Multidimensional Scaling

- Have distances between individuals.
- Want to draw a picture (map) in 2 dimensions showing individuals so that distances (or order of distances) as close together as possible.
- If want to preserve actual distances, called *metric multidimensional scaling* (in SAS, `level=absolute`)
- If only want to preserve order of distances, called *non-metric multidimensional scaling* (in SAS, `level=ordinal`).
- Metric scaling has solution that can be worked out exactly.
- Non-metric only has iterative solution.
- Assess quality of fit via quantity “stress”, whether use of resulting map is reasonable. (Try something obviously 3-dimensional and assess its failure.) Stress has min 0 and max 1.

Metric scaling: European cities

The file `europe.dat` contains road distances (in km) between 16 European cities. Can we reproduce a map of Europe from these distances?

First, reading in the data (as TYPE=DISTANCE):

```
data euro(type=distance);
  infile "europe.dat" delimiter=",";
  input city $ Amsterdam Athens Barcelona Berlin
    Cologne Copenhagen Edinburgh Geneva London
    Madrid Marseille Munich Paris Prague Rome Vienna
```

- Values in spreadsheet.
- Save as .csv.
- Take out quotes.
- Values separated by commas, suitable for reading by SAS.

The code, using PROC MDS

```
proc mds level=absolute out=y outres=z;
proc print data=y;
proc sort data=z;
  by residual;
proc print data=z;
  var _row_ _col_ residual;
proc plot data=y;
  plot dim1 * dim2 $ _label_;
```

- Run PROC MDS using `level=absolute` to reproduce the exact distances (to scale).
- Two output data sets: one containing the coordinates for our map, and one containing the observed and predicted (from map) distances and residuals.
- Print coordinates.
- Sort residuals and print them (with the cities they belong to).
- Plot coordinates, labelling each point by its city.

The coordinates

In Dim1 and Dim2:

			TYPE	_LABEL_	_NAME_	Dim1	Dim2
1	2	.	CRITERION			0.07	.
2	2	.	CONFIG	Amsterdam	Amsterdam	-300.71	558.62
3	2	.	CONFIG	Athens	Athens	2599.74	-375.74
4	2	.	CONFIG	Barcelona	Barcelona	-704.34	-1012.29
5	2	.	CONFIG	Berlin	Berlin	402.29	619.72
6	2	.	CONFIG	Cologne	Cologne	-83.70	396.98
7	2	.	CONFIG	Copenhagen	Copenhagen	97.17	1241.96
8	2	.	CONFIG	Edinburgh	Edinburgh	-1232.60	906.77
9	2	.	CONFIG	Geneva	Geneva	-185.99	-342.22
10	2	.	CONFIG	London	London	-574.43	406.08
11	2	.	CONFIG	Madrid	Madrid	-1341.37	-1088.16
12	2	.	CONFIG	Marseille	Marseille	-319.76	-750.10
13	2	.	CONFIG	Munich	Munich	326.13	-25.17
14	2	.	CONFIG	Paris	Paris	-525.60	49.92
15	2	.	CONFIG	Prague	Prague	541.20	285.90
16	2	.	CONFIG	Rome	Rome	541.38	-1031.08
17	2	.	CONFIG	Vienna	Vienna	760.58	158.80

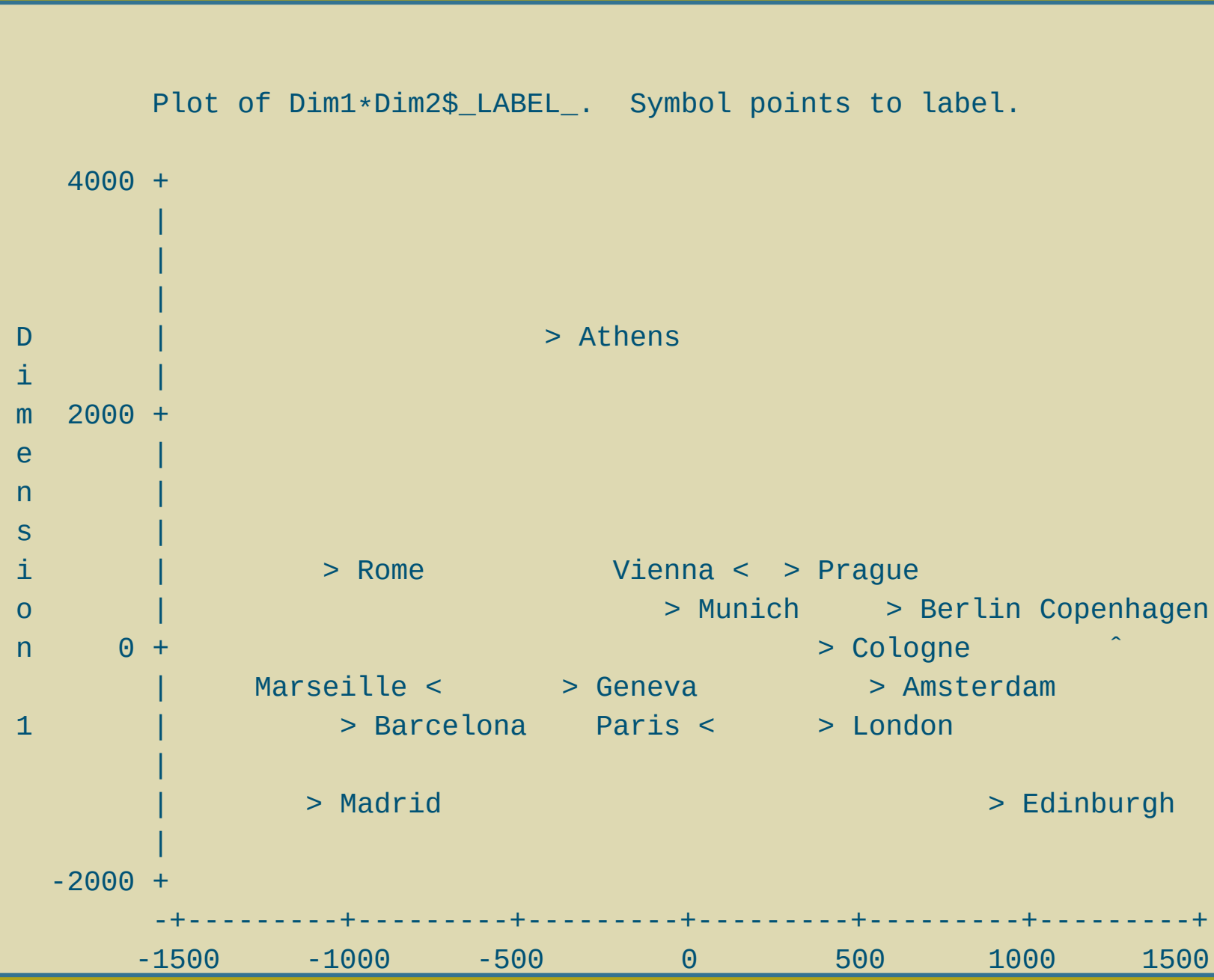
Stress 0.07 is small.

The sorted residuals (edited)

Obs	_ROW_	_COL_	RESIDUAL
1	Vienna	London	-445.723
2	Edinburgh	Athens	-273.247
3	Cologne	Athens	-230.477
4	London	Edinburgh	-170.966
5	Madrid	Cologne	-170.119
6	London	Athens	-170.038
...			
115	Rome	Madrid	215.393
116	Rome	Barcelona	225.139
117	Madrid	Edinburgh	374.108
118	Rome	Athens	390.827
119	Copenhagen	Athens	434.100
120	Edinburgh	Copenhagen	492.631

Edinburgh and Athens feature in a lot of the large residuals.

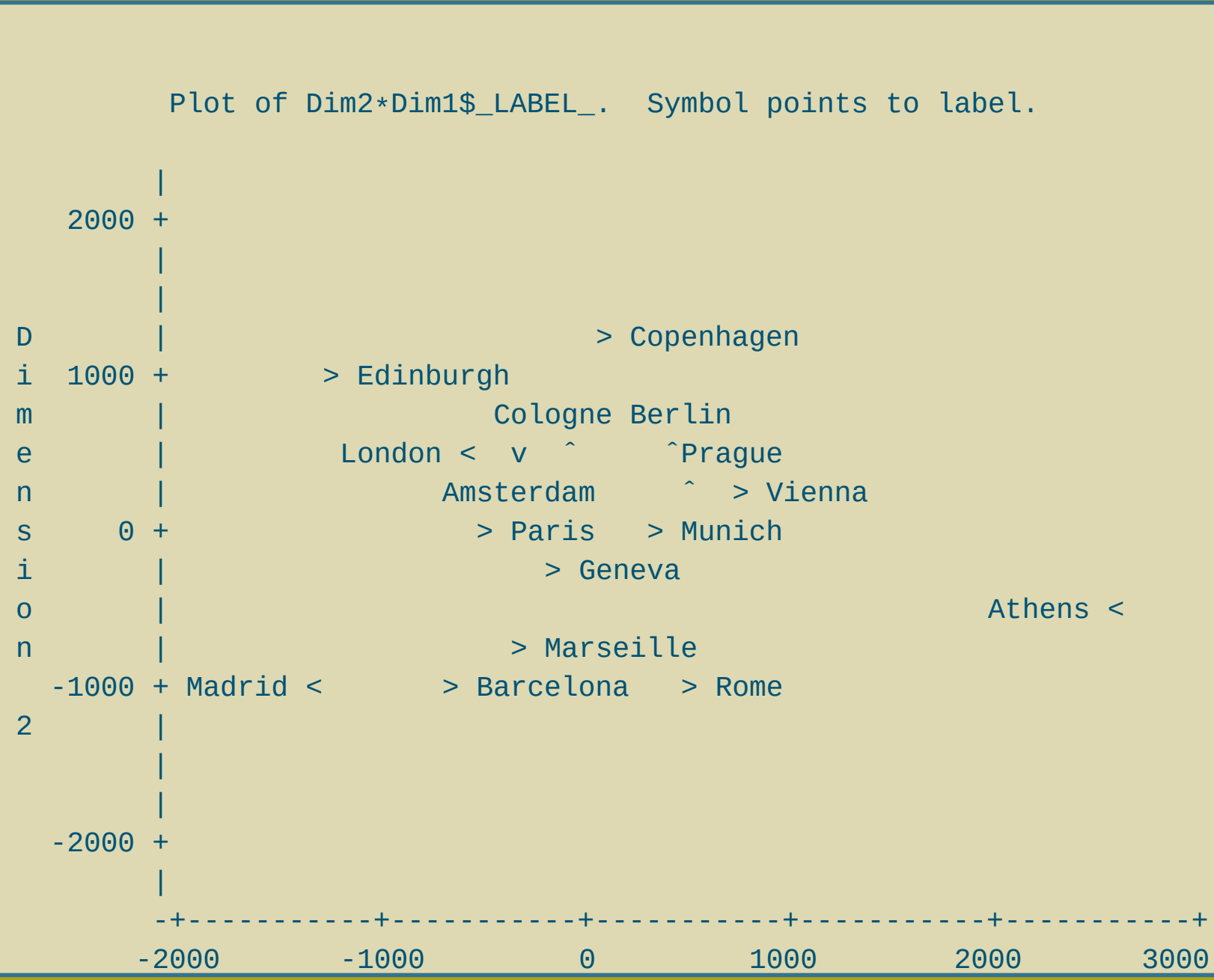
The map


Plot of Dim1*Dim2$_LABEL_. Symbol points to label.
4000
2000
0
-2000
Dimension 1
> Athens
> Rome
Vienna <
> Prague
> Munich
> Berlin
Copenhagen
> Cologne
Marseille <
> Geneva
> Amsterdam
> Barcelona
Paris <
> London
> Madrid
> Edinburgh
-1500
-1000
-500
0
500
1000
1500
Dimension 2

Comments on map

- The map looks upside down!
- MDS doesn’t know about directions, only distances, so map could come out reflected (vertically or horizontally) or rotated.
- Given all that, cities look in about right relative places.
- City pairs with largest positive residuals have large bodies of water between them (affecting road distance considerably):
  - Edinburgh–Copenhagen (North Sea)
  - Rome–Athens (Adriatic)
- As it happens, plotting `Dim2*Dim1` produces almost reasonable map:

Map 2


Plot of Dim2*Dim1$_LABEL_. Symbol points to label.
2000
1000
0
-1000
-2000
Dimension 2
> Copenhagen
> Edinburgh
Cologne Berlin
London < v ^ ^Prague
Amsterdam ^ > Vienna
> Paris > Munich
> Geneva
Athens <
> Marseille
Madrid < > Barcelona > Rome
-2000 -1000 0 1000 2000 3000
Dimension 1

Non-metric scaling: languages

- Recall language data (from cluster analysis): 1–10, measure dissimilarity between two languages by how many number names *differ* in first letter. Data:

```
en 0 2 2  7 6  6   6  6  7  9 9
no 2 0 1  5 4  6   6  6  7  8 9
dk 2 1 0  6 5  6   5  5  6  8 9
nl 7 5 6  0 5  9   9  9 10  8 9
de 6 4 5  5 0  7   7  7  8  9 9
fr 6 6 6  9 7  0   2  1  5 10 9
es 6 6 5  9 7  2   0  1  3 10 9
it 6 6 5  9 7  1   1  0  4 10 8
pl 7 7 6 10 8  5   3  4  0 10 9
hu 9 8 8  8 9  10 10 10 10  0 8
sf 9 9 9  9 9  9   9  8  9  8 0
```

- Only want to reproduce *order* of dissimilarities; actual numbers don't matter. (Map only reproduces *relative* closeness of languages.)

Code

- Read data as distances, use `level=ordinal`. Print coordinates and residuals, plot map (labelled by language):

```
data lang(type=distance);
  infile "one-ten.dat";
  input lang $ en no dk nl de fr es it pl hu sf;

proc mds level=ordinal out=coords outres=dist;
  id lang;

proc print data=dist;
  var _row_ _col_ data distance residual;

proc print data=coords;

proc plot data=coords;
  plot dim2 * dim1 = '*' $ lang;
```

Output from PROC MDS

Multidimensional Scaling: Data=WORK.LANG.DATA
Shape=TRIANGLE Condition=MATRIX Level=ORDINAL
Coef=IDENTITY Dimension=2 Formula=1 Fit=1
Mconverge=0.01 Gconverge=0.01 Maxiter=100 Over=2 Ridge=0.0001

Iteration	Type	Badness-of-Fit Criterion	Change in Criterion	Convergence Measures: Monotone	Convergence Measures: Gradient
0	Initial	0.2009	.	.	.
1	Monotone	0.1478	0.0531	0.1358	0.6781
2	Gau-New	0.1126	0.0352	.	.
3	Monotone	0.1020	0.0105	0.0483	0.3363
4	Gau-New	0.0997	0.002376	.	.
5	Monotone	0.0928	0.006869	0.0374	0.2226
6	Gau-New	0.0923	0.000483	.	.
7	Monotone	0.0915	0.000823	0.0138	0.2190
8	Gau-New	0.0914	0.0000983	.	.
9	Monotone	0.0910	0.000349	0.009497	0.2341
10	Gau-New	0.0888	0.002191	.	0.0533
11	Gau-New	0.0887	0.000106	.	0.0169
12	Gau-New	0.0887	0.0000126	.	0.006850

Iterative procedure converges (stress stops getting smaller at 0.0887, which is small).

The residuals (selected)

Shown: pair of languages, dissimilarity, distance on map, residual (based on ordered data). Large residual means data and distance on map don't match.

Obs	_ROW_	_COL_	DATA	DISTANCE	RESIDUAL
7	de	en	6	0.81928	0.49528
55	sf	hu	8	2.00452	0.35904
49	sf	nl	9	3.15422	-0.34249
40	hu	nl	8	2.02361	0.33995
48	sf	dk	9	2.48611	0.32562
31	pl	dk	6	1.62422	-0.30966
6	nl	dk	6	1.61869	-0.30413
50	sf	de	9	3.10815	-0.29643
5	nl	no	5	1.31502	-0.27280
32	pl	nl	10	3.23354	0.24178
54	sf	pl	9	2.54350	0.26823

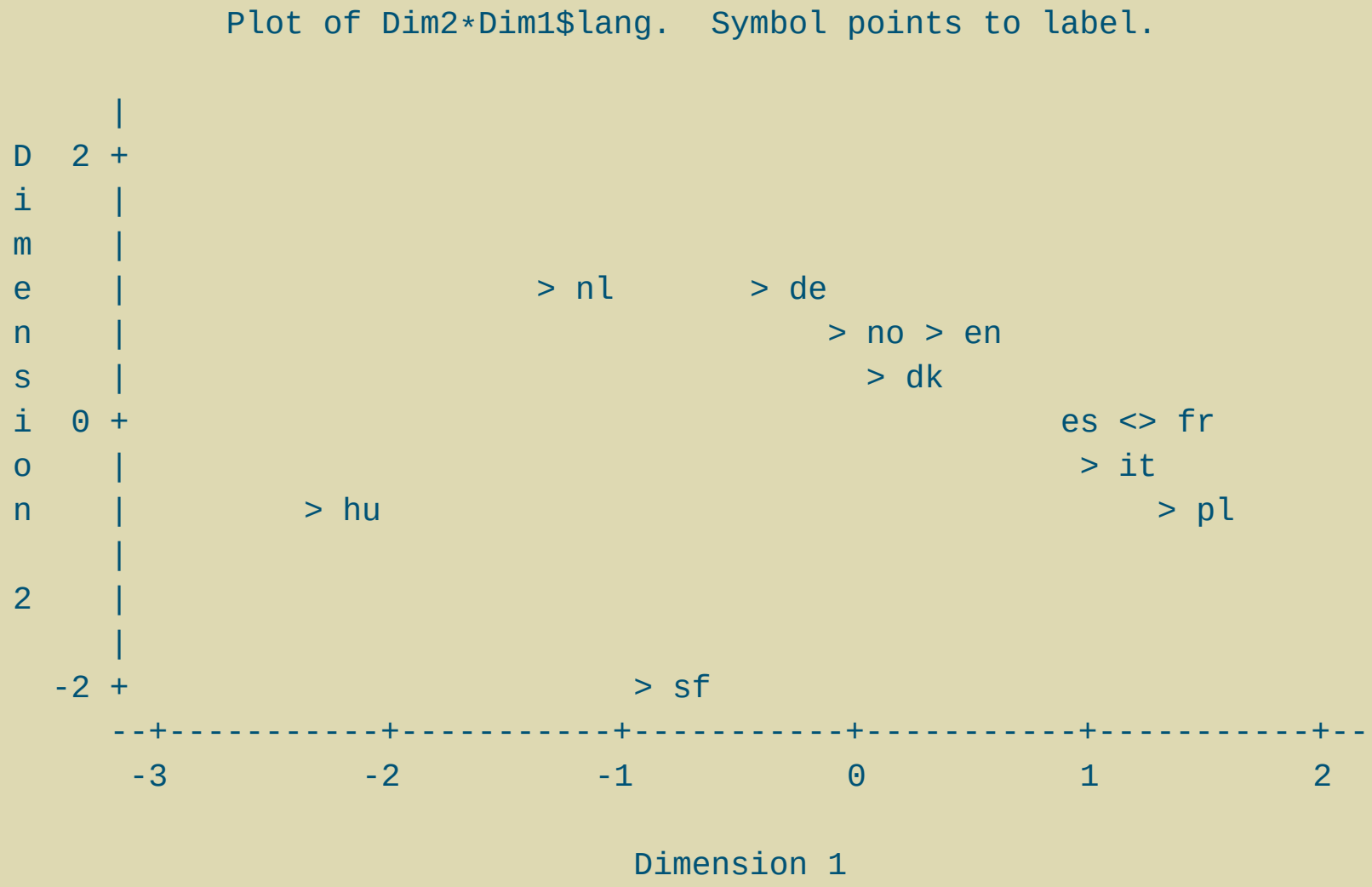
- Positive residual: actual dissimilarity greater than expected (compared to map)
- Negative residual: actual dissimilarity less than expected from map.

The coordinates

Obs	_DIMENS_	_MATRIX_	_TYPE_	lang	_NAME_	Dim1	Dim2
1	2	.	CRITERION			0.08872	.
2	2	.	CONFIG	en	en	0.30099	0.65225
3	2	.	CONFIG	no	no	-0.11417	0.58068
4	2	.	CONFIG	dk	dk	0.08220	0.30450
5	2	.	CONFIG	nl	nl	-1.30472	1.13912
6	2	.	CONFIG	de	de	-0.39587	1.08307
7	2	.	CONFIG	fr	fr	1.22529	0.07596
8	2	.	CONFIG	es	es	1.12900	-0.15541
9	2	.	CONFIG	it	it	0.96244	-0.35587
10	2	.	CONFIG	pl	pl	1.33098	-0.73409
11	2	.	CONFIG	hu	hu	-2.33345	-0.60349
12	2	.	CONFIG	sf	sf	-0.88268	-1.98673

- 1st row: stress value (max 1, min 0).
- CONFIG lines: Dim1 and Dim2 have coordinates.

The map


Plot of Dim2*Dim1$lang. Symbol points to label.
Dimension 2
2
0
-2
> nl
> de
> no
> en
> dk
es <> fr
> it
> hu
> pl
> sf
-3
-2
-1
0
1
2
Dimension 1

Comments on map

- See how distant Hungarian and Finnish are from each other, and the rest.
- See tight grouping of Italian, French and Spanish (Polish nearby).
- See looser grouping of Germanic languages at top (English, German, Dutch, Norwegian, Danish).

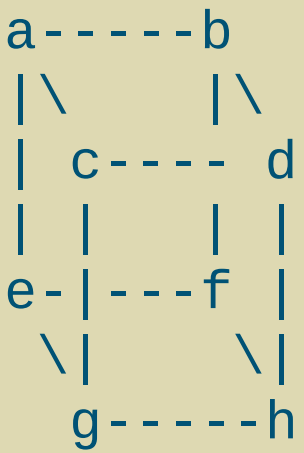
Guidelines for stress values

Smaller is better:

Stress value	Interpretation
Less than 0.05	Excellent: no prospect of misinterpretation (rarely achieved)
0.05–0.10	Good: most distances reproduced well, small prospect of false inferences
0.10–0.20	Fair: usable, but some distances misleading.
More than 0.20	Poor: may be dangerous to interpret

- Cities and languages examples both had stress in “good” range.

A cube



Cube has side length 1, so distance across diagonal on same face is $\sqrt{2} \simeq 1.4$ and “long” diagonal of cube is $\sqrt{3} \simeq 1.7$.

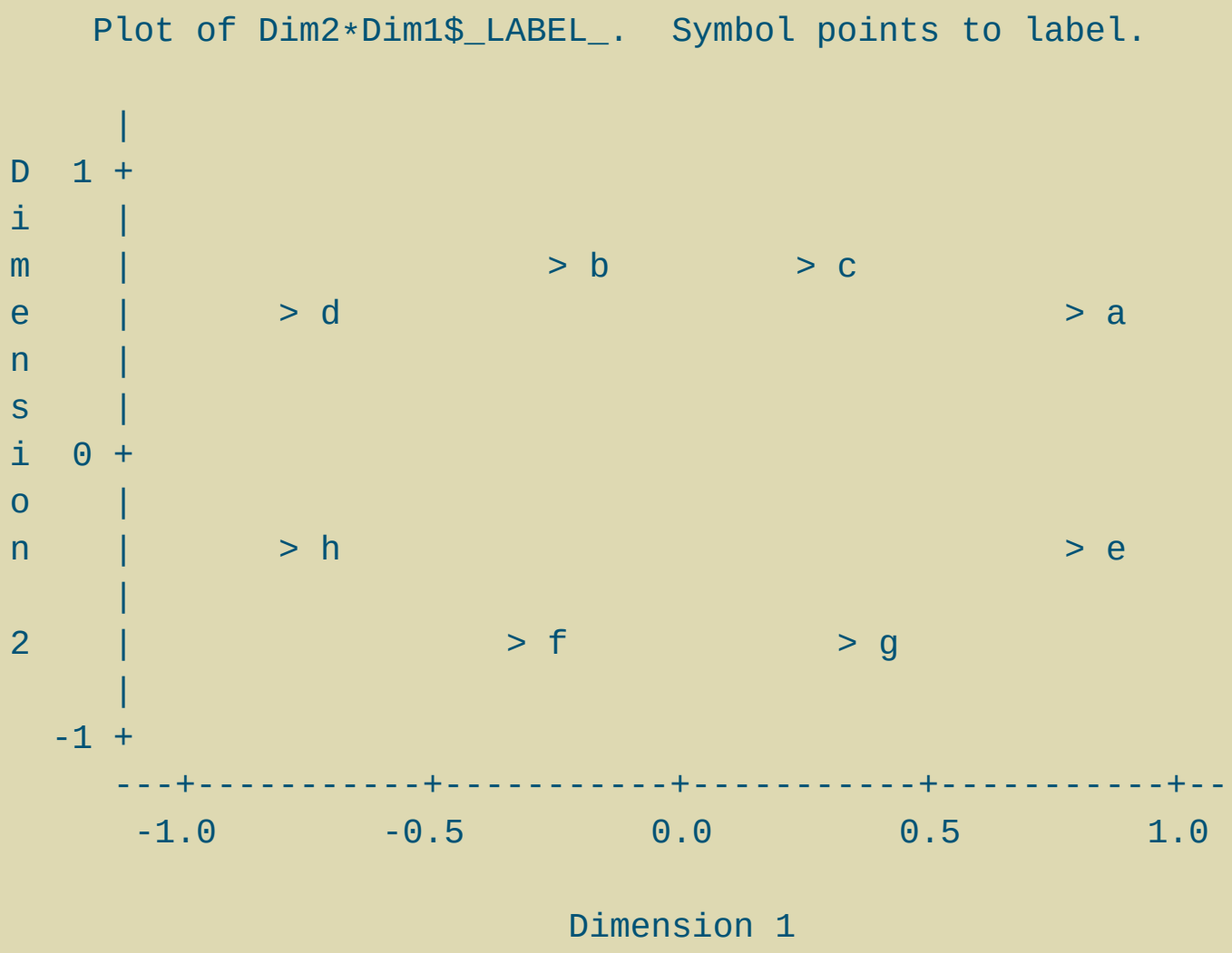
Try MDS on this obviously 3-dimensional data.

Converges OK

Iteration	Type	Badness-of-Fit Criterion	Change in Criterion	Convergence Measure
0	Initial	0.2987	.	0.6106
1	Lev-Mar	0.2275	0.0711	0.1308
2	Gau-New	0.2251	0.002446	0.0409
3	Gau-New	0.2248	0.000263	0.0164
4	Gau-New	0.2248	0.0000426	0.006667

but stress, at 0.2248, in “poor” range. Map probably won’t reproduce cube very well.

“Map” of cube


Plot of Dim2*Dim1$_LABEL_. Symbol points to label.
Dimension 2
1
0
-1
b
c
d
a
h
e
f
g
-1.0
-0.5
0.0
0.5
1.0
Dimension 1

Comments

- Map doesn't resemble cube.
- Some of the residuals are large: eg. `g` and `f`: actual distance is 1.4, map distance 0.7.
- Might have guessed this with stress in "poor" range.
- SAS lets you choose dimension of map. Use this PROC MDS line:
  ```
  proc mds dim=3 level=absolute outres=res2;
  ```
  (no point saving coordinates since we cannot plot them.)
- Resulting stress is 0.0342, "excellent".
- Largest residual (in size) is -0.1, most much smaller.
- Can't "squeeze" 3-D data into 2 dimensions!