Contents

Scaling Techniques

- Introduction
  - PCA starts with a 'Euclidean n×p data matrix'
    - **n** cases **p** variables, continuous measurements
  - Produces low dimensional data set
    - display in a few scatterplots
  - Look at subgroups, outliers etc
  - i.e. look at ***distances between cases***
    - distances calculated from low number of dimensions
      - **¿Can we do this in reverse?**

- i.e. start with distances and retrieve low dimensional Euclidean data matrix
- Why?
  - Some data inherently 'distances'
  - Some data inherently 'disimilarities'
    - train journey times reflect distances between towns
    - numbers of times two Morse code symbol confused
      - reflects how similar/dissimilar the codes are
      - similar/dissimilar as perceived by the brain

- Converting these to [Euclidean] data allows more routine statistical analysis
  - e.g scatterplots &c to discover structure of [&/or relationship between] cases
- Similarity/dissimilarity measures exchangeable
  - convert from one to other
    - e.g. subtract from 100 or take reciprocals or ……….
  - Which is used depends on context
- Measure of dissimilarity may be very rough
  - e.g. on 0/1 scale : 0 ≡ dissimilar, 1 ≡ similar

Departments of France

Departments i and j are similar if they have a common border dissimilar if no common border

δ_{ij}=1 if common border

δ_{ij}=0 if no common border

Departments of France

Departments i and j are similar if they have a common border dissimilar if no common border

δ_{ij}=1 if common border

δ_{ij}=0 if no common border

δ_{12}=1

Departments of France

Departments i and j are similar if they have a common border
dissimilar if no common border

δ_{ij}=1 if common border

δ_{ij}=0 if no common border

δ_{12}=1

δ_{24} = 0

Departments of France

	1	2	3	4	…	9
1	★	1	0	0	0	0
2	1	★	0	0	0	0
3	0	0	★	1	0	0
4	0	0	1	★	…	
:					★	
9						★
:						

Departments of France

	Department									
	1	**2**	**3**	**4**	**……..**	**14**	**15**	**16**	**17**	**…**
1	★	1	0	0	……..	1	0	0	0	…
2	1	★	0	0	……..	1	1	0	0	…
3	0	0	★	1	……..	0	1	1	0	…
4	0	0	1	★	……..	0	0	0	0	…
………………………………………………………										
14	1	1	0	0	……..	★	1	0	0	…
15	0	1	1	0	……..	1	★	1	0	…
16	0	0	1	0	……..	0	0	★	1	…
17	0	0	0	0	……..	0	0	1	0	…
…	…	…	…	…	……..	…	…	…	…	…

The similarity matrix

Similarities between graves

- Graves containing artefacts of M types
  - 0 ≡ artefact absent: 1 ≡ artefact present

	Artefact														
	1	**2**	**3**	**4**	**5**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**M**
grave i	0	0	1	1	.	.	1	0	1	0	0	.	.	.	1
grave j	0	1	1	0	.	.	0	0	1	0	0	.	.	.	.

define δ_{ij} = # in common in graves i & j
= 2

(many other measures of similarity possible)

- **Objectives**
  - learn more about the measure itself
    - e.g. Morse code :—
      what factors cause brain
      to confuse Morse symbols?

1: • – – – –   6: – • • • •
2: • • – – –   7: – – • • •
3: • • • – –   8: – – – • •
4: • • • • –   9: – – – – •
5: • • • • •   0: – – – – –

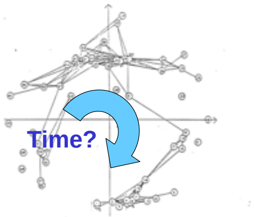
signal transmitted

signal received

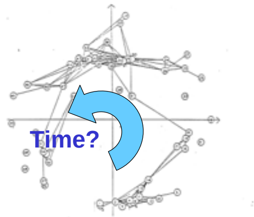
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	84	62	16	6	12	12	20	37	57	52
	2	89	59	23	8	14	25	25	28	18
		3	86	38	27	33	17	16	9	9
			4	89	56	34	24	13	7	7
				5	90	30	18	10	5	5
					6	86	65	22	8	18
						7	85	65	31	15
							8	88	58	39
								9	91	79
									10	94

% times signal **i** received as **j**

- discover underlying structure in data
  - e.g. dissimilarities matched by points on line can line be interpreted as variable of interest
  - e.g. time in the prehistoric graves example



- discover underlying structure in data
  - e.g. dissimilarities matched by points on line can line be interpreted as variable of interest
  - e.g. time in the prehistoric graves example



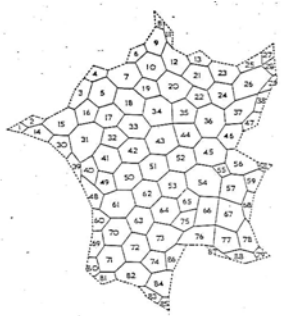
- do cases divide ‘naturally’ into groups ?
  - used in market research for ‘market segmentation’
  - do the customers divide into different target groups ⇒ different forms of advertising
- is there a ‘gap’ in the market ?
  - fill with new product
  - e.g. yellow plum tomatoes on vine
  - white flake chocolate bar (snowflake)
    - (could use ‘cluster analysis’ for groupings — see appendix — but wont detect ‘gaps’)

- **Validation of technique**
  - e.g. where answer is known (e.g. map of France)
    - If map can be reconstructed with rough abuttal data suggests that technique ***works*** so map drawn from other data is ***valid.***
    - e.g. reconstructing map of ancient towns
      - inscriptions on *Linear B* clay tablets
      - two towns scored 1 names on same tablet
      - 0 otherwise
      - (tablets referred to trading between the towns)

Departments of France

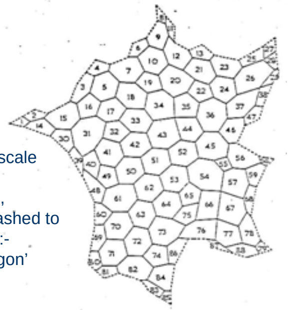
Map of the Departments of France

Departments of France



Reconstructed Map of France

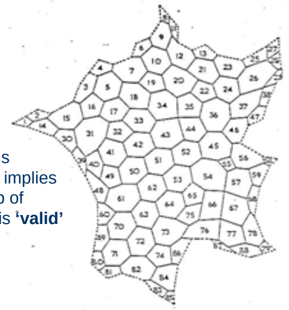
Departments of France



Orientation and scale initially arbitrary: 'solution' rotated, stretched & squashed to fit overall outline:-
The 'Hexagon'

Reconstructed Map of France

Departments of France



Reconstruction is **'successful'** so implies constructed map of Assyrian towns is **'valid'**

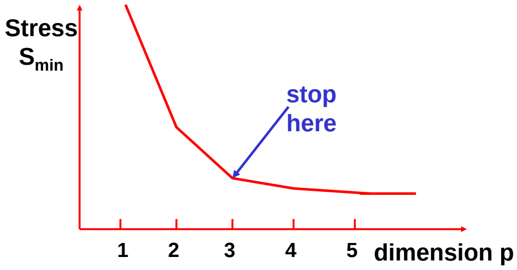
Reconstructed Map of France

- Other examples
  - British towns by road
    - Validation of various iterative techniques
  - Artificial data set of graves
    - Objects come into fashion and then fashion wanes
    - Reconstructed sequence is near perfect
    - ⇒ sequence on real graves is '**valid**'

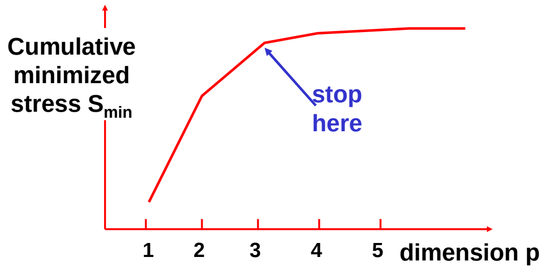
- **Note:**
  - All of these are *'proof by example'*
    - If techniques '*works'* when we know answer then believe answers using same methods
  - Little mathematical theory to *justify* results
    - But much mathematics to construct methods
  - One theorem available in one simple case rest is *by analogy* (e.g. with PCA)
    - Scree plots
    - examination of eigenvalues
    - &c.

- **Setup:** n×n matrix (δ_{ij}) of dissimilarities
  - Objective is to find n p-dimensional points with calculated distance matrix (d_{ij}) which matches the (δ_{ij}) as closely as possible
- Metric methods:-
  - Aim to achieve $d_{ij} = \delta_{ij}$ all i & j
    - or at least ≈ at least for most i & j
- Non-Metric methods
  - Aim to achieve d_{ij} have same relative ordering as δ_{ij}, at least weakly (i.e. ≤ if not actually <)

- Non-Metric methods
  - Methods typically iterative
    - Start with a trial configuration
    - Perturb configuration to improve match
    - Measure quality of match by a ***stress function***
    - Obtain minimum possible **stress S_{min}**
      - Since iterative no guarantee of finding global minimum
      - Try variety of initial trial configurations
    - Repeat this for p = 1, 2, 3, ……. and choose 'best value of p' by a **scree graph**



Scree graph of minimized stress vs dimension



Scree graph of cumulative minimized stress vs dimension

(may be easier to spot kink in cumulative graph of trade-off between less stress and increasing dimension (but be careful in interpretation…

- Note that interpretation of this graph is ***by analogy*** with PCA etc
  - ***Stress*** is an *ad hoc* function which measures how well a given configuration produces distances d_{ij} matching the given δ_{ij}
  - No strict mathematical interpretation in terms of 'quantity of stress' **unlike** PCA and 'amount of information'
    - so be careful interpreting cumulative stress
  - Variations in methods differ mainly in form of stress function
    - e.g. Kruskal, Sammon, ….

- Implementation:
  - Specialist software:
    - CD-Rom supplied with Cox & Cox, (2001) Multidimensional Scaling, Chapman & Hall.
  - Packages:
    - S-PLUS: functions **`isoMDS(.)`** and **`sammon(.)`**
    - Not in Minitab
    - Not in SPSS
      - note that Analyze>Scale>Multidimensional scaling gives classical scaling (see later)

- **Metric Methods**
  - Classical scaling (Principal Coordinates)

  Start with a matrix $D=(d_{ij})$ of distances

  Derive a matrix B by

  (i) define $A=(a_{ij})$ by $a_{ij} = -\frac{1}{2} d_{ij}^2$

  (A is a similarity matrix)

  (ii) B = A (centred by row and column means)

  Then

- **Theorem**
  - D is Euclidean iff B is positive semi-definite
  - Euclidean ≡ there is a configuration of n points X′ in Euclidean p-space whose interpoint distances are given by D
  - B is positive semi-definite ≡ $B \geq 0$ ≡ all the eigenvalues of B are non-negative
    - Details of proof: see notes

- Key steps of proof 'f' part:
  - Note that $d_{ij}^2 = (x_i - x_j)'(x_i - x_j)$
  - Verify that $b_{ij} = (x_i - \bar{x})'(x_j - \bar{x})$
  - Recognise that a matrix of the form

    $B = Z'Z$ is ≥ 0

- 'if' part:
  - Proof is by construction of points and then verification that distance matrix is as desired
  - $B \geq 0$ is important since need square roots of eigenvalues of B
  - Take the p non-trivial eigenvectors of B
    - i.e corresponding to the p non-zero eigenvalues
  - Arrange these p eigenvectors in a $n \times p$ matrix
    - (remember B is $n \times n$ so eigenvectors are n-vectors)
  - This matrix X′ is the data matrix required

- Note that if we start with an $n \times p$ X′ data matrix
  - Calculate distance matrix D
  - Derive A and then centre it to get B
  - Get the eigenvectors of B
- We get the p PCs of X′ (transposed)
- i.e. same p-dimensional map of n points but rotated &/or reflected
  - **i.e. applying classical scaling methods to ordinary p-dimensional numeric data is POINTLESS**

- **Assessing number of dimensions**
  - Plot stress *vs* number of dimensions
  - Plot or cumulative stress *vs* # of dimensions
  - Plot eigenvalues *vs* number of dimensions
  - Plot cumulative partial sums of eigenvalues *vs* number of dimensions (**ONLY IF ALL** $\geq$ **0**)
  - If some eigenvalues < 0 then possibilities are
    - Discard any < 0
    - Sum modulus of eigenvalues
    - Sum squares of eigenvalues
  - DO NOT USE eigenvectors corresponding to negative eigenvalues
    - – controversial as to information in & use of such eigenvectors

- Implementation:
  - **Specialist software:**
    - CD-Rom supplied with Cox & Cox, (2001) Multidimensional Scaling, Chapman & Hall.
  - **Packages:**
    - **R** and S-PLUS: functions `cmdscale(.)`
    - SPSS Analyze>Scale>Multidimensional scaling
    - Minitab: use command line or write macro
      - – **(see notes)**

- Examples: **Morse codes**

1: ● – – – – 6: – ● ● ● ●
2: ● ● – – – 7: – – ● ● ●
3: ● ● ● – – 8: – – – ● ●
4: ● ● ● ● – 9: – – – – ●
5: ● ● ● ● ● 0: – – – – –

signal transmitted

signal received

	1	2	3	4	5	6	7	8	9	10
1	84	62	16	6	12	12	20	37	57	52
2		89	59	23	8	14	25	25	28	18
3			86	38	27	33	17	16	9	9
4				89	56	34	24	13	7	7
5					90	30	18	10	5	5
6						86	65	22	8	18
7							85	65	31	15
8								88	58	39
9									91	79
10										94

% times signal **i** received as **j**

- Eigenvalues:
  - Two negative values so 'distances' are not Euclidean
    - But 'nearly' Euclidean (intuitively)

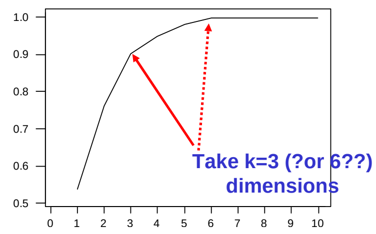
```
187.376
121.036
 95.443
 55.405
 46.583
 31.547
  9.628
  4.485
 -4.103
 -0.001
```


- Eigenvalues:
  - Two negative values so 'distances' are not Euclidean
    - But 'nearly' Euclidean (intuitively)

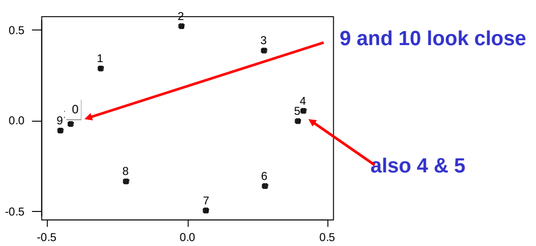
```
187.376
121.036
 95.443
 55.405
 46.583
 31.547
  9.628
  4.485
 -4.103
 -0.001
```

Two negative eigenvalues → -4.103, -0.001

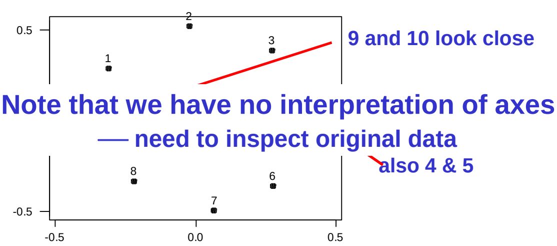
Scree plot of squared eigenvlaues

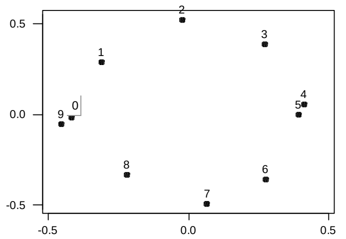


Plot on first two principal coordinates



Plot on first two principal coordinates





1: • – – – –
2: • • – – –
3: • • • – –
4: • • • • –
5: • • • • •
6: – • • • •
7: – – • • •
8: – – – • •
9: – – – – •
0: – – – – –

Ordering of dots & dashes

increasing number of dots

signal transmitted

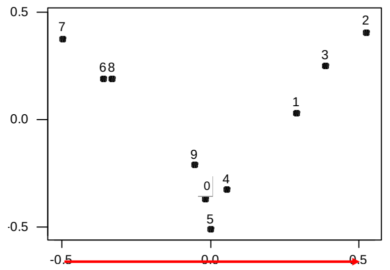
signal received

	1	2	3	4	5	6	7	8	9	10
1	84	62	16	6	12	12	20	37	57	52
2		89	59	23	8	14	25	25	28	18
3			86	38	27	33	17	16	9	9
4				89	56	34	24	13	7	7
5					90	30	18	10	5	5
6						86	65	22	8	18
7							85	65	31	15
8								88	58	39
9									91	79
0										94

% times signal **i** received as **j**

?

Plot on coordinates 2 & 3



Position of change from dot to dash

Ordering of dots & dashes

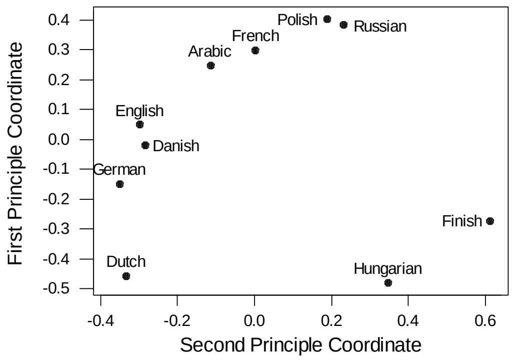
- Example: **Language discordancies**
  - Number words in different languages
    - One,two,three,four,….,ten
    - Ein,zwei,drei,vier,….,zehn
    - Un,deux,trois,quatre,…,dix
    - Egy,ketto,harom,negy,…,tiz
    - О д и н , д в а , т р и , ч е т ы р е ,…, д е с я т ъ
    - Um, dois, três, quatro,…,dez
    - ενα,δυο,τρια,…,δεκα
  - Define dissimilarity/distance as number of numerals with **different** first letters

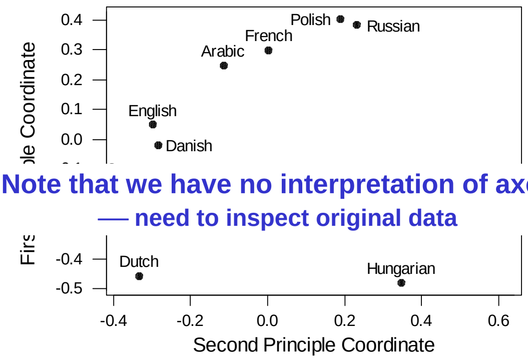
English & German have 6 first letters which are different
(& 4 in common: five/fünf, six/sechs, seven/sieben,nine/neun)

Row	Language	Eng	Dan	Dut	Ger	Fre	Pol	Hun	Fin	Rus	Arab
1	English	0	2	7	6	6	7	9	9	7	6
2	Danish	2	0	6	5	6	6	8	9	7	7
3	Dutch	7	6	0	5	9	10	8	9	10	10
4	German	6	5	5	0	7	8	9	9	8	8
5	French	6	6	9	7	0	5	10	9	5	7
6	Polish	7	6	10	8	5	0	10	9	2	7
7	Hungarian	9	8	8	9	10	10	0	8	10	10
8	Finnish	9	9	9	9	9	9	8	0	9	10
9	Russian	7	7	10	8	5	2	10	9	0	8
10	Arabic	6	7	10	8	7	7	10	10	8	0

- Eigenvalues:
  - Two negative eigenvalues
    - Eigenvectors corresponding to negative eigenvalues discarded

```
101.915
 66.901
 39.399
 29.916
 18.015
 13.180
  7.026
  4.699
 -3.651
 -0.000
```





Note that we have no interpretation of axes
— need to inspect original data

- **Minimum Spanning Trees**
  - A graph connecting all points
  - Shortest total length of all such trees
    - (therefore no circuits)
  - Actual distance between two points must be greater than maximum link in the MST.
  - Usually, **but not necessarily**, nearest neighbours are linked in MST

- **USE**
  - If 2-Dim scaling plot accounts for only a small proportion of stress then maybe a few points not well represented in those 2 dims
    - (need a further dimension to accommodate them)
  - If two points appear close in a scaling plot may be able to assess whether they are 'really close'
    - Could be well separated in further dimensions accounting for further stress

- If two points appear close in a scaling plot may be able to assess whether they are 'really close'
  - Could be well separated in further dimension
- If points are close together in plot but are ***NOT joined in MST*** then actual distance must be greater than longest link in path in tree from one to other
- **NB** cannot conclude anything if they are joined
  - **(so only partially useful)**

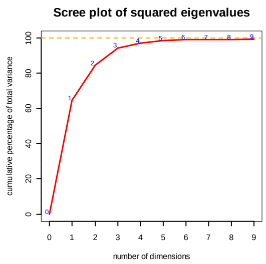
- Implementation in **R**
  - Many functions in downloadable libraries
    - (not available in basic packages nor MASS)
      - (but is in S+ version of MASS)
  - Illustrations use **`spantree()`** in package **`vegan`**
    - Many others available
    - See help system

- **Examples**
  - Morse code confusions
    - Are 0 and 9 ‘really’ close together?
      - MST inconclusive
  - Number words in languages
    - Are French and Arabic ‘really’ similar?
      - MST indicates more differences
  - note use of **`CMDscree()`**
    - (available from course webpages)

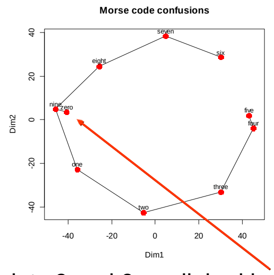
- Morse code confusions

```
> library(MASS)
> library(vegan)
This is vegan 1.17-12
> load("morse.Rdata")
> CMDscreeplot<-
  function(mydata,raw=T,abs=T,maxcomp=10) {
+ n=min(dim(mydata))-1
.........(commands for function CMDscale()
.... }

> CMDscreeplot(morse,abs=FLASE)
Warning message:
In cmdscale(as.matrix(mydata), k = n, eig = TRUE) :
  only 7 of the first 9 eigenvalues are > 0
> morse.tr<-spantree(morse)
> plot(morse.tr,cmdscale(morse),pch=16,col="red",
+ cex=1.5,main="Morse code confusions")
> text(cmdscale(morse), adj=c(0.5,-0.6))
```

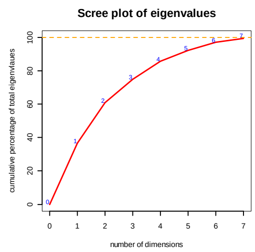
- Two components account for reasonable proportion of squared eigenvalues
  - Probably not much distortion in 2-dim plot



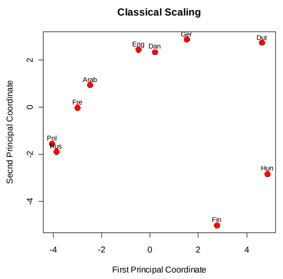
- Points 9 and 0 are linked in MST
  - Conclude only that ‘actual distance’ is greater than length of link

- Language similarities via number words

```
> library(MASS)
> library(vegan)
This is vegan 1.17-12
0
> CMDscreeplot(lang, maxcomp=7)
Warning message:
In cmdscale(as.matrix(mydata), k = n, eig = TRUE) :
  only 7 of the first 9 eigenvalues are > 0
# NB screeplot of only positive eigenvalues
> plot(-cmdscale(lang),pch=16,col="red",
+ cex=1.5, main="Classical Scaling",
+ xlab="First Principal Coordinate",
+ ylab="Secnd Principal Coordinate")
> text(-
  cmdscale(lang),names(lang),cex=0.8,adj=c(0.5,-0.6))
>
```

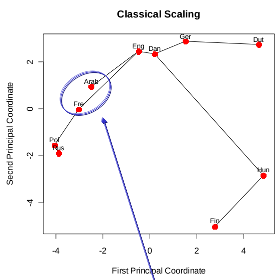
- Two components do not account for reasonable proportion of eigenvalues
  - Probably distortion in 2-dim plot



- Anticipate distortion noting screeplot
  - Where is it? --- look at MST

♦ Language similarities via number words

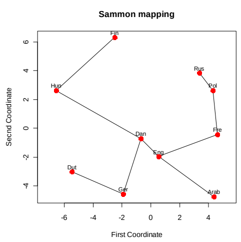
```
>
> lang.tr<-spantree(lang)
> plot(lang.tr,-mdscale(lang),pch=16,col="red",
+ cex=1.5, main="Classical Scaling",
+ xlab="First Principal Coordinate",
+ ylab="Secnd Principal Coordinate")
> text(-cmdscale(lang),names(lang),cex=0.8,
+ adj=c(0.5,-0.6))
>
>
```

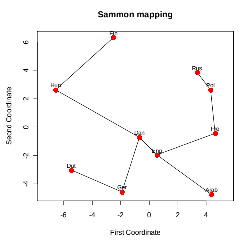



- French & Arabic close in plot but **NOT** joined in MST
  - Distortions here
  - 'real distance must be greater than link between English & French

♦ Look at trees on Sammon & Kruskal versions

```
> plot(lang.tr,sammon(lang),pch=16,
+ col="red",cex=1.5,
+ main="Sammon mappingg",
+ xlab="First Coordinate",
+ ylab="Secnd Coordinate")
> text(sammon(lang)$points,names(lang),
+ cex=0.8,adj=c(0.5,-0.6))
> m.iso<-isoMDS(lang,cmdscale(lang))
> plot(lang.tr,m.iso,pch=16,col="red",cex=1.5,
+ main="Kruskal scaling",
+ xlab="First Coordinate",
+ ylab="Secnd Coordinate")
> text(m.iso$points,names(lang),cex=0.8,
+ adj=c(1.1,-0.6))
```



- Both of these techniques separate French & Arabic more successfully

- For further examples see lecture notes
- Notes
  - ♦ Doesn't really matter whether we use similarities or dissimilarities
    - (see tasks/exercises to come)
  - ♦ Other *'unsupervised learning'* techniques for investigating similarities include
    - Cluster Analysis (see Appendix 4)
    - Kohinen self-organising maps (see Appendix 9)

- **Summary**
  - Multidimensional scaling produces points in low dimensional space from matrix of interpoint distances
  - The distances can be a general measure of similarity or equivalently dissimilarity
  - The purpose may be to learn about the measure of (dis)similarity itself or to identify structure in the points themselves
  - Interpretation of axes is **ONLY** possible after inspection of original data

- **Summary(ct[d])**
  - Application when the 'answer is known' gives confidence when applying it to similar but unknown situations
    - (e.g. French Departments)
  - Principal Coordinate Analysis constructs a display.
    - If the distance matrix is Euclidean solution is exact
      - (up to arbitrary rotations and reflections)
    - Otherwise starting point for iterative techniques

- **Summary(ct[d])**
  - If non-Euclidean some eigenvalues < 0
    - be careful with scree plots
    - don't use corresponding coordinates
  - Principal Coordinate Analysis is the dual of Principal Component Analysis
  - Other *'unsupervised learning'* techniques are
    - Cluster Analysis (Appendix 4)
    - Kohinen self-organising maps (Appendix 9)