Finding Needles in Haystacks:

Tools for Finding Structure in Large Datasets

Brian D. Ripley

Visualization

Challenge is to explore data in more than two or perhaps three dimensions.

via projections

Principal components is the most obvious technique: kD projection of data with largest variance matrix (in several senses). Usually 'shear' the view to give uncorrelated axes.

Lots of other projections looking for 'interesting' views, for example groupings, outliers, clumping. Known as (exploratory) *projection pursuit*.

Implementation via numerical brute-force: freely available in XGobi.
'Random' searching (so-called *grand tours*) are not viable even in 5D.

Glyph representations

There are many ways to represent each case by a small diagram, of which Chernoff's faces are the most (in)famous.

Wilkinson, L. (1999) *The Grammar of Graphics*. Springer.

These glyph plots do depend on the ordering of variables and perhaps also their scaling, and they do rely on properties of human visual perception. So they have rightly been criticised as subject to manipulation, and one should be aware of the possibility that the effect may differ by viewer. (Especially if colour is involved; it is amazingly common to overlook the prevalence of red–green colour blindness.)

Alabama

Connecticut

Illinois

Maine

Missouri

New Mexico

Oregon

Texas

Wisconsin

Alaska

Delaware

Indiana

Maryland

Montana

New York

Pennsylvania

Utah

Wyoming

Arizona

Florida

Iowa

Massachusetts

Nebraska

North Carolina

Rhode Island

Vermont

Arkansas

Georgia

Kansas

Michigan

Nevada

North Dakota

South Carolina

Virginia

California

Hawaii

Kentucky

Minnesota

New Hampshire

Ohio

South Dakota

Washington

Colorado

Idaho

Louisiana

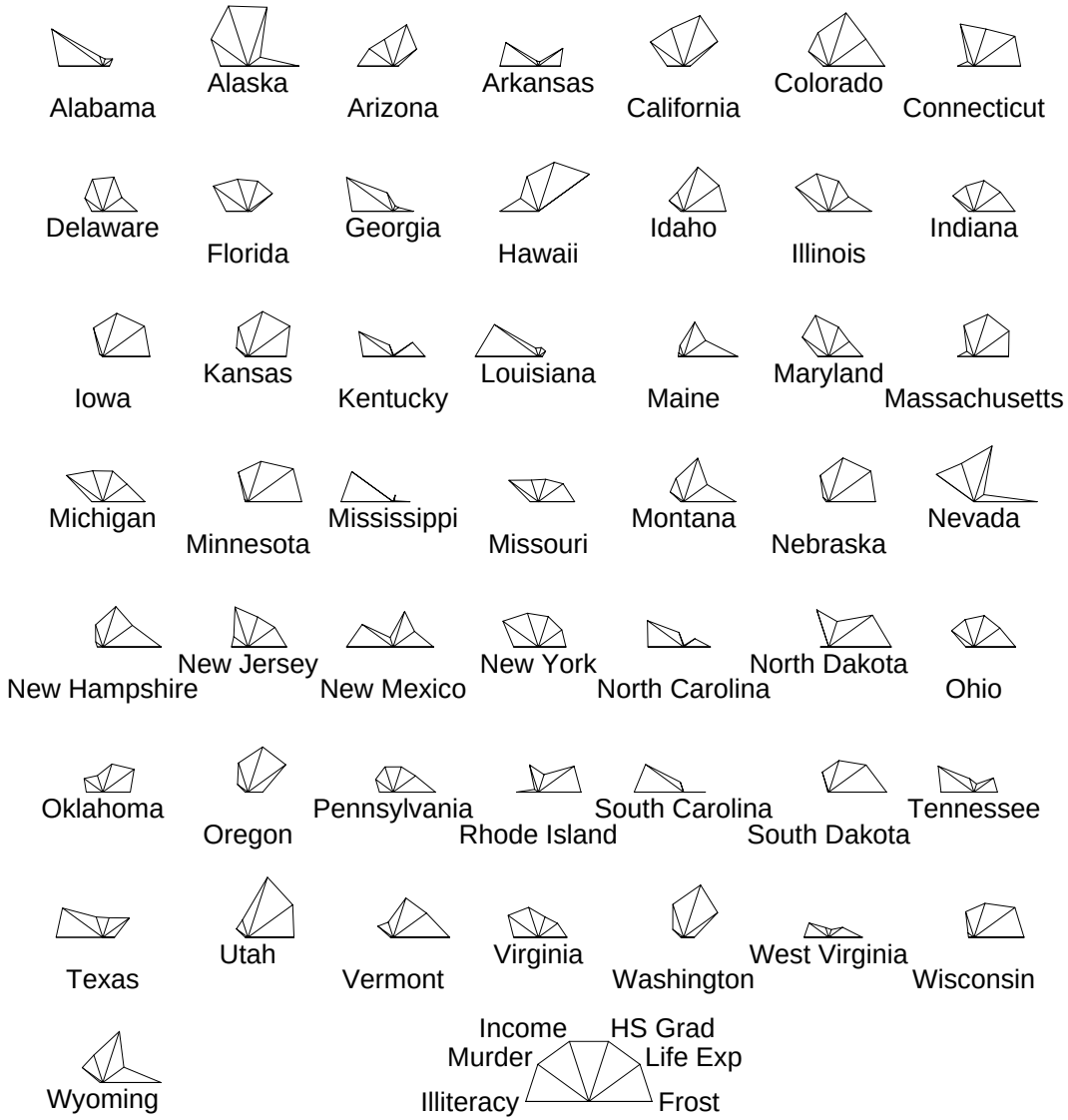
Mississippi

New Jersey

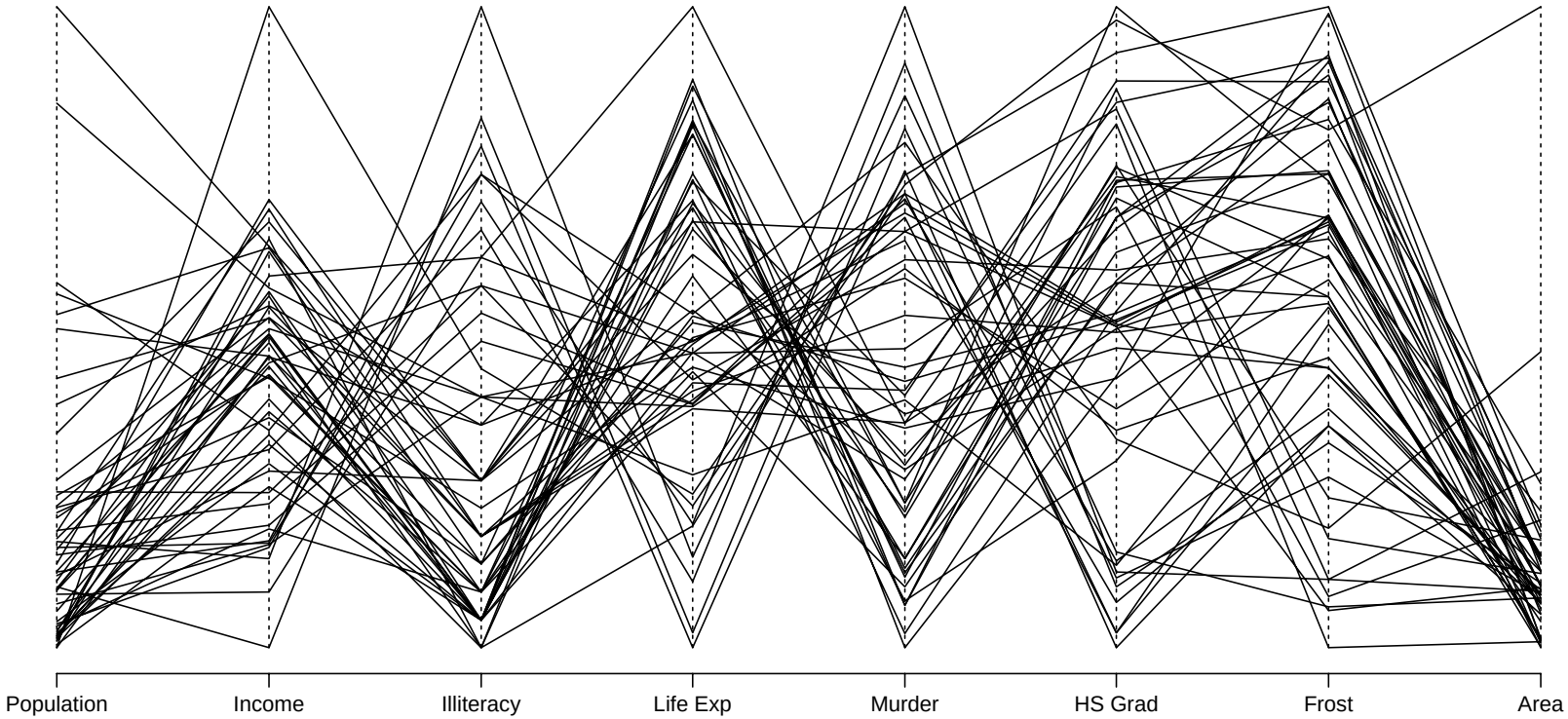
Oklahoma

Tennessee

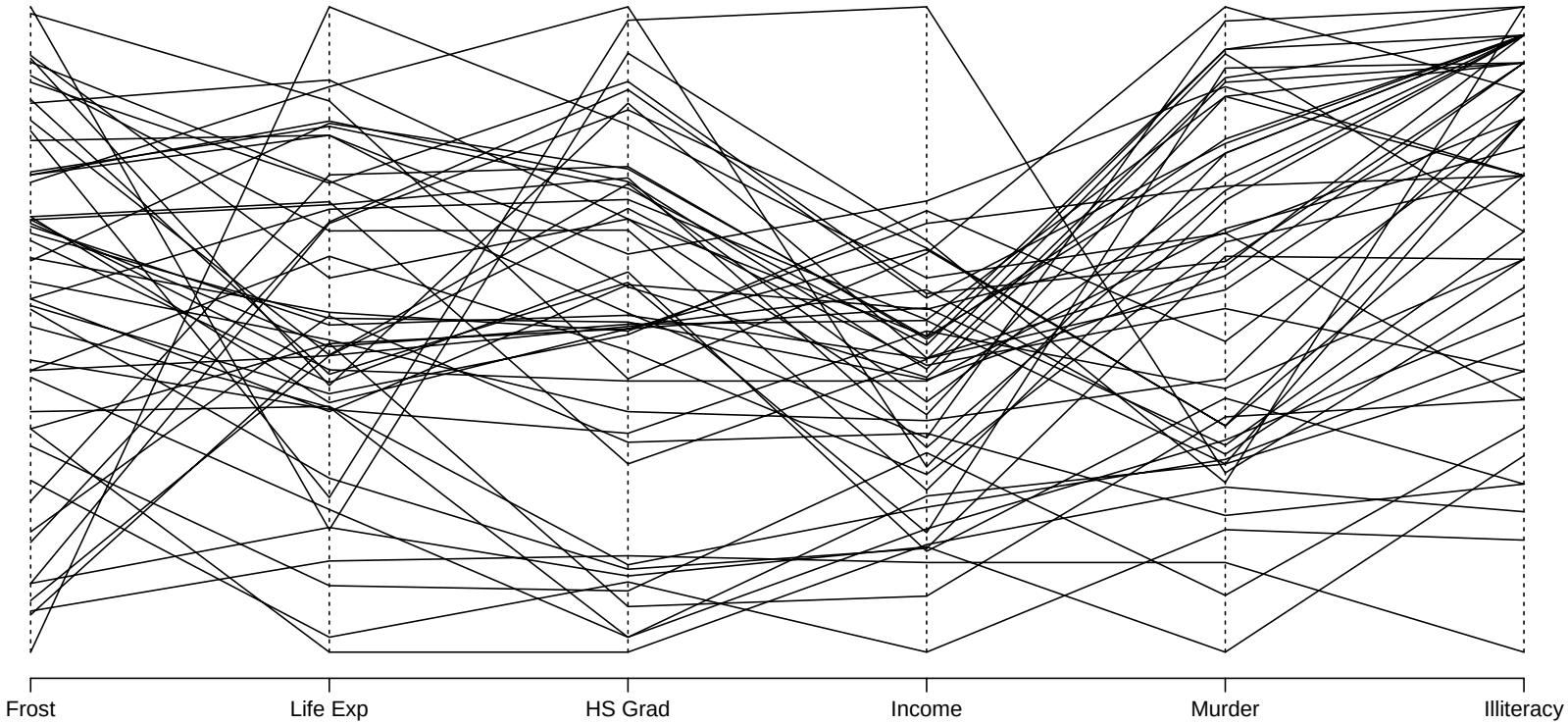
West Virginia



R version of stars plot of the `state.x77` dataset.



Parallel coordinate plot of the `state.x77` dataset.



A better parallel coordinate plot of the `state.x77` dataset.

Leptograpsus variegatus Crabs

200 crabs from Western Australia. Two colour forms, blue and orange; collected 50 of each form of each sex. Are the colour forms species?

Measurements of carapace (shell) length `CL` and width `CW`, the size of the frontal lobe `FL`, rear width `RW` and body depth `BD`.

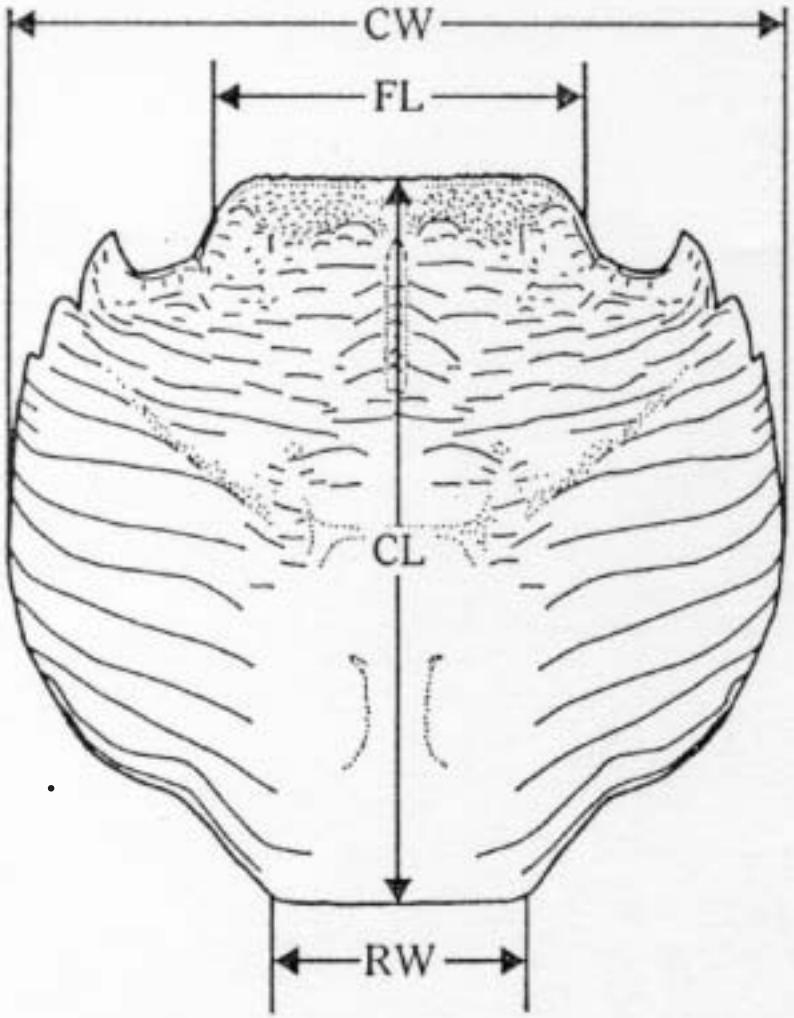
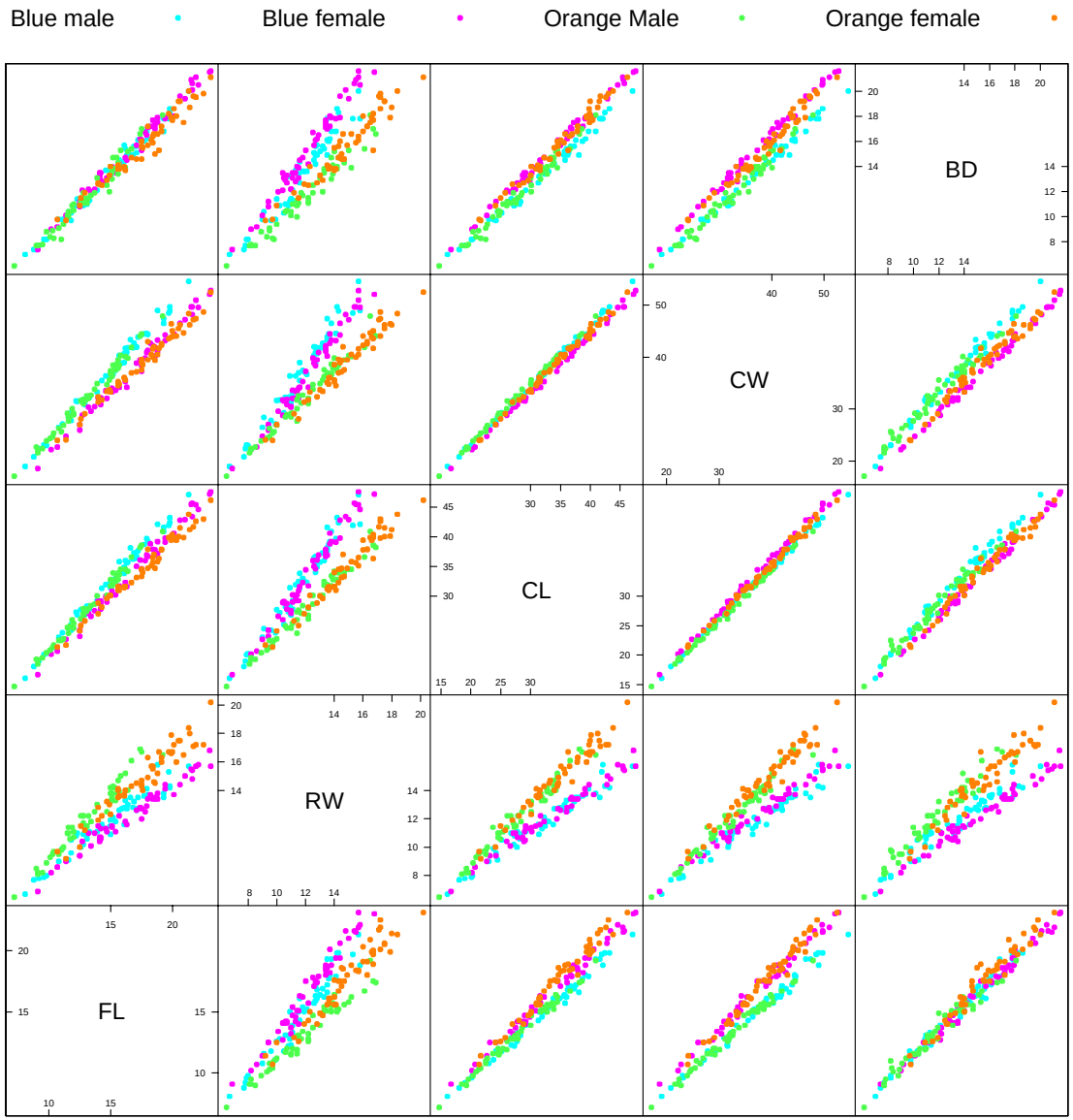
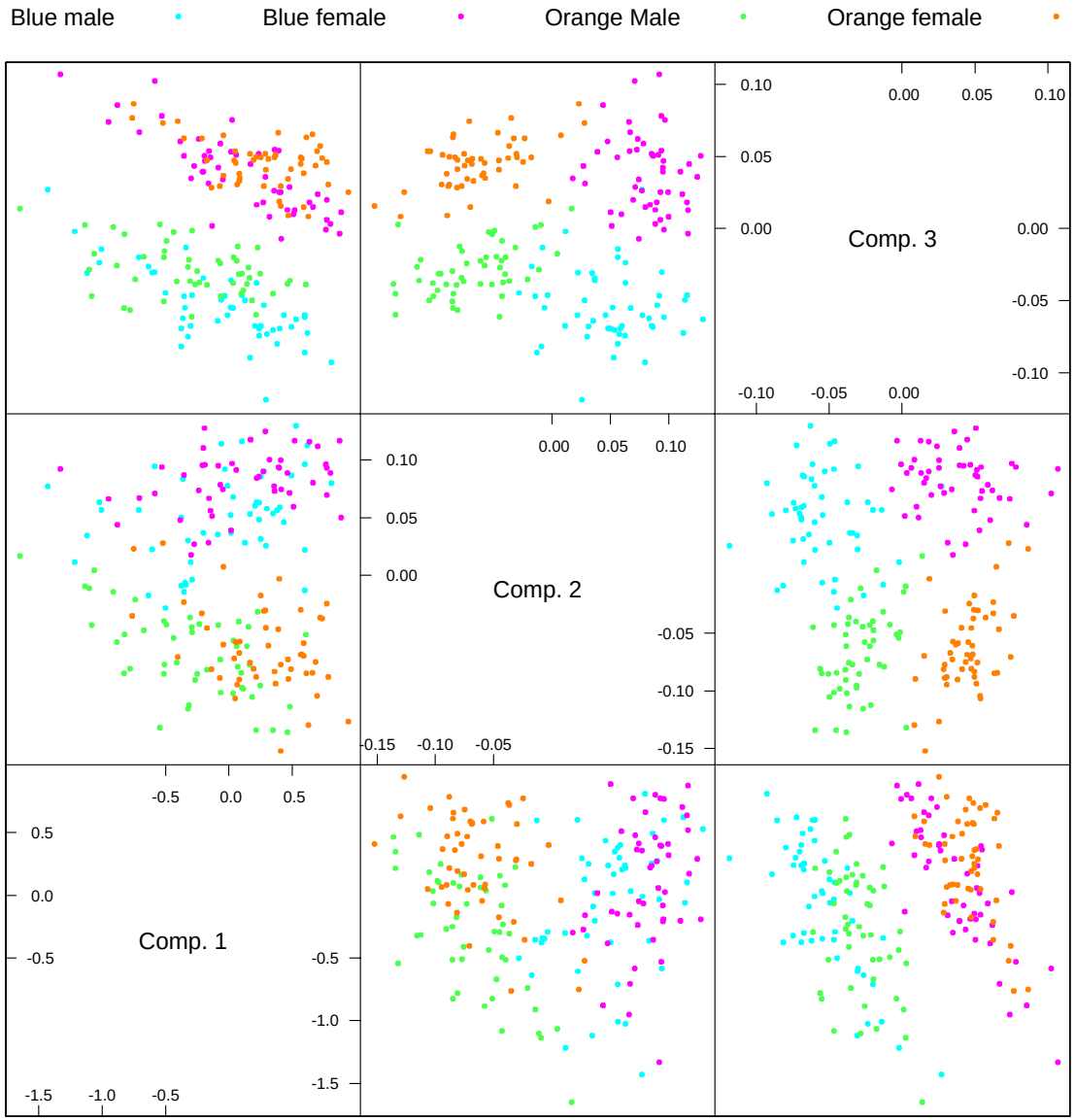
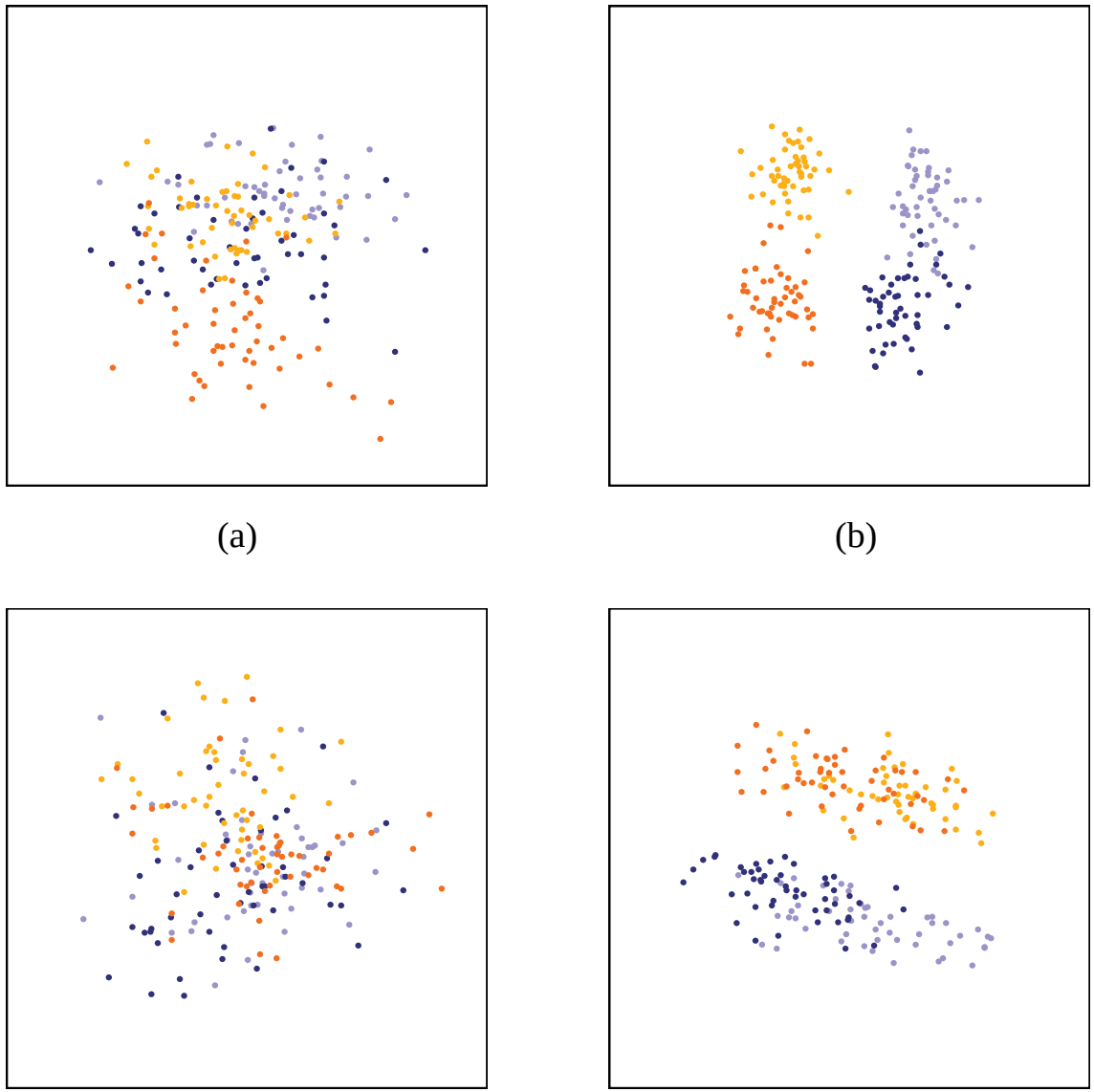


Fig. 1. Dorsal view of carapace of *Leptograpsus*, showing measurements taken. *FL*, width of frontal region just anterior to frontal tubercles. *RW*, width of posterior region. *CL*, length along midline. *CW*, maximum width. The body depth was also measured; in females but not in males the abdomen was first displaced.


Blue male
Blue female
Orange Male
Orange female
BD
CW
CL
RW
FL



First three principal components on log scale.



Projections of the *Leptograpsus* crabs data found by projection pursuit. View (a) is a random projection. View (b) was found using the natural Hermite index, view (c) by the Friedman–Tukey index and view (d) by Friedman's (1987) index.

Independent Components Analysis

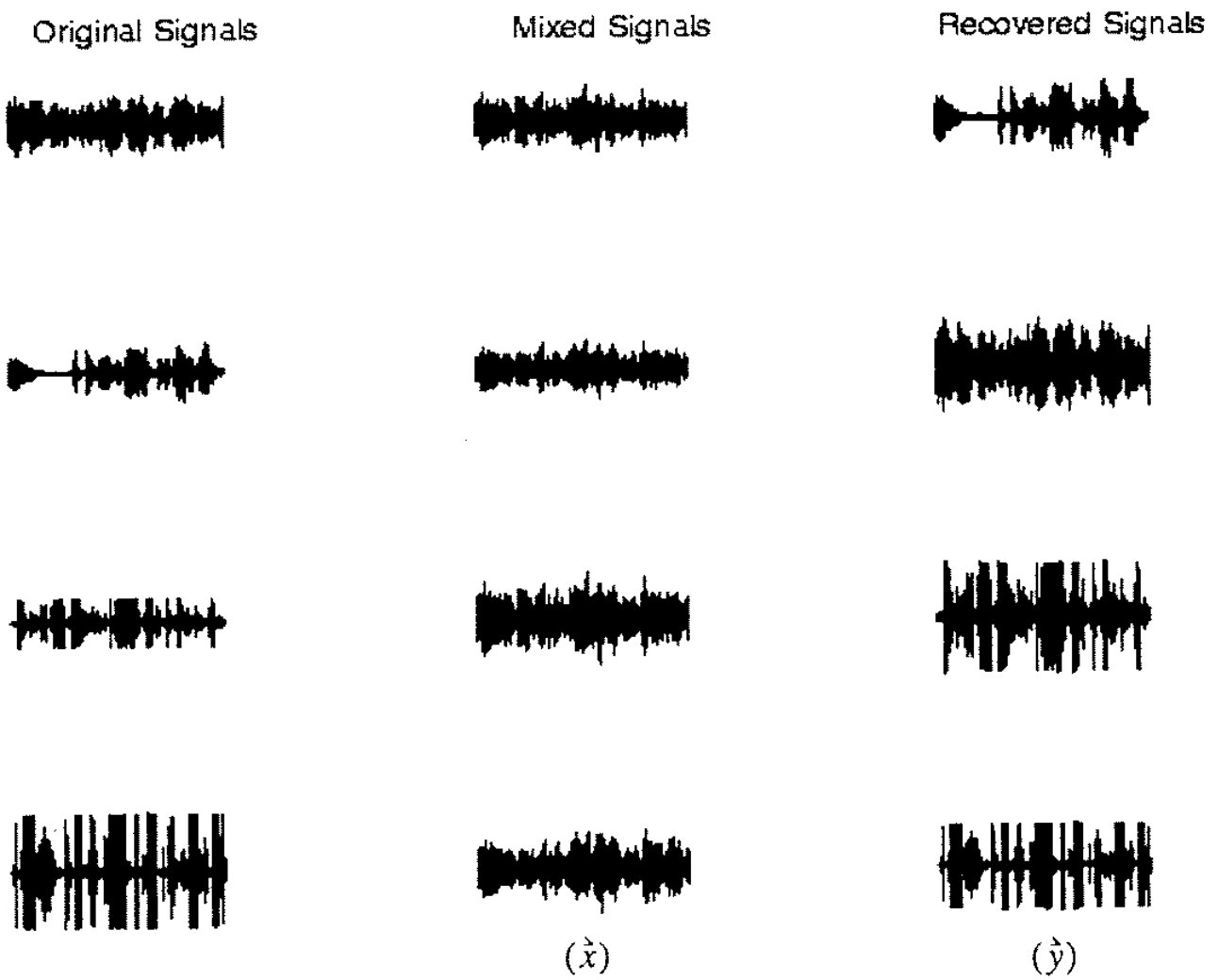
A ‘hot topic’ that has moved from field to field over the last decade. Originally(?) used for blind source signal separation in geophysics.

A projection pursuit technique in which the objective is to find k independent linear combinations. So minimize entropy difference between joint kD projection distributions and the product of their marginals.

Many local minima. No guarantee that you will find k signals not k noise sources. Choice of k may be crucial.

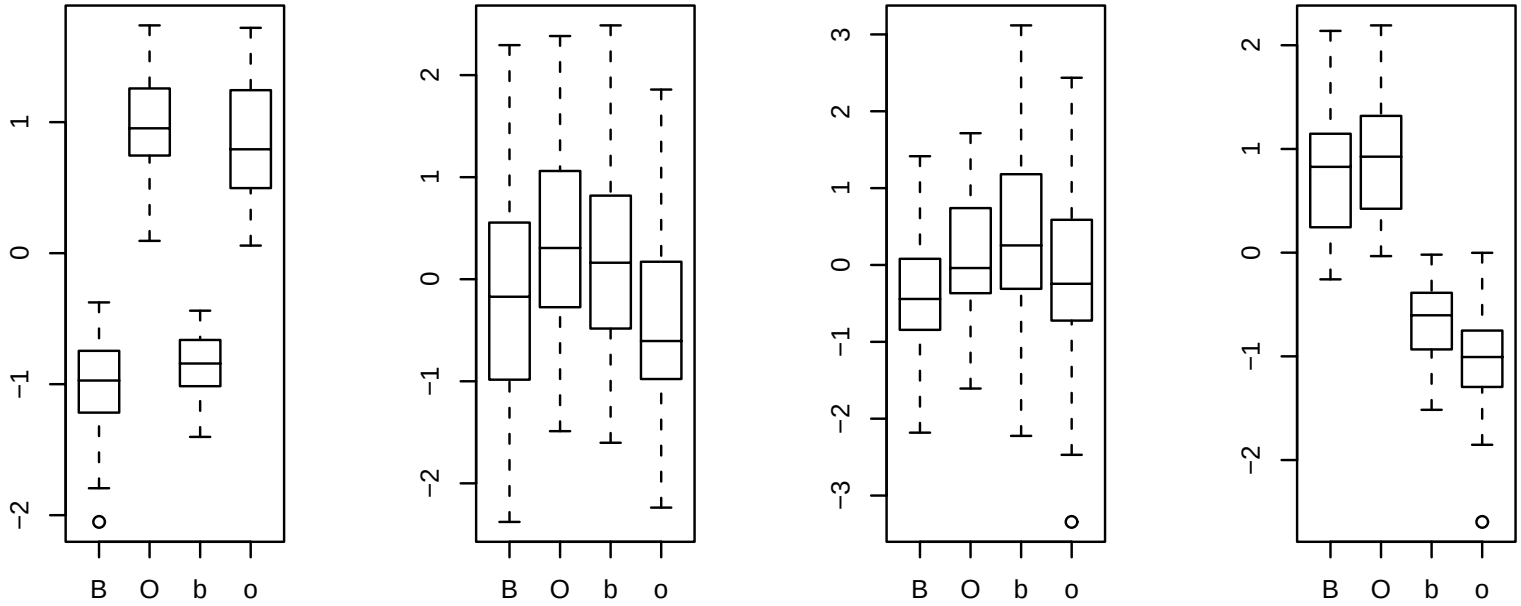
Many impressive results: but often every other visualization technique finds them. ‘*In the land of the blind . . . .*’

A close relative of *factor analysis* and other latent variable methods.



ICA experiment, from Deco & Obradovic (1996).

ICA for the Crabs data



Boxplots of four ‘signals’ recovered by ICA from the `crabs` data.

There is a lot of arbitrariness in the use of ICA, in particular in choosing the number of signals. We might have expected to need two here, when the results are much less impressive.

Multidimensional Scaling

Aim is to represent distances between points well.

Suppose we have distances (d_{ij}) between all pairs of n points, or a *dissimilarity* matrix. Classical MDS plots the first k principal components, and minimizes

$$\sum_{i \neq j} d_{ij}^2 - \widetilde{d}_{ij}^2$$

where $(\widetilde{d}_{ij})$ are the Euclidean distances in the kD space.

More interested in getting small distances right. Sammon (1969) proposed

$$\min E(d, \widetilde{d}) = \frac{1}{\sum_{i \neq j} d_{ij}} \sum_{i \neq j} \frac{(d_{ij} - \widetilde{d}_{ij})^2}{d_{ij}}$$

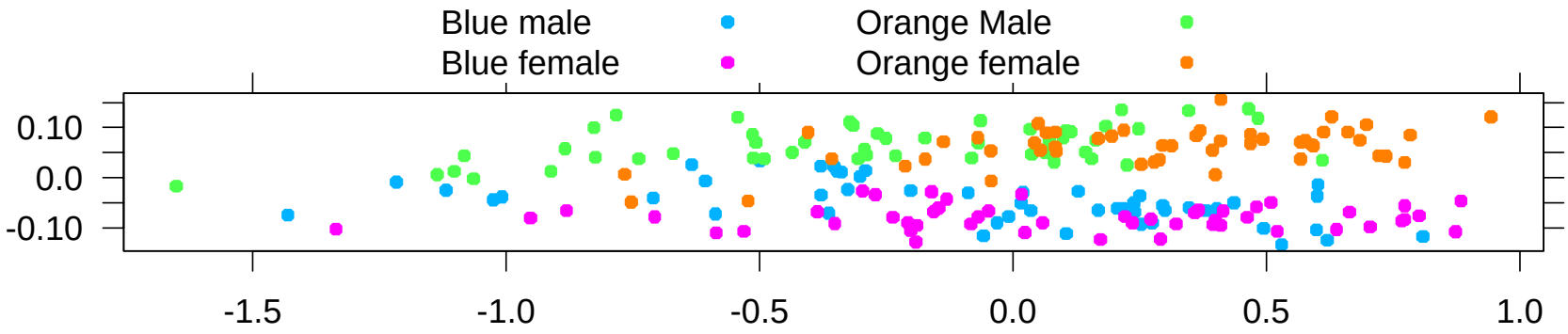
Shepard and Kruskal (1962–4) proposed only to preserve the ordering of distances, minimizing

$$STRESS^2 = \frac{\sum_{i \neq j} \left[\theta(d_{ij}) - \widetilde{d}_{ij}\right]^2}{\sum_{i \neq j} \widetilde{d}_{ij}^2}$$

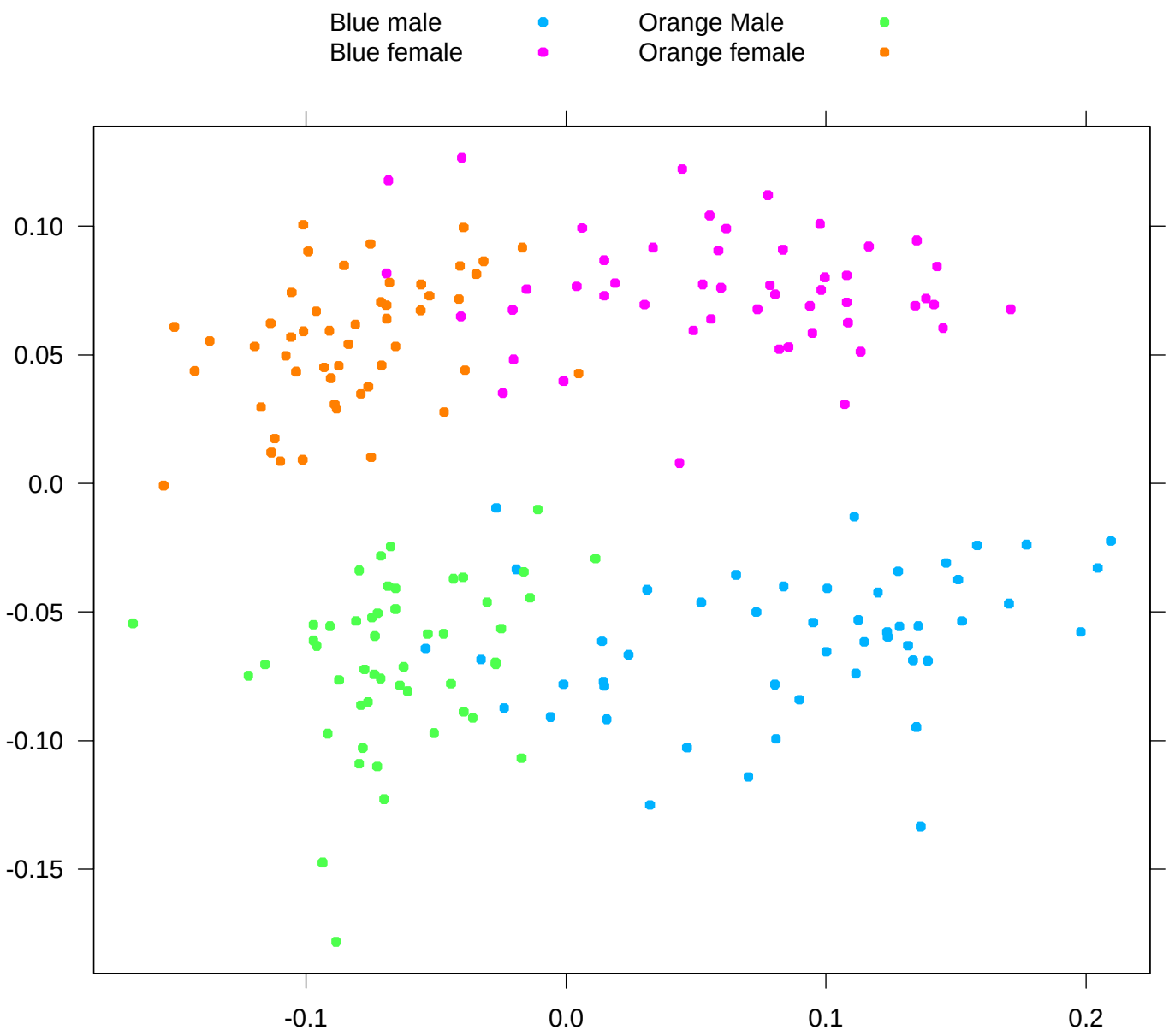
over both the configuration of points and an increasing function θ.

The optimization task is quite difficult and this can be slow.

Multidimensional scaling



An order-preserving MDS plot of the (raw) crabs data.

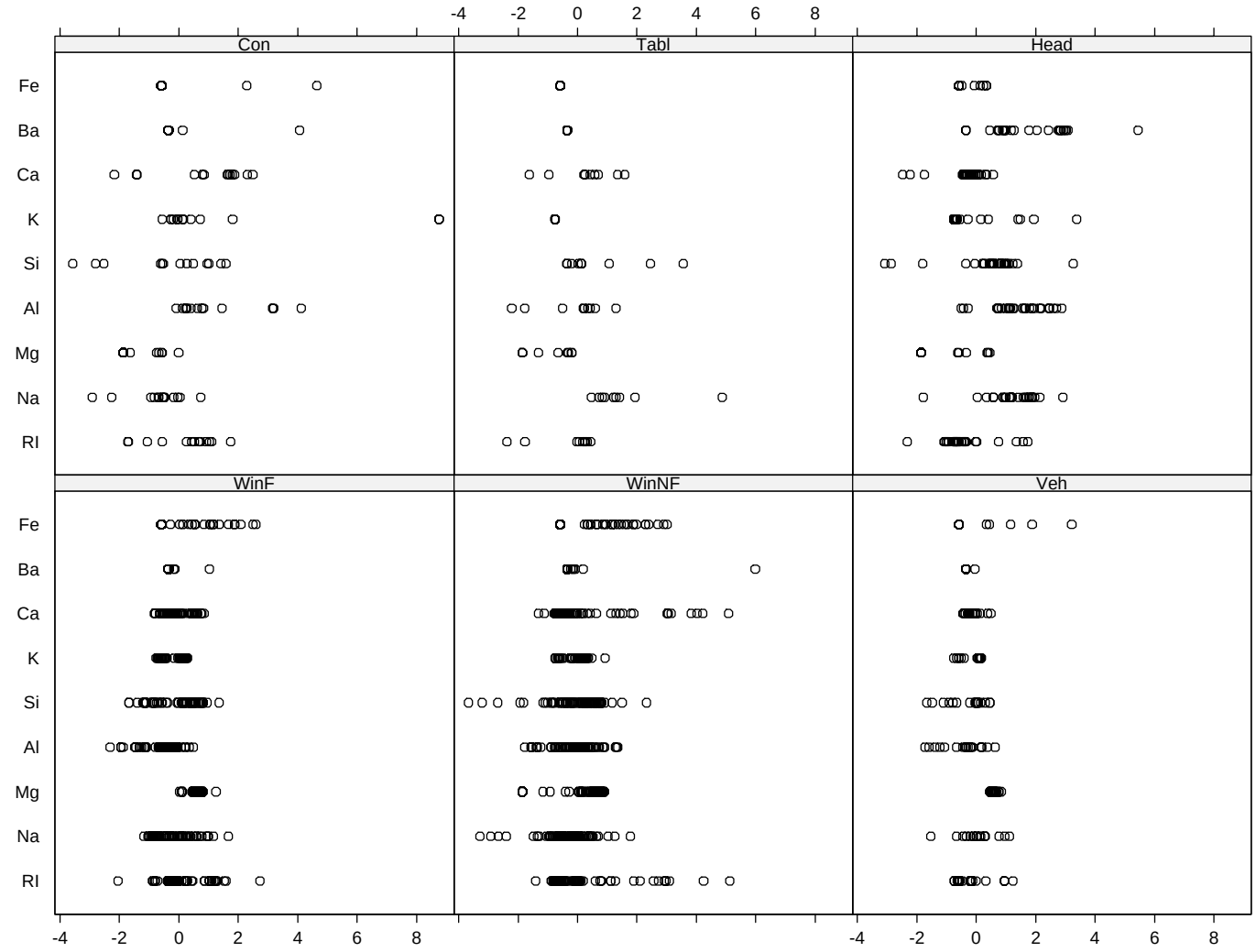


After re-scaling to (approximately) constant carapace area.

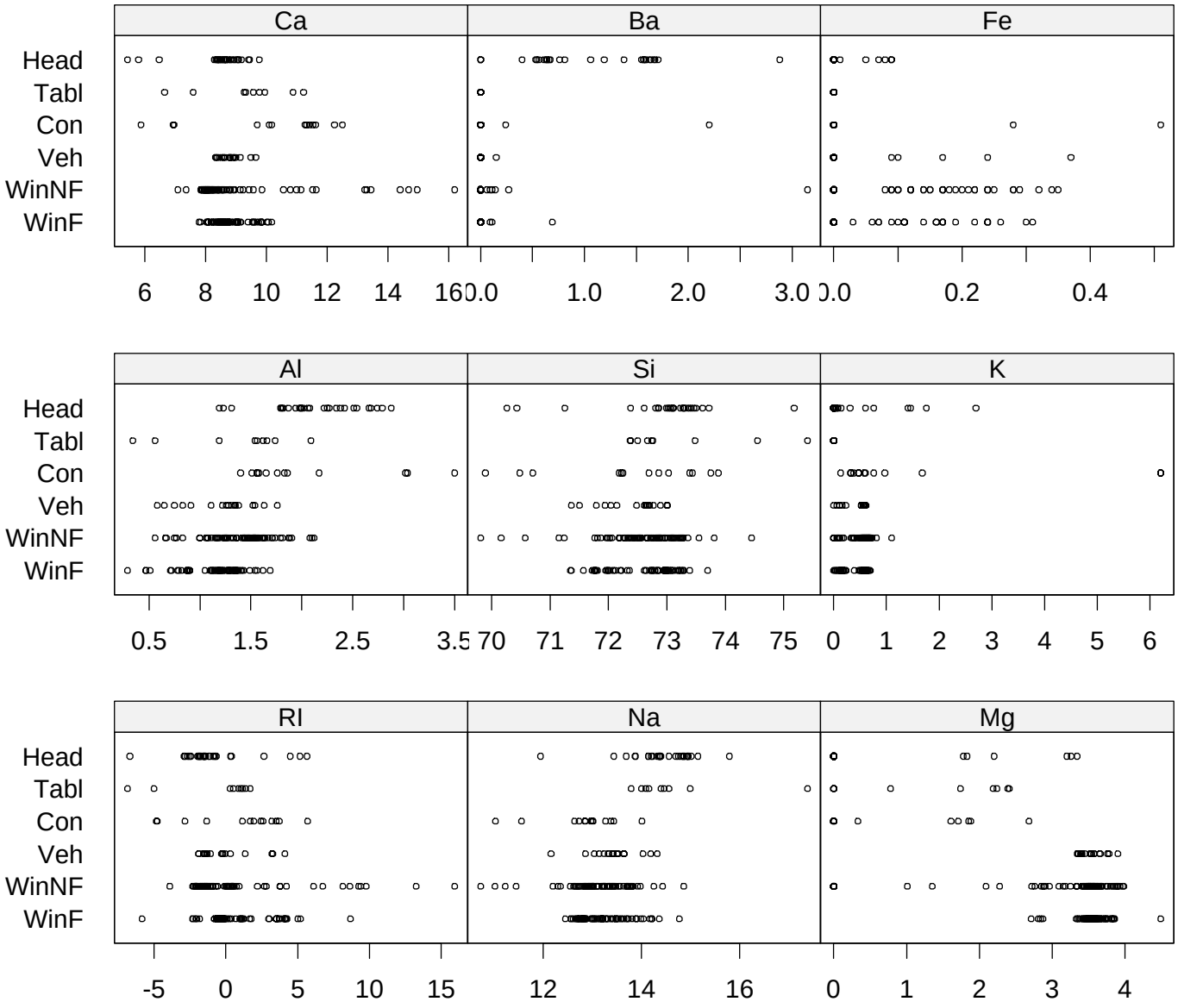
A Forensic Example

Data on 214 fragments of glass collected at scenes of crimes. Each has a measured refractive index and composition (weight percent of oxides of Na, Mg, Al, Si, K, Ca, Ba and Fe).

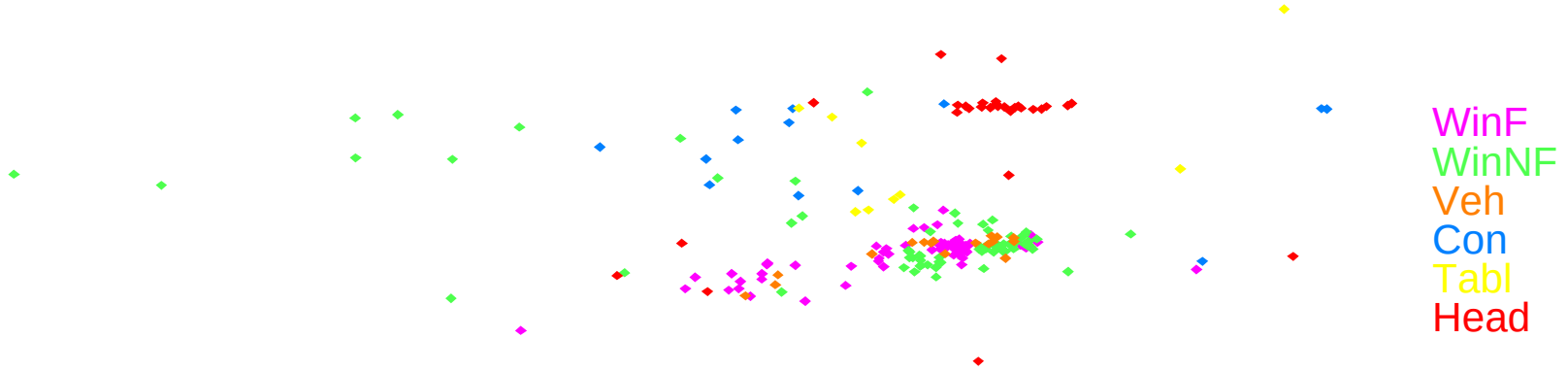
Grouped as window float glass (70), window non-float glass (76), vehicle window glass (17) and other (containers, tableware, headlamps) (22).



Strip plot by type of glass.



Strip plot by type of analyte.



Isotonic multidimensional scaling representation.

Kohonen's Self-Organizing Maps

Kohonen describes his own motivation as:

> 'I just wanted an algorithm that would effectively map similar patterns (pattern vectors close to each other in the input signal space) onto contiguous locations in the output space.'
>
> Kohonen (1995, p. VI)

He interpreted 'contiguous' via a rectangular or hexagonal 2-D lattice.

In K-*means clustering* the data are split into K groups, and each example is assigned to the cluster whose representative $\boldsymbol{m}_j$ is nearest to the example. The cluster representatives ('centre') are then adjusted to be the centroid of the group, and iteration gives a simple, finite, algorithm.

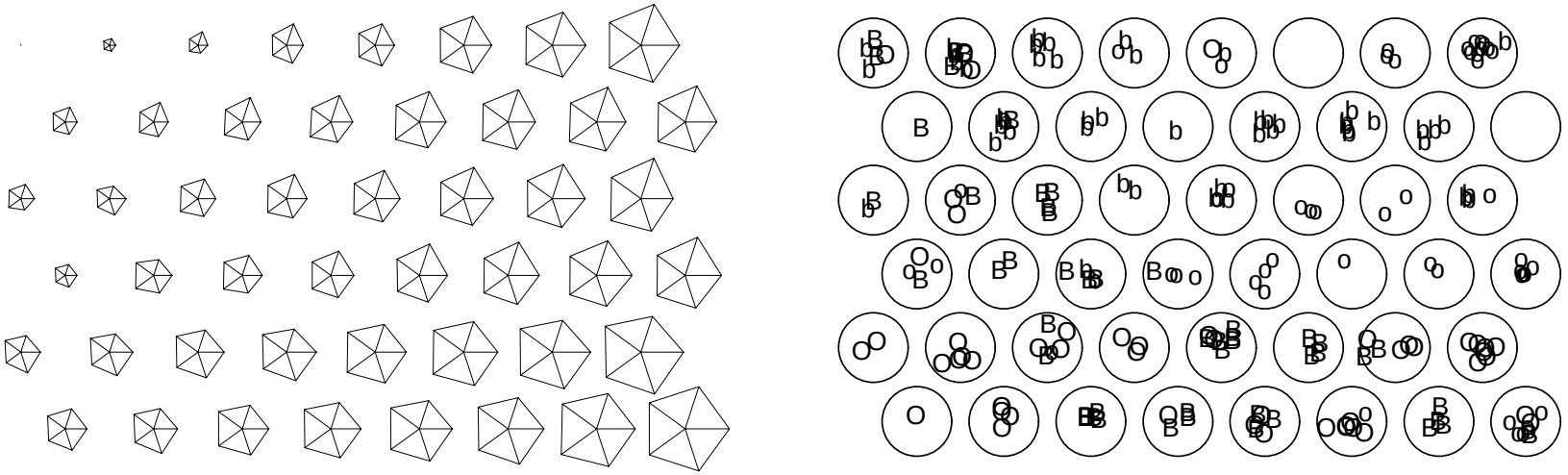
In SOM (self-organizing mapping) the representatives $(\boldsymbol{m}_j)$ are arranged on a regular grid, with representatives at nearby points on the grid are more similar that those which are widely separated.

Examples are presented in turn until convergence. The $\boldsymbol{m}_j$ are initially assigned at random. Whenever an example $\boldsymbol{x}$ is presented, the closest representative $\boldsymbol{m}_j$ is found. Then

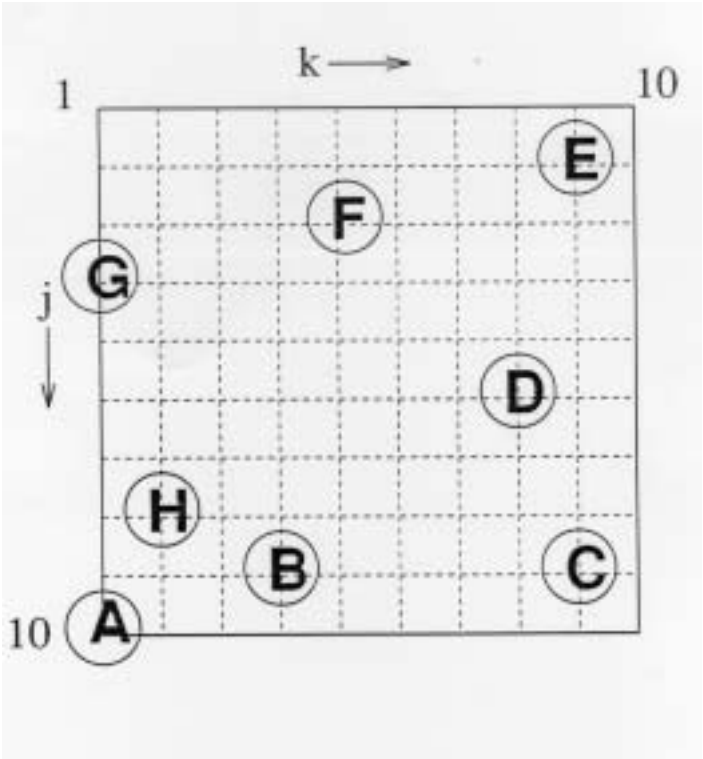
$$\boldsymbol{m}_i \leftarrow \boldsymbol{m}_i + \alpha[\boldsymbol{x} - \boldsymbol{m}_i] \qquad \text{for all neighbours } i \text{ .}$$

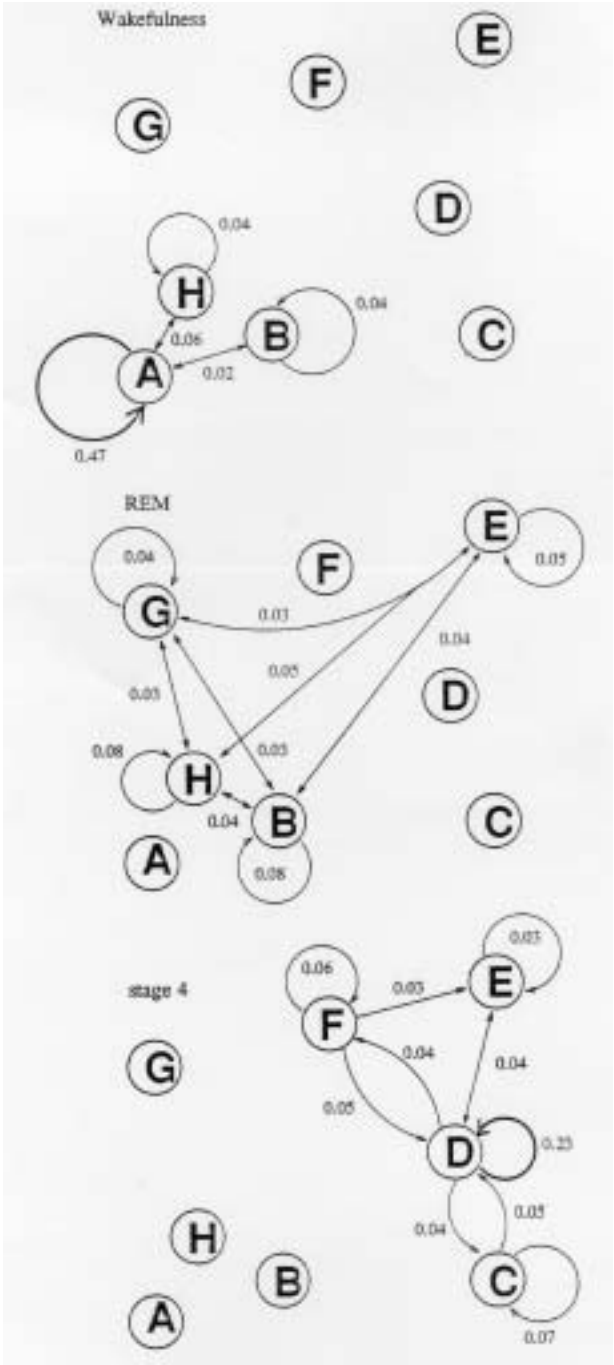
Both the constant α and the definition of 'neighbour' change with time.

A cruder form of MDS, but one that scales to 100,000+ examples.



SOM mapping of the crabs data to a 8×6 hexagonal grid. The left panel is a stars plot of the representatives. In the right panel the labels of those examples mapped to each cluster are distributed randomly within the circle representing the cluster.


k
1
10
j
10
E
F
G
D
H
B
C
A


Wakefulness
E
F
G
D
C
H
B
A
0.04
0.04
0.06
0.02
0.47
REM
E
F
G
D
H
B
C
A
0.04
0.05
0.03
0.04
0.05
0.05
0.08
0.05
0.04
0.08
stage 4
E
F
G
D
H
B
C
A
0.03
0.06
0.03
0.04
0.04
0.05
0.25
0.05
0.04
0.07

Clustering

General idea is to divide data into groups such that the points within a group are more similar to each other than to those in other groups.

Important details:

- The number k of groups may or may not be known.
- May wish to allow 'outliers' assigned to no group.
- Could allow overlap in group membership ('fuzzy clustering').

Note that need a measure of (dis)similarity or distance between a point and a group of points.

A Clustering of Cluster Methods

- Agglomerative hierarchical methods.
  - Produces a set of clusterings, usually one for each $k = n, \ldots, 2$.
  - Main differences are in calculating group–group dissimilarities from point–point dissimilarities.
  - Computationally easy.
- Optimal partitioning methods.
  - Produces a clustering for fixed K.
  - Need an initial clustering.
  - Lots of criteria to optimize, some based on (joint normal) probability models.
  - Can have distinct 'outlier' group(s).
- Divisive hierarchical methods.
  - Produces a set of clusterings, usually one for each $k = 2, \ldots, K \ll n$.
  - Computationally nigh-impossible.
  - Most available methods are *monothetic* (split on one variable at each stage).

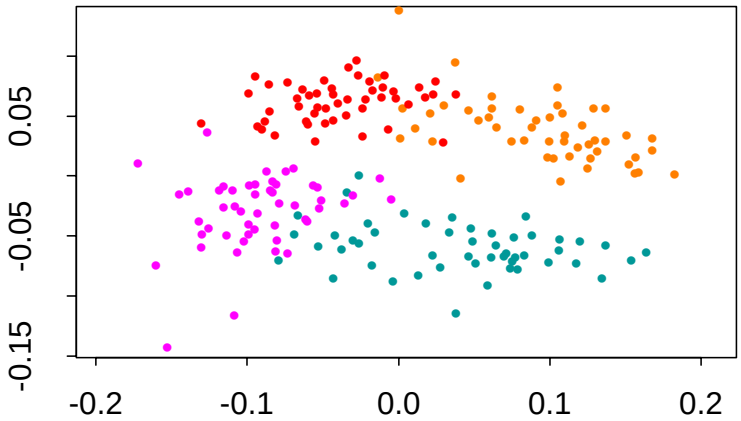
References

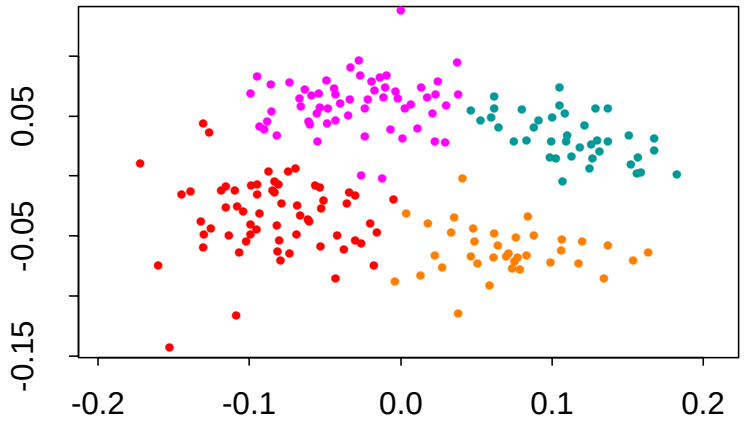
Comprehensive reference:
Gordon, A. D. (1999) *Classification.* Second Edition.

Good introduction:
Kaufman, L. and Rousseeuw, P. J. (1990) *Finding Groups in Data.*

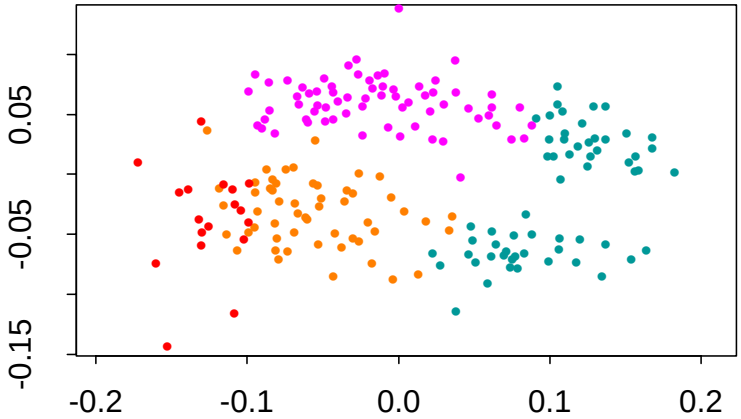
An example

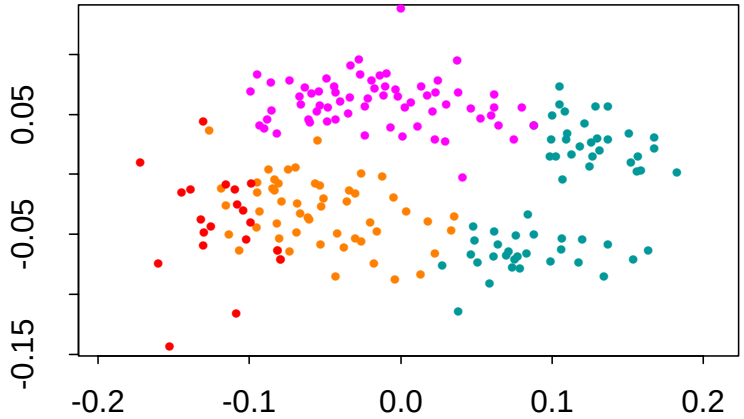
The *Leptograpsus* crabs data, with 4 groups (known in advance here). Same information as available to projection pursuit and MDS.





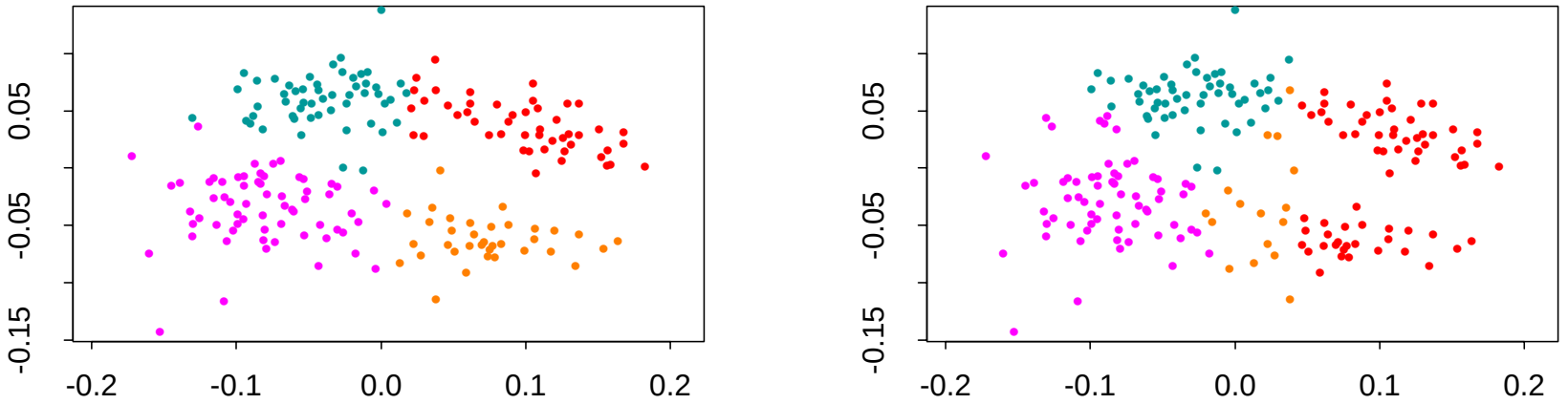
Left: True. Right: $(K = 4)$–means.





Left: complete-link hierarchical clustering.

Right: ‘maximum likelihood’ clustering with ellipsoidal clusters.



Left: Macnaughton-Smith *et al.*'s divisive method.

Right: 'hardened' classification from fuzzy clustering.