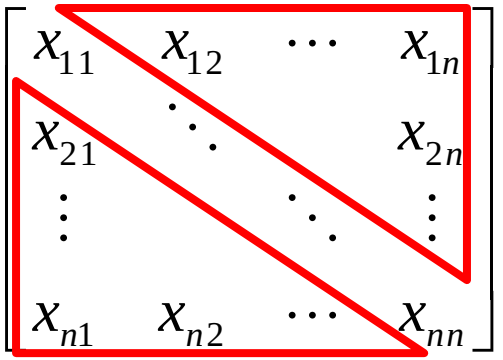


3相3元データ

- M個の相とN個の元をもつデータ
  - M相N元データ (M≦N)
- データの相
  - 1組の対象を意味
- 元の数
  - 相がいくつ組合されているかで決定

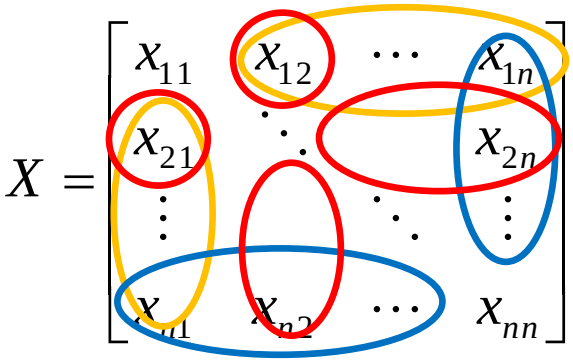
非対称データ

- 非対称データ
  - データの各行・列の総和が外的影響で異なる(データの上三角部分と下三角部分の値が異なる)
  - マーケティングでは分析のニーズが高い
    - ブランドスイッチングデータや複数商品の同時購買確率
- 非対称データの分析：多元・多相データへの拡張
  - 解析前に何らかの基準化(行列の再構成)を行う
    - Harshman,Green, Wind, and Lundy(1982)
  - 非対称モデルを活用
    - Okada and Imaizumi(1997)
    - De Rooij & Heiser(2003)



Harshman et al.(1982)の再構成法

- Harshman et al.(1982)



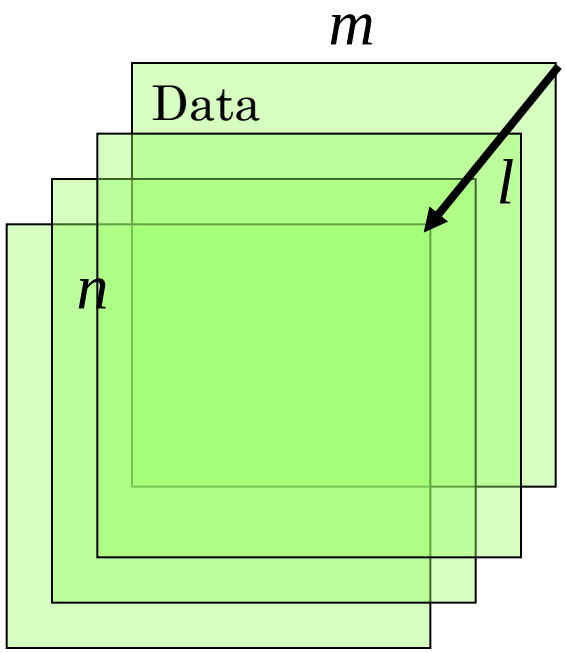
- という行列で，各iに対して，

$$\bar{X} = \frac{1}{2n}\sum_{\substack{j=1 \\ j\neq i}}^{n}(x_{ij} + x_{ji})$$

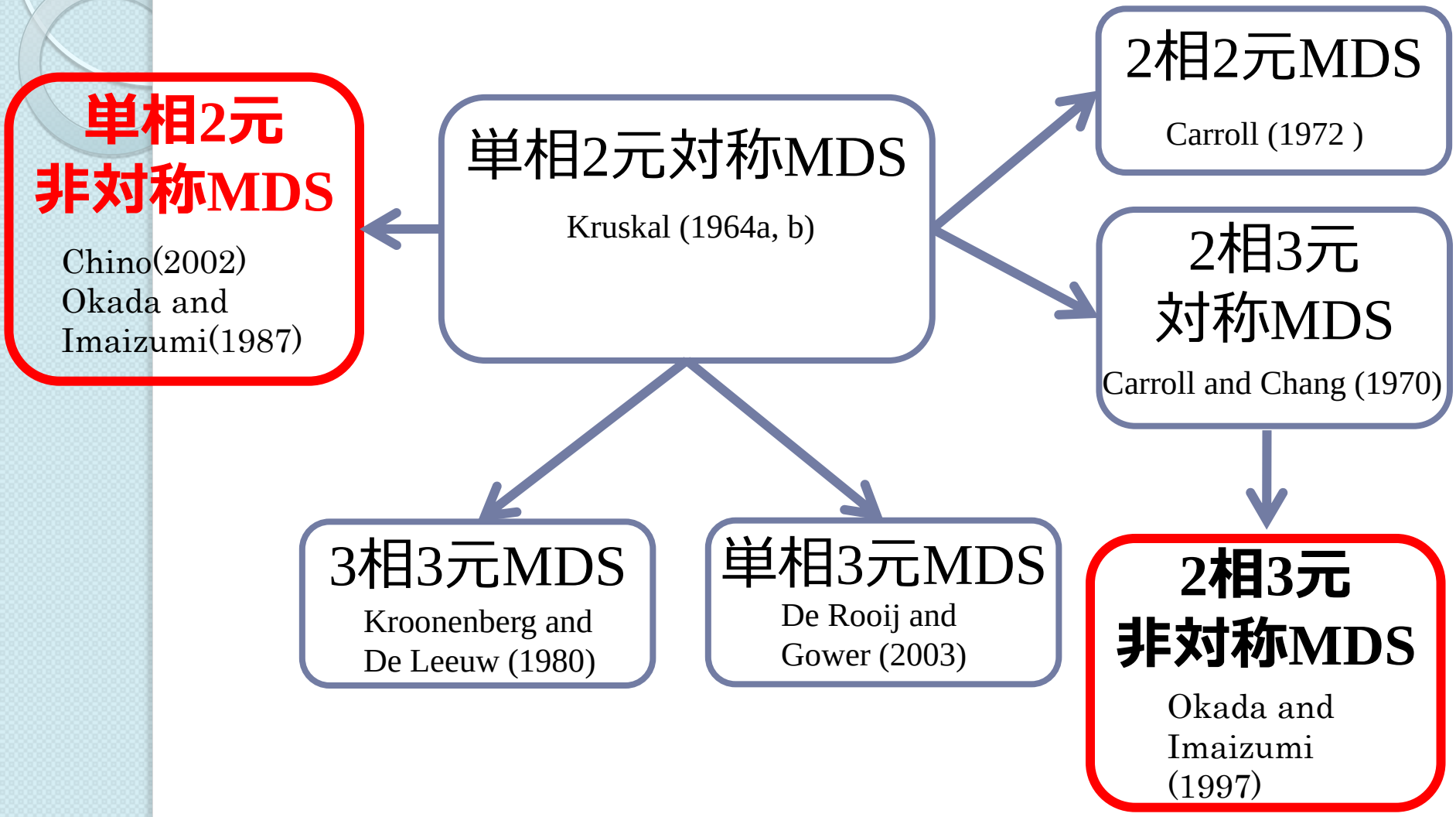
と対角要素を除く各行と列の和が一定となるように再構成

多次元尺度構成法(MDS)のモデル

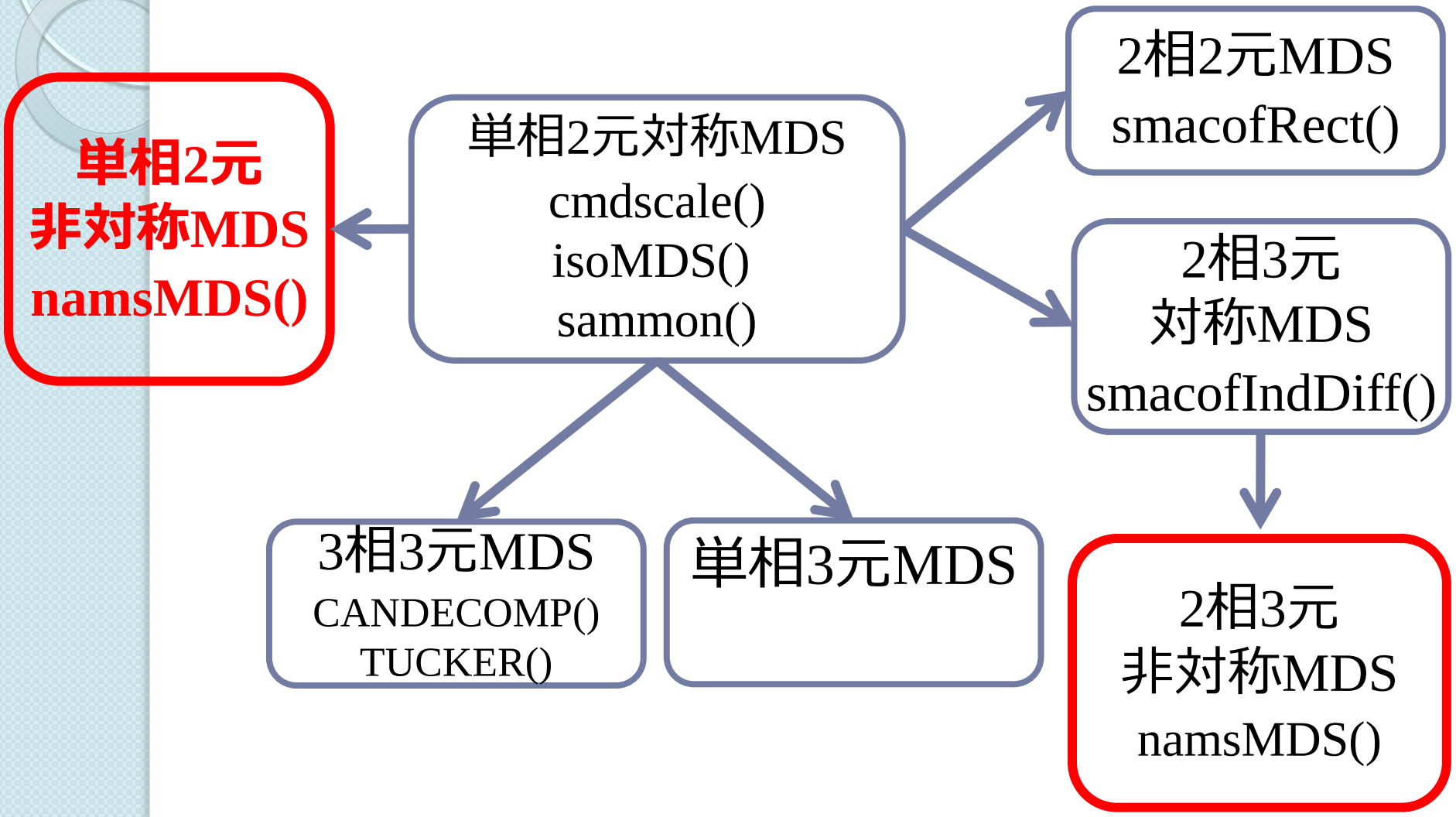
- 2元モデル
  - 単相2元モデル(対象×対象)
    - 対称・非対称
  - 2相2元モデル　(対象×個人)
- 3元モデル
  - 単相3元モデル(対象×対象×対象)
  - 2相3元モデル (対象×対象×個人)
    - 対称・非対称
  - 3相3元モデル (対象×個人×評価)



データとの対応関係


単相2元
非対称MDS
Chino(2002)
Okada and
Imaizumi(1987)
単相2元対称MDS
Kruskal (1964a, b)
2相2元MDS
Carroll (1972)
2相3元
対称MDS
Carroll and Chang (1970)
3相3元MDS
Kroonenberg and
De Leeuw (1980)
単相3元MDS
De Rooij and
Gower (2003)
2相3元
非対称MDS
Okada and
Imaizumi
(1997)

Rでの分析可能モデル


単相2元
非対称MDS
namsMDS()
単相2元対称MDS
cmdscale()
isoMDS()
sammon()
2相2元MDS
smacofRect()
2相3元
対称MDS
smacofIndDiff()
2相3元
非対称MDS
namsMDS()
3相3元MDS
CANDECOMP()
TUCKER()
単相3元MDS

計量MDSとは

計量MDS

- 分析に使用する対象間の(非)類似度データは，原則として比尺度の水準で得られていなければならない
- データは順序尺度の水準で得られていれば，対象は多次元距離空間内の点として位置づけることが可能
- データの水準に対する制約がすくないという特徴

計量MDS

- 複数の対象間の非類似度が比率尺度（特に，対象間のユークリッド距離として推定されている場合
- 固有値分解により固有値と固有ベクトルを求め，対象を多次元ユークリッド空間内に位置づける

- 実際には，得られている非類似度データが比尺度の水準で得られていても，距離の性質を満たしているとは限らない
- データに定数(加算定数)を加えることで，距離の性質を満たすように変換して分析

計量MDS

- m個の対象(製品やブランドなど)のペアーについて類似度あるいは非類似度が得られているとする
  - 最終的に推定したいのはm個の対象の空間布置$\boldsymbol{X}$($m \times r$の座標行列)

- Young and Householder(1938)が提起した問題意識
  - 未知の$\boldsymbol{X}$が観測されたデータとどのような関係になっていれば望ましいだろうか

計量MDS

- Torgerson (1952)の古典的MDSとGower (1966)の主座標分析の考え
  - もし$\boldsymbol{X}$によって測られるt次元空間での対象間のユークリッド距離が非類似度に近似するなら，空間布置と人間の判断データには斉合性
  - 両者の違いは古典的MDSが非類似度データから出発するのに対し，主座標分析は類似度データから出発する

- 関数cmdscale()が採用している方法は古典的MDS(Torgerson の方法)

計量MDS： Torgersonの方法

- 固有値分解により，多次元空間における対象間の布置を求めることが可能
- 実際には，得られている非類似度データが比尺度の水準で得られても，距離(ユークリッド距離)の性質を満たしているとは限らない
  - データを観測する際や，判断を行う際に誤差が含まれ，得られた非類似度のデータが距離に正確には対応しない
  - 実際には得られている非類似度データが，距離に対応していない
- Torgersonの方法では，距離に定数(加算定数)を加えることで，距離の性質を満たす場合に適応

計量MDS：加算定数

- 正確な距離の性質を備えた対象間の真の距離を推定する必要

- 距離の性質を備えていない距離は比較距離h_{ij}と呼ばれ，距離空間の特性を備えている真の距離d_{ij}と比較距離h_{ij}との間には，

$$d_{ij} = h_{ij} + \mathrm{c}$$

- という関係が成立

- 定数cは加算定数と呼ばれる

計量MDS：加算定数

- 真の距離d_{ik}はユークリッド空間の場合，3つの対象が一直線上にあれば，

$$d_{ij}=d_{ik}+d_{jk}$$

- が成立

- 真の距離d_{ij}と比較距離h_{ij}との間の関係式から，

$$c=h_{ij}-h_{ik}-h_{jk}$$

- 実際には，全ての対象が一直線上にあるとは限らないので，直線上にない3つの対象間の距離について三角不等式が成り立つことを利用
- 真の距離d_{ij}と比較距離h_{ij}との間の関係式から，

$$c \geq h_{ij}-h_{ik}-h_{jk}$$

- が成り立つ

計量MDS：加算定数

- すべての3つの対象の組について$h_{ij}-h_{ik}-h_{jk}$の最大となるc_1を計算
- このc_1を加算定数の下限

$$c \geq c_1 = \max(h_{ij}-h_{ik}-h_{jk})$$

- 真の距離d_{ij}と比較距離h_{ij}との間の関係式から対象間の真の距離d_{ij}が負にならないとの制約を課す
- 全ての2つの刺激に関する比較距離h_{ij}の符号を反転したものを最大値を$c_2=\max(-h_{jk})$とする
- 加算定数cを

$$c = \max(c_1, c_2)$$

と求める

計量MDS：関数cmdscale()での実行

- 関数cmdscale で計量MDSを実行するには

  > cmdscale(data, k = 2,eig=FALES･･･)

- 引数dataは距離構造のデータ
- 引数kは次元数で，デフォルトでは2に設定
- 引数eigは固有値を返すか否かを指定
  - デフォルトではFALESになっており固有値を返さない
  - 引数eig=TRUEにすると座標値は$points に記録

簡単な実習

計量MDSをRで実行

計量MDS：関数cmdscale()での実行

- 関数cmdscale で計量MDSを実行するには

  > cmdscale(data, k = 2,eig=FALES･･･)

- 引数dataは距離構造のデータ
- 引数kは次元数で，デフォルトでは2に設定
- 引数eigは固有値を返すか否かを指定
  - デフォルトではFALESになっており固有値を返さない
  - 引数eig=TRUEにすると座標値は$points に記録

入力データ

- データは距離行列

	Athens	Barcelona	Brussels	Calais
Barcelona	3313			
Brussels	2963	1318		
Calais	3175	1326	204	
Cherbourg	3339	1294	583	460

- もしくは，非類似度で表される対称行列

	Athens	Barcelona	Brussels	Calais	Cherbourg
Athens	0	3313	2963	3175	3339
Barcelona	3313	0	1318	1326	1294
Brussels	2963	1318	0	204	583
Calais	3175	1326	204	0	460
Cherbourg	3339	1294	583	460	0

データの読み込み・分析

- eurodistというデータが組み込まれている
  - 今回はこれを利用
  - 距離行列（下三角）
  - ヨーロッパ21都市の距離

- result1<-cmdscale(eurodist, add=T)
  - 加算定数を用いる
  - 2次元で分析

結果と布置

```
result1$points
             [,1]        [,2]
 [1,]  2716.561820  3549.216493
 [2,] -1453.753109   455.895291
 [3,]   217.426476 -1073.442137
 [4,]     1.682974 -1135.742982
 [5,]  -461.875781  -871.913389
 [6,]   594.256798 -1029.818247
```

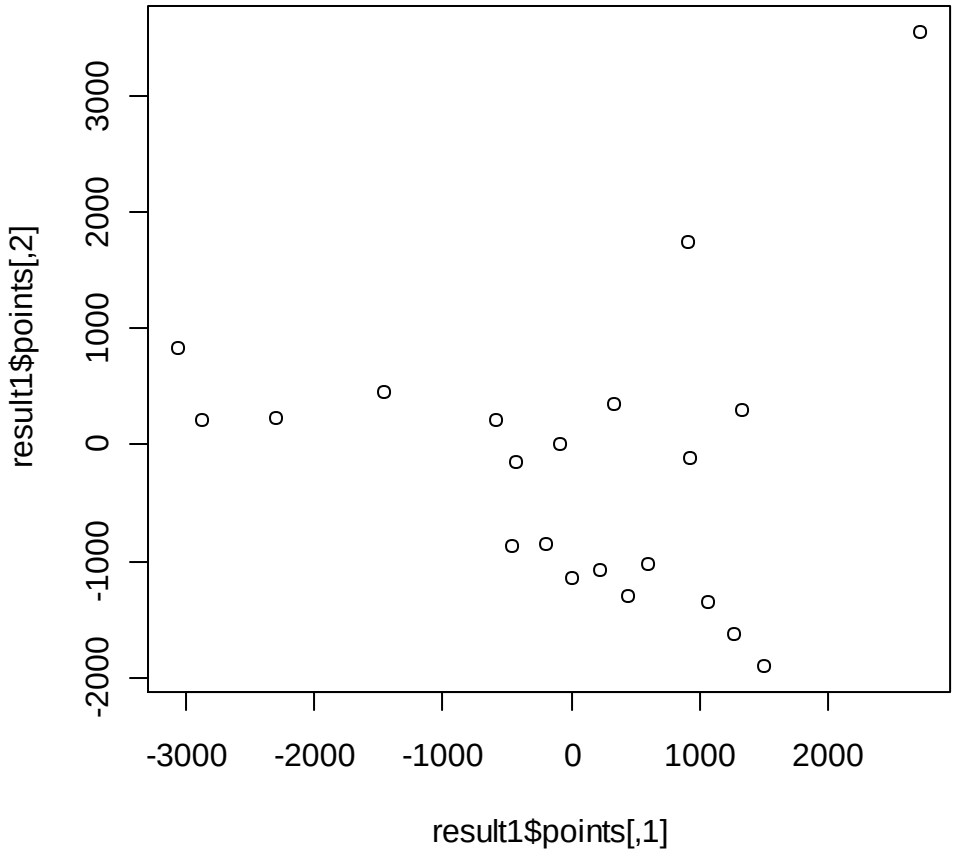
・

・

・

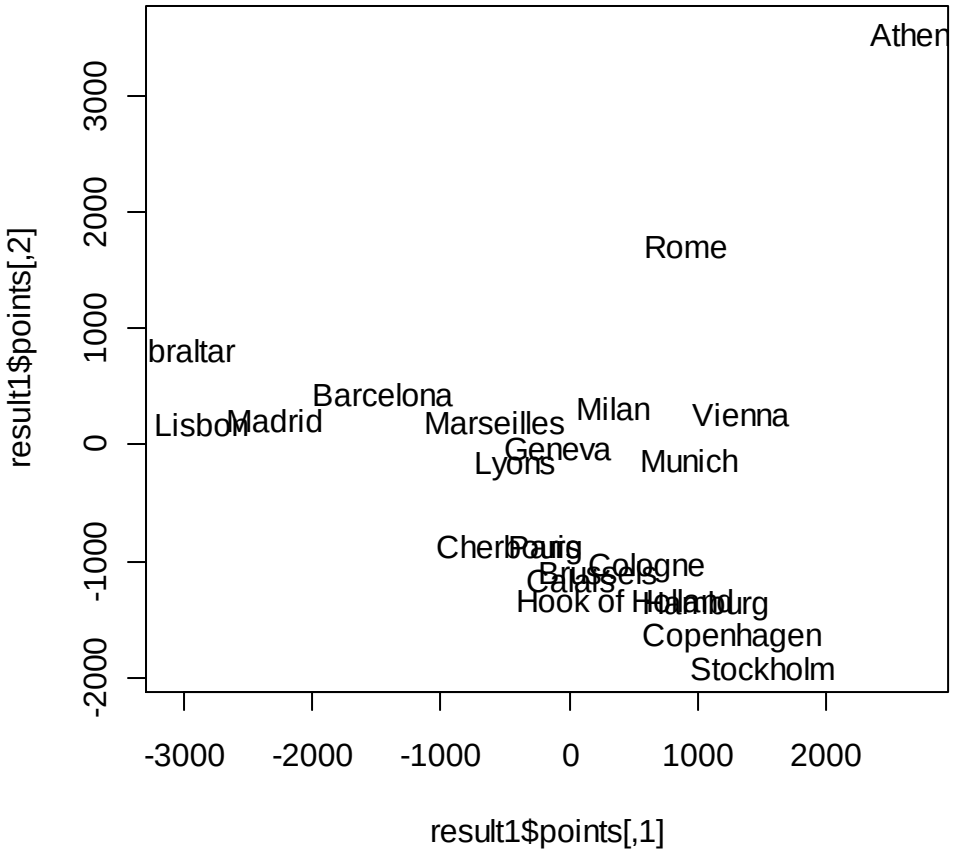
結果と布置

- plot(result1$points)
  - ただしどの点がどの都市を表すかわからない



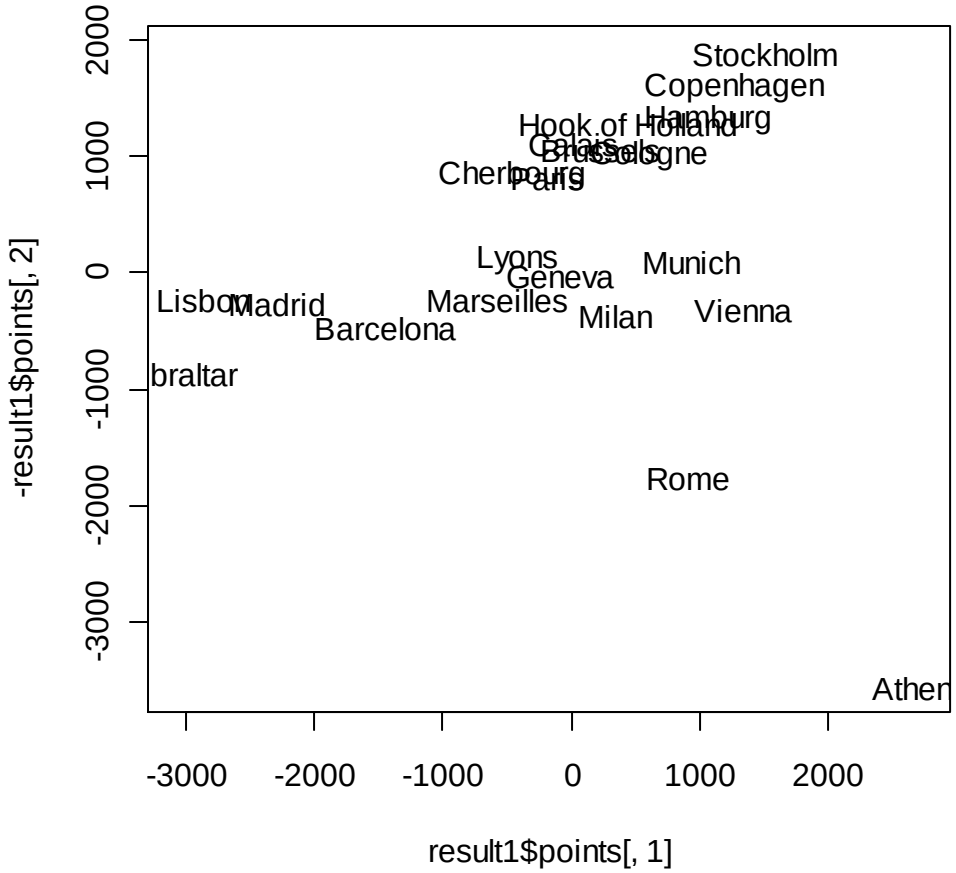
布置

- plot(result1$points, type="n")
- text(result1$points, labels(eurodist))



布置（y軸を反転させる）

- plot(result1$points[,1], -result1$points[,2], type="n")
- text(result1$points[,1], -result1$points[,2], labels(eurodist))



非計量MDSとは

非計量MDS

- Kruskalの方法が最も一般的
  - 2つの対象間の (非) 類似度関係を分析するための多次元尺度構成法である

- 類似度は少なくとも順位尺度であれば良い
- 類似度と距離の関係は単調関係（順序関係）を満たせばよく，線形関係を満たす必要性はない
- 低次元での当てはまりが良く，視覚的理解がしやすい
- 様々な距離に適用可能
- 欠測値に対する許容性ある

Kruskalの方法

- 対象間の関係を，類似度と対象間の距離とが単調関係を満たすように対象を点として多次元空間内に位置付けることで表現
  - 類似度の大きい対象同士はそれぞれの対象を表現する点間の距離は小さく
  - 類似度の小さい対象同士はそれぞれの対象を表現する点間の距離が大きく

- 次のような単調関係をできるだけ満たすように多次元空間内の点の座標を決定

$$\delta_{ij} > \delta_{rs} \Rightarrow d_{ij} \geq d_{rs}$$

ただし，δ_{ij}は対象iとjの非類似度，d_{ij}は対象iとjの距離

Kruskalの方法

- 布置では，対象間の類似度の大小の関係が，各対象の点間の距離によって視覚的に把握可能
  - 対象を多次元空間内に位置づけた図を布置と呼ぶ

- 対象iの次元tでの座標をx_{it}，pを多次元空間の次元数としたときユークリッド空間での点間距離

$$d_{ij} = \sqrt{\sum_{t=1}^{p} (x_{it} - x_{jt})^2}$$

ミンコフスキーの一般距離

- ミンコフスキーの一般距離(M:ミンコフスキー定数)

$$d_{ij} = \left(\sum_{t=1}^{p} \left| x_{it} - x_{jt} \right|^{M} \right)^{1/M} \qquad (M \geq 1)$$

- ユークリッド距離はM=2の場合
- 非ユークリッド距離：M=2以外の場合
  - M=1 のとき，市街地距離 (マンハッタン距離)
  - M=∞の場合は優勢次元距離

ミンコフスキーの一般距離

- 通常はユークリッド距離で分析することが一般的
- 各次元の示す特性が個別に識別できる場合には非ユークリッド距離 (特に市街地距離) を用いる場合もある
  - 何種類かのMを用いて分析
  - 次元数を決定した上で，最小のストレスが得られたMの結果を解
  - 非ユークリッド距離を用いた分析から得られた布置は次元の方向が一意に定まり，回転できない

Kruskalの方法

- 多次元空間内の対象を表す点の位置を決定するためには，はじめに，ある次元における*n*個(分析対象の対象数)の点の仮の位置を決定
- 点間距離が類似度と単調減少関係になるように，多次元空間内での対象の点の座標を徐々に改善
- 最適な位置関係となる各対象の座標を算出

- 対象の座標を求めていく過程
  - 対象間の類似度と対象を表す点間の距離が必ずしも単調関係を満たさない場合もある

- 多次元空間内の点の座標を決定
  - 単調関係をどの程度満たしているのかを示す尺度である不適合度(ストレス)を基準とする

適合度(ストレス)

- 距離とデータの単調関係の当てはまりの程度示す尺度
  - ストレス1

$$S = \sqrt{\sum_{i}^{n}\sum_{j<i}^{n}(d_{ij} - \hat{d}_{ij})^2 \Big/ \sum_{i}^{n}\sum_{j<i}^{n} d_{ij}^2}$$

ただし，$\hat{d}_{ij}$は単調関係を満たし，距離に対して分子を最小とする値

  - ストレス2(解の退化を防ぐ)

$$S = \sqrt{\sum_{i}^{n}\sum_{j<i}^{n}(d_{ij} - \hat{d}_{ij})^2 \Big/ \sum_{i}^{n}\sum_{j<i}^{n}(d_{ij} - \bar{d})^2}$$

布置の基準化

- ストレスは点間の距離にもとづいて定義されているため，布置の大きさ，原点の位置，次元の方向には影響を受けない

- 布置の重心が原点となるように

$$\sum_{i=1}^{n} x_{it} = 0 \quad (t = 1, \cdots, p)$$

- とし，また，座標値の2乗和が対象の個数に等しくなるように，

$$\sum_{i=1}^{n} \sum_{t=1}^{p} x_{it}^2 = n$$

と布置を基準化する必要がある

- 具体的には，平均が0，分散が1となるように得られた布置の座標を基準化したものを次元数の平方根で除す

ストレス*S*が最小となる座標の導出

- 制約のない最適化問題を解くための勾配法である最急降下法を利用
  - ストレス*S*が最小となる方向に解 (座標値) を探索
- 次元数*t*のもとでストレスを最小化する布置を求めるとすると,
  - 仮の*t*次元布置 (初期布置) を求める
  - この布置のストレスが減少するように対象の位置を少しずつ動かして布置を反復的に改善
  - ストレスが一定以上減少しなくなるまで反復を続行

ストレスSが最小となる座標の導出

- このときに得られる布置
  - ストレスがそれ以上改善できない布置(局所極小布置)
  - ストレスが必ずしも最小であるとは限らないことに注意
- 初期の座標値をどのように与えるかに解が依存
- 局所最適解 (局所的な最小値であって，全体の最小値ではない値) が得られてしまう場合がある
- 様々な初期値を用いて分析を行い，ストレスの最小値が得られている結果を解として採用する必要

ストレス*S*が最小となる座標の導出

- ストレス*S*は点間の距離によって定義されるため，あらかじめ布置の次元数を決めておく必要
- 布置の次元数がいくつであるかは通常は分析を行う前には不明
- 何種類かの次元数のもとで分析を行い，各次元数においてストレスを最小にする布置を求める
- 得られたストレスや布置を検討して何次元の布置を解にするかを決定

簡単な実習

非計量MDSをRで実行

非計量MDSのプログラム

- 今回はisoMDSとsammonを用いる
  - ストレスの計算方法等が異なる
  - library(MASS)の読み込みが必要
- isoMDS(d, k = 2)
- sammon(d, k = 2)
  - d : データ（距離形式か非類似度正方行列）
  - k : 次元

準備

- データの読み込み
  - tourism<-read.table("tourism.dat", header=T, sep="¥t", row.names=1)
- 類似度データなので非類似度データに
  - tourism.dis<-8-tourism
- データを距離行列に変換する
  - tourism.dist<-as.dist(tourism.dis)

	Greece	Hawaii	West Coast	Hong Kong	LondonParis	East America
Hawaii	5					
West Coast	3	1				
Hong Kong	7	4	6			
LondonParis	2	7	4	7		
East America	3	6	2	5	1	
Australia	2	3	2	6	4	5

isoMDSの実行

- tourism.iso<-isoMDS(tourism.dist, k=2)

initial value 11.510834
iter   5 value 2.153903
iter 10 value 1.845324
iter 15 value 1.734993
iter 15 value 1.733410
iter 15 value 1.733410
final value 1.733410
converged
と出力

isoMDSの実行

- tourism.isoで結果が見れる

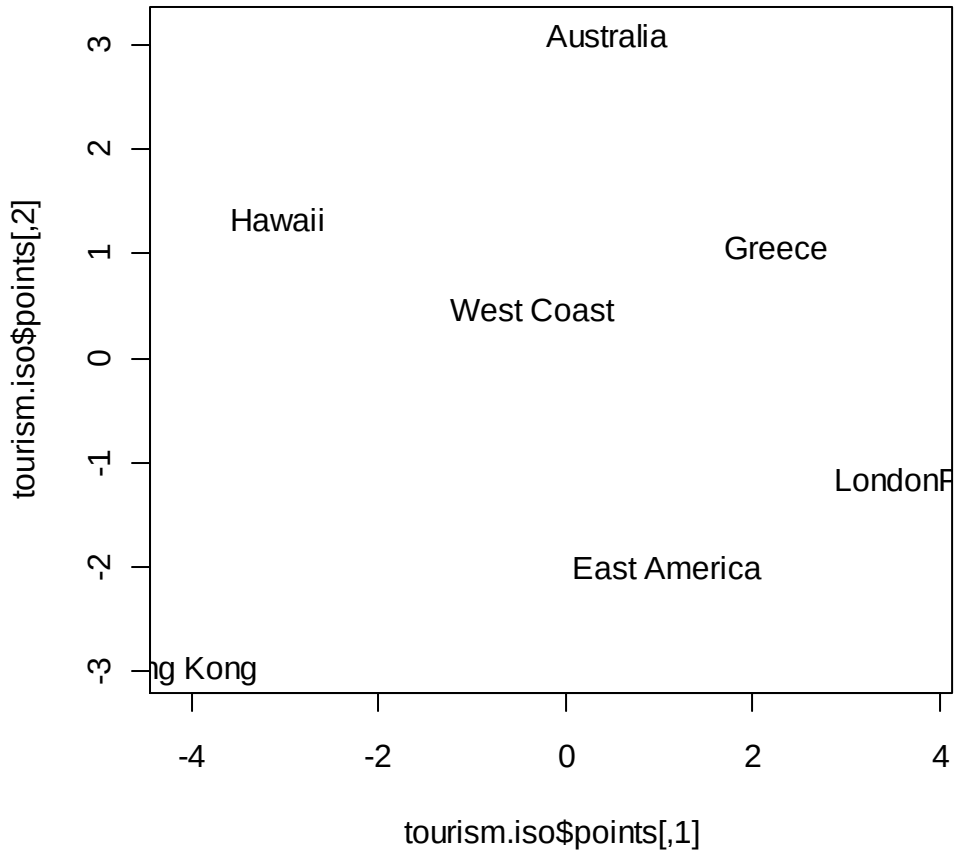
```
$points
                [,1]      [,2]
Greece       2.2442891  1.089808
Hawaii      -3.0727925  1.358510
West Coast  -0.3686213  0.497304
Hong Kong   -4.1223052 -2.955376
LondonParis  3.8157995 -1.130995
East America 1.0741955 -1.976969
Australia    0.4294350  3.117718

$stress
[1] 1.73341
```

isoMDSの実行

- $pointsが布置，$stressがストレス
- Tourism.iso$points，tourism.iso$stressで直接値を出力できる
- 布置を描く
  - plot(tourism.iso$points)
  - ラベル付きなら
    - plot(tourism.iso$points, type="n")
    - text(tourism.iso$points, rownames(tourism.iso$points))

isoMDSの実行

- 布置



sammonの実行

- tourism.sammon<-sammon(tourism.dist, k=2)

  Initial stress : 0.10883

  stress after 10 iters: 0.02960, magic = 0.500

  stress after 20 iters: 0.02960, magic = 0.500

  と出力

sammonの実行

- tourism.sammonで結果が見れる

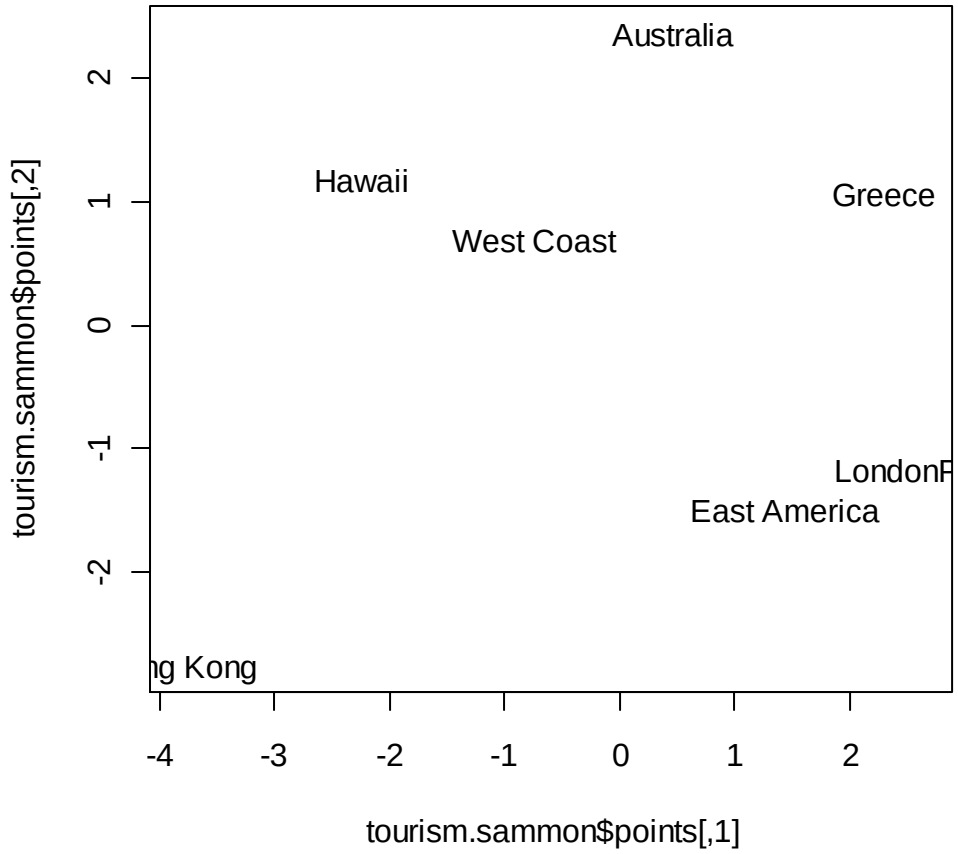
```
$points
                [,1]       [,2]
Greece        2.2905441  1.0953602
Hawaii       -2.2375811  1.2020838
West Coast   -0.7482011  0.7155929
Hong Kong    -3.8165712 -2.7563568
LondonParis   2.6331857 -1.1547582
East America  1.4242927 -1.4781861
Australia     0.4543310  2.3762642

$stress
[1] 0.02959711

$call
sammon(d = tourism.dist, k = 2)
```

sammonの実行

- $pointsが布置，$stressがストレス，$callが実行したコマンド
- tourism.sammon$points，tourism.iso$stressで直接値を出力できる
- 布置を描く
  - plot(tourism.sammon$points)
  - ラベル付きなら
    - plot(tourism.sammon$points, type="n")
    - text(tourism.sammon$points, rownames(tourism.sammon$points))

sammonの実行

- 布置



パソコン多次元尺度構成法での結果

- 布置