Multidimensional Scaling with the smacof package

Barbara Jancewicz

OBM UW

barbara@jancewicz.net

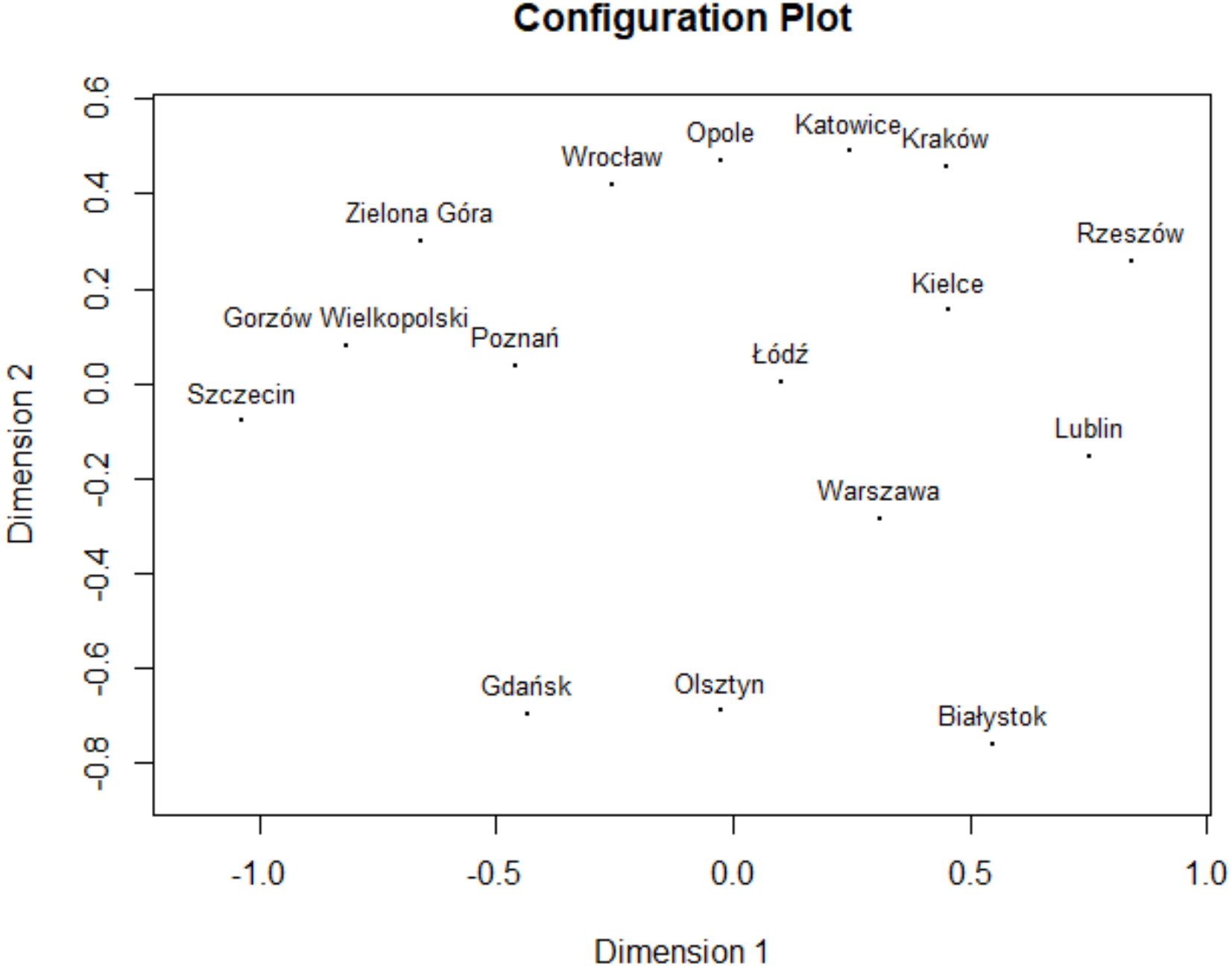
Multidimensional scaling (MDS) is often used to visualise data, to reduce the number of dimensions, and to find patterns.

In its simplest form Multidimensional Scaling turns distances into „maps”.

	Szczecin	Gdańsk	Olsztyn	Biały-stok	Warszawa	Poznań	Gorzów	Zielona Góra
Szczecin	0							
Gdańsk	287,5	0						
Olsztyn	393,2	135,8	0					
Biały-stok	574,1	327,1	192,2	0				
Warszawa	454,3	283,9	175,8	176,9	0,0			
Poznań	195,7	244,9	282,2	428,1	278,8	0		
Gorzów	90,0	288,7	367,7	533,8	395,4	119,8	0	
Zielona Góra	178,2	340,8	391,6	535,2	377,8	109,9	90,5	0

Basic mds is conducted by smacofSym()

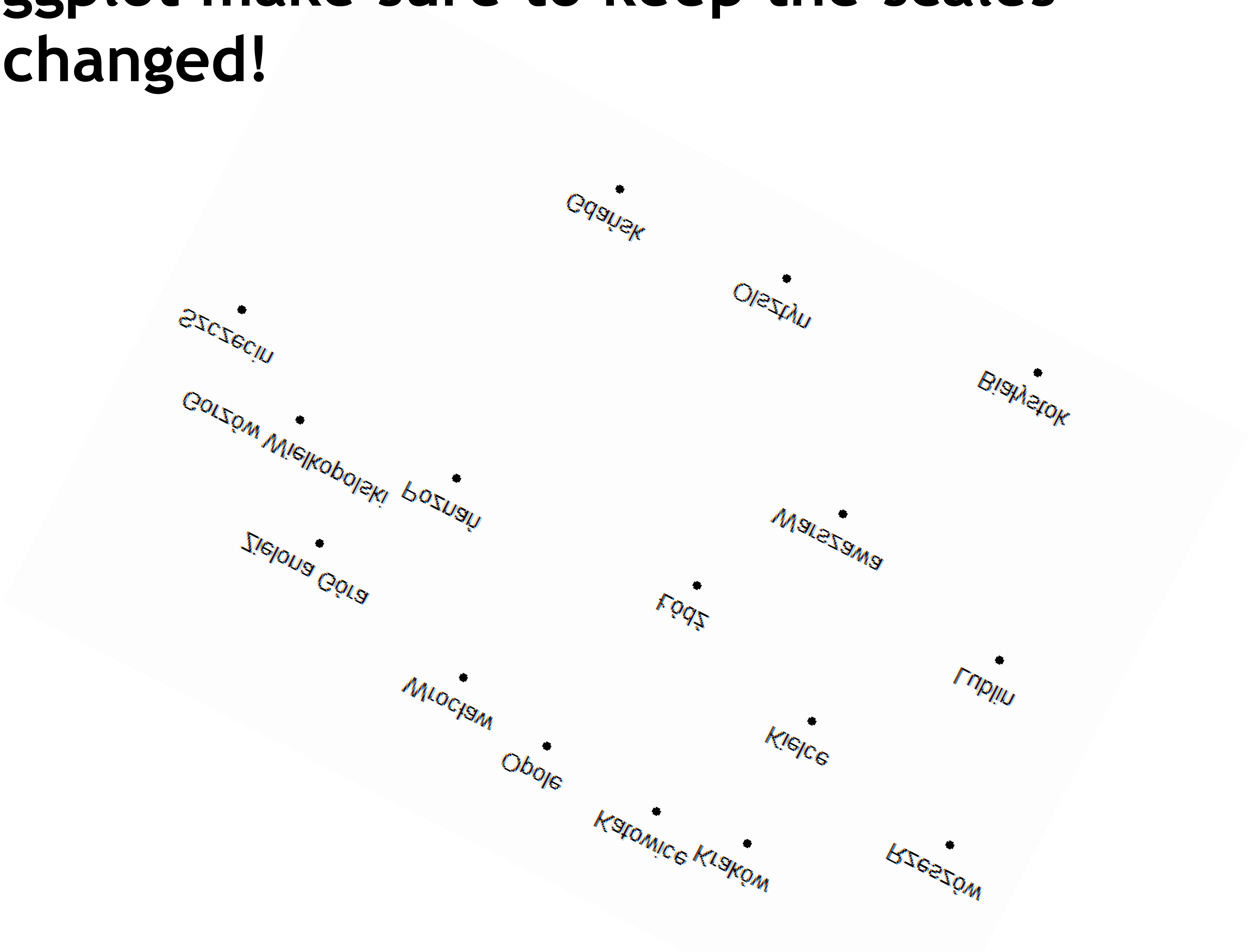
```
> library(smacof)
> miasta_mds <- smacofSym(odleglosci_miast)
> miasta_mds
Call: smacofSym(delta = odleglosci_miast,
ndim = 2, type = "ratio", init = "torgerson")
Model: Symmetric SMACOF
Number of objects: 16
Stress-1 value: 0
Number of iterations: 1

> plot(miasta_mds)
```


Configuration Plot
Opole
Katowice
Kraków
Wrocław
Zielona Góra
Rzeszów
Kielce
Gorzów Wielkopolski
Poznań
Łódź
Szczecin
Lublin
Warszawa
Gdańsk
Olsztyn
Białystok
Dimension 2
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-0.5
0.0
0.5
1.0
Dimension 1

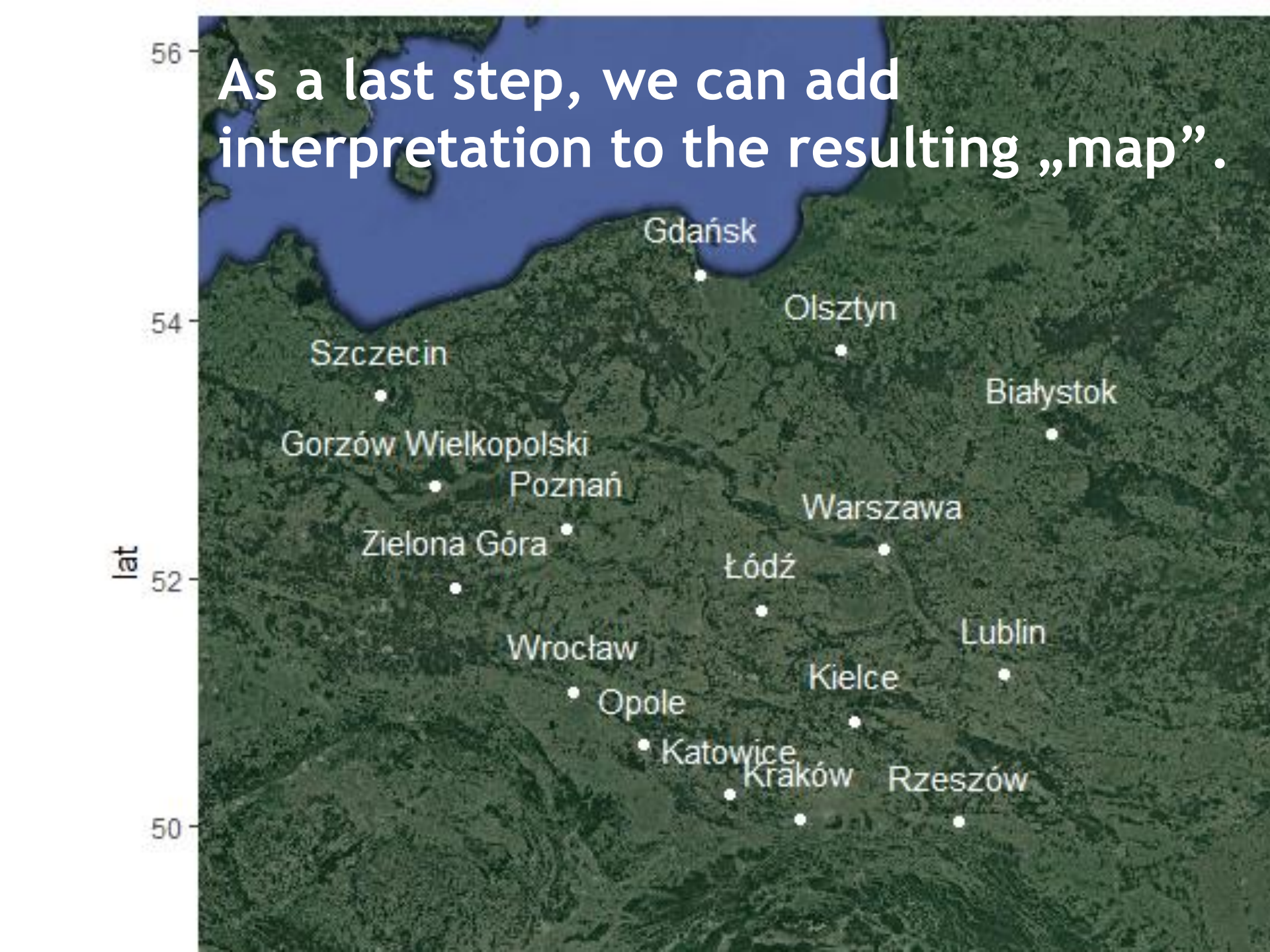
In ggplot make sure to keep the scales unchanged!

In ggplot make sure to keep the scales unchanged!

In ggplot make sure to keep the scales unchanged!



The configuration and distances stay the same when we flip or rotate the plot.


As a last step, we can add interpretation to the resulting „map”.
Gdańsk
Olsztyn
Szczecin
Białystok
Gorzów Wielkopolski
Poznań
Warszawa
Zielona Góra
Łódź
Lublin
Wrocław
Kielce
Opole
Katowice
Kraków
Rzeszów
lat
56
54
52
50

If we interpret distances as differences between objects, we can apply Multidimensional Scaling to variety of situations.

Differences can take many forms

- Correlations
- Items confused with one another
- Direct answers on differences between products
- Groupings
- …

An example

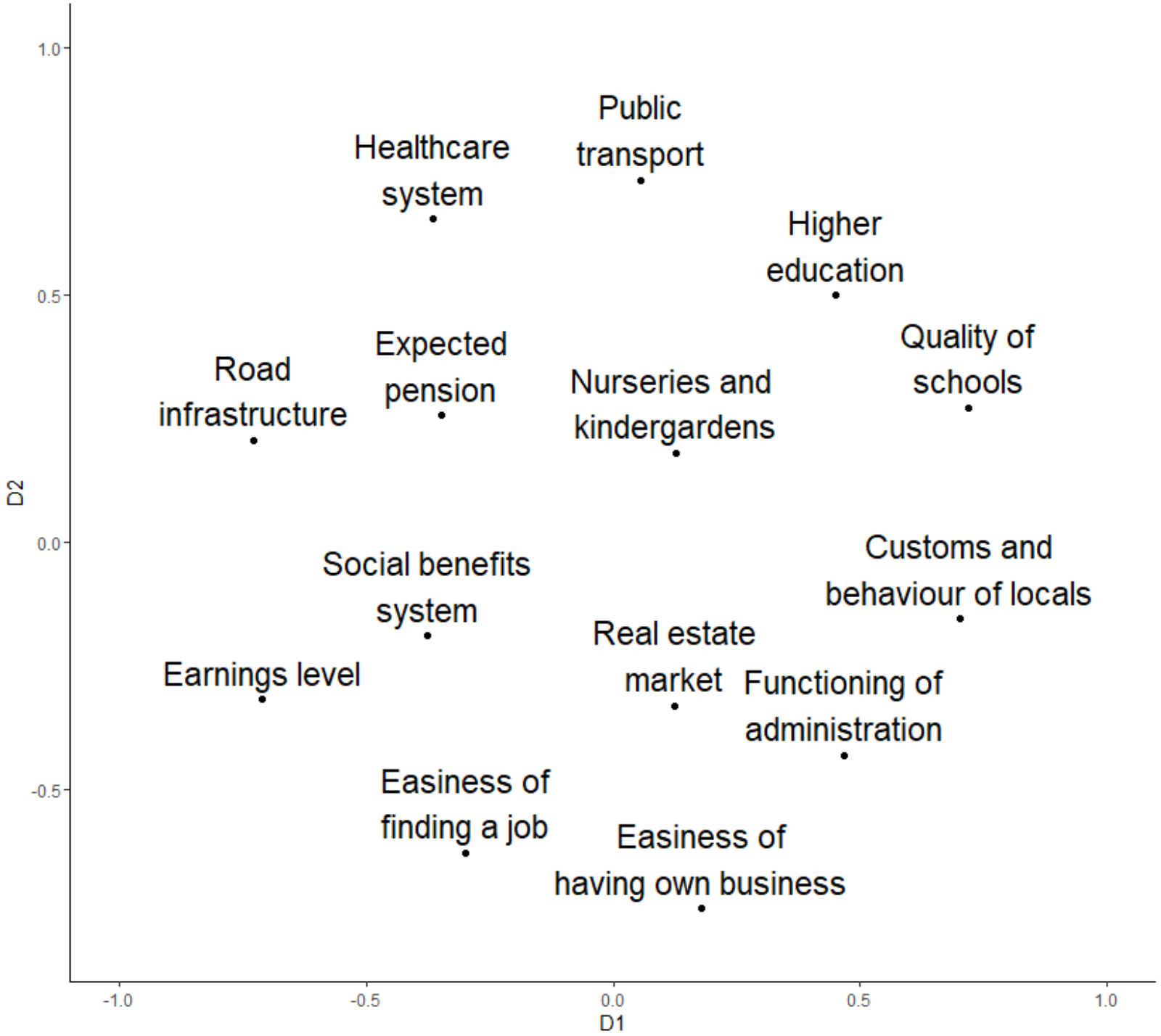
Please evaluate the following elements of socio-economic reality in Germany compared to Poland?

	definitely better in Germany	slightly better in Germany	no difference	slightly better in Poland	definitely better in Poland
	1	2	3	4	5
Easiness of finding a job					
Easiness of having own business					
Earnings level					
Social benefits system					
Healthcare system					
Expected pension					

```
> oceny_Kendall <- cor(oceny_data,
    method = "kendall",
    use = "pairwise.complete.obs")
```

	Easiness_of_finding_a_job	Easiness_of_having_own_buisness	Earnings_level	Social_benefits_system	Healthcare_system	Expected_pension	Functioning_of_administration	Nurseries_and_kindergardens	Quality_of_schools	Higher_education	Public_transport	Road_infrastructure	Real_estate_market	Customs_and_behaviour_of_locals
Easiness_of_finding_a_job	1	0.216674	0.207112	0.222064	0.07024	0.20594	0.211718	0.192332	0.059487	0.125124	0.15219	0.161621	0.206793	0.084884
Easiness_of_having_own_buisness	0.216674	1	0.106796	0.14789	0.041625	0.173403	0.209076	0.12935	0.145311	0.179453	0.08848	0.03914	0.231637	0.145842
Earnings_level	0.207112	0.106796	1	0.214672	0.173334	0.168041	0.144054	0.130552	0.037516	0.049917	0.15056	0.170623	0.106151	0.093622
Social_benefits_system	0.222064	0.14789	0.214672	1	0.1679	0.207044	0.214064	0.254282	0.068309	0.140152	0.134588	0.282446	0.222151	0.123249
Healthcare_system	0.07024	0.041625	0.173334	0.1679	1	0.185011	0.124701	0.219342	0.109609	0.140885	0.156531	0.166522	0.127965	0.1111
Expected_pension	0.20594	0.173403	0.168041	0.207044	0.185011	1	0.136402	0.174423	0.058555	0.164121	0.137651	0.157079	0.162311	0.182901
Functioning_of_administration	0.211718	0.209076	0.144054	0.214064	0.124701	0.136402	1	0.158916	0.196198	0.142755	0.145781	0.148435	0.231367	0.239064
Nurseries_and_kindergardens	0.192332	0.12935	0.130552	0.254282	0.219342	0.174423	0.158916	1	0.172537	0.238438	0.11618	0.15588	0.263137	0.142839
Quality_of_schools	0.059487	0.145311	0.037516	0.068309	0.109609	0.058555	0.196198	0.172537	1	0.272562	0.159541	0.071233	0.145073	0.189328
Higher_education	0.125124	0.179453	0.049917	0.140152	0.140885	0.164121	0.142755	0.238438	0.272562	1	0.137474	0.100736	0.145202	0.173508
Public_transport	0.15219	0.08848	0.15056	0.134588	0.156531	0.137651	0.145781	0.11618	0.159541	0.137474	1	0.158627	0.134822	0.082599
Road_infrastructure	0.161621	0.03914	0.170623	0.282446	0.166522	0.157079	0.148435	0.15588	0.071233	0.100736	0.158627	1	0.133316	0.081694
Real_estate_market	0.206793	0.231637	0.106151	0.222151	0.127965	0.162311	0.231367	0.263137	0.145073	0.145202	0.134822	0.133316	1	0.214489
Customs_and_behaviour_of_locals	0.084884	0.145842	0.093622	0.123249	0.1111	0.182901	0.239064	0.142839	0.189328	0.173508	0.082599	0.081694	0.214489	1

```
> oceny <- 1 - oceny_Kendall
> library(smacof)

> mds_oceny <- smacofSym(oceny)

> plot(mds_oceny)
```

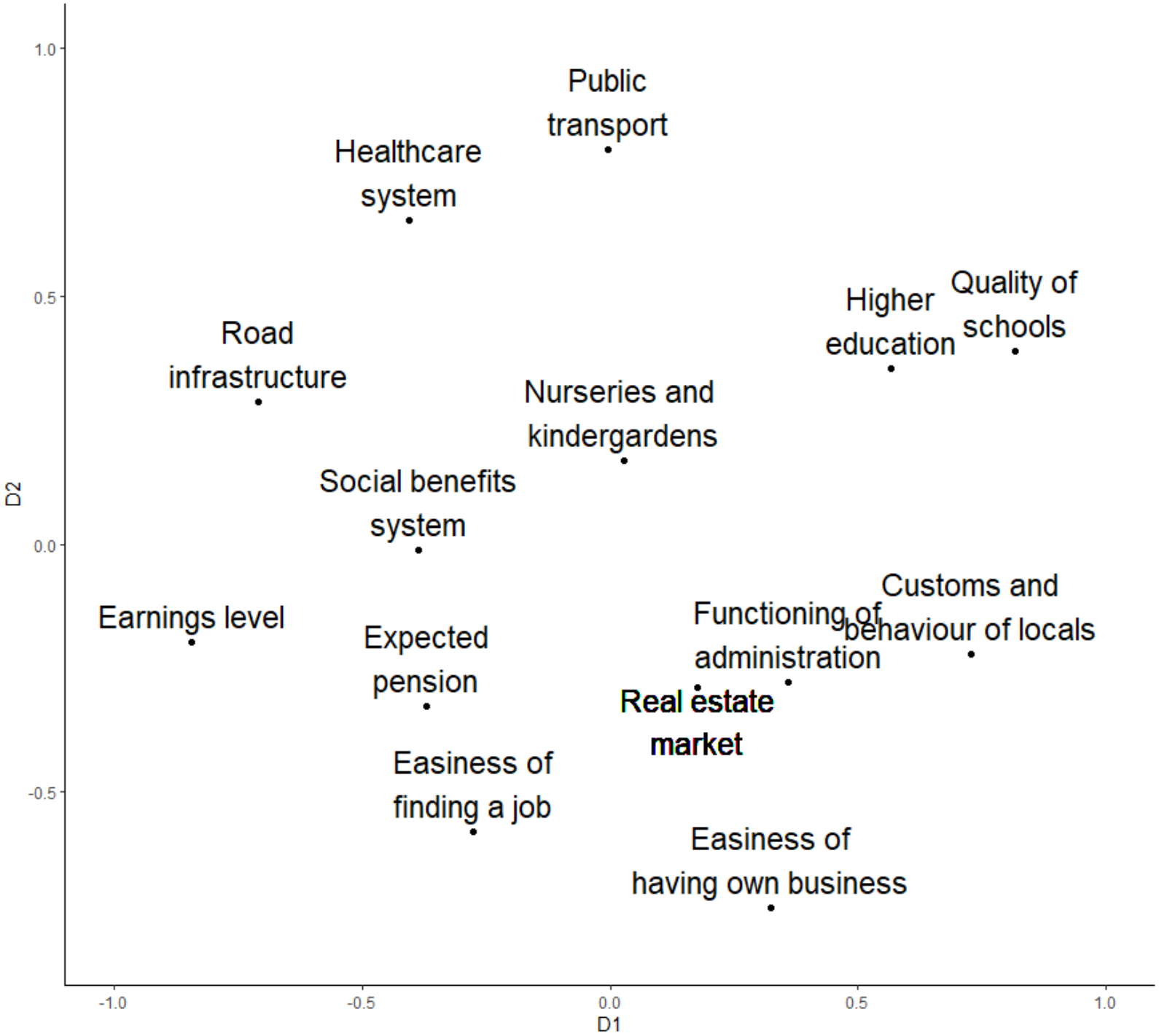

Public transport
Healthcare system
Higher education
Quality of schools
Expected pension
Road infrastructure
Nurseries and kindergardens
Customs and behaviour of locals
Social benefits system
Real estate market
Functioning of administration
Earnings level
Easiness of finding a job
Easiness of having own business
1.0
0.5
0.0
-0.5
D2
-1.0
-0.5
0.0
0.5
1.0
D1

Dissimilarities: levels of measurement

- Ratio
- **Interval**
- Ordinal: monotonne regression or splines

$$\delta_{i,j} = 1 - \tau_B(v_i, v_j)$$

```
> mds_oceny <- smacofSym(oceny)

> mds_oceny <- smacofSym(oceny,
                         type = "ratio")

> mds_oceny <- smacofSym(oceny,
                         type = "interval")
```


Public transport
Healthcare system
Quality of schools
Higher education
Road infrastructure
Nurseries and kindergardens
Social benefits system
Customs and behaviour of locals
Earnings level
Functioning of administration
Expected pension
Real estate market
Easiness of finding a job
Easiness of having own business
1.0
0.5
0.0
-0.5
D2
-1.0
-0.5
0.0
0.5
1.0
D1

The „best” map

Multidimensional Scaling chooses the map that reflects given dissimilarities best.

$$\sigma^2(\hat{\mathbf{D}}, \mathbf{X}) = \sum_{i<j} w_{ij}(\hat{d}_{ij} - d_{ij}(\mathbf{X}))^2$$

$$\hat{d}_{i,j} = f(\delta_{i,j})$$

A normalised version of the stress function: **stress-1**

$$\sigma_1(\hat{\mathbf{D}}, \mathbf{X}) = \sqrt{\frac{\sum_{i<j} w_{ij}(\hat{d}_{ij} - d_{ij}(\mathbf{X}))^2}{n(n-1)/2}}$$

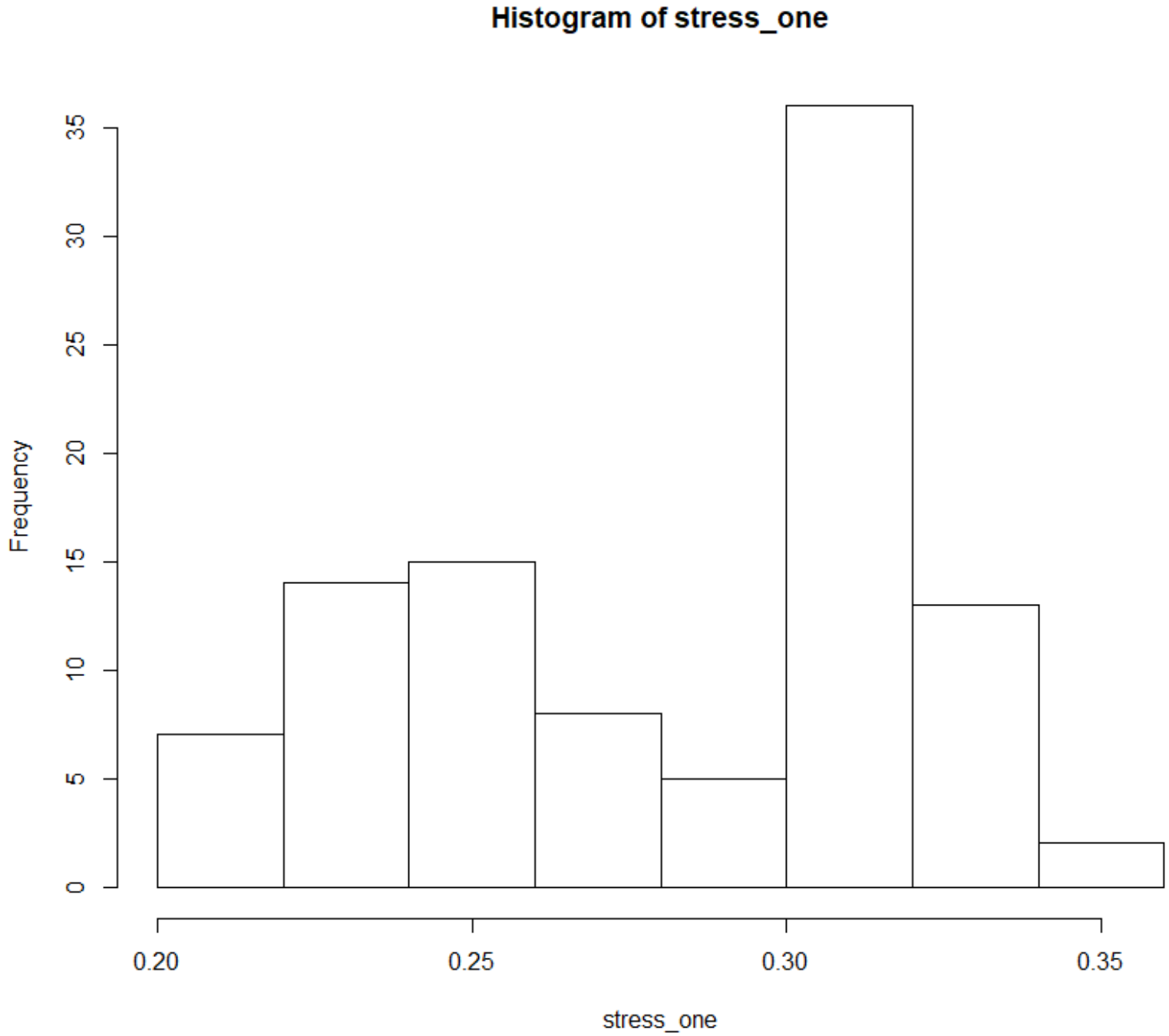
Multidimensional scaling is an iterative algorithm that uses majorization to make each new step (map) better than the previous one.

```
> smacofSym(oceny,
            ndim = 2,
            type = "interval",
            init = "torgerson")
Call:
mds(delta = oceny, type =
"interval")

Model: Symmetric SMACOF
Number of objects: 14
Stress-1 value: 0.211
Number of iterations: 26
```

```
> set.seed(123)
> smacofSym(oceny,
            ndim = 2,
            type = "ratio",
            init = "random" )
Call:
mds(delta = oceny)

Model: Symmetric SMACOF
Number of objects: 14
Stress-1 value: 0.238
Number of iterations: 193
```

```
>
> smacofSym(oceny,
            ndim = 2,
            type = "ratio",
            init = "random" )
Call:
mds(delta = oceny)

Model: Symmetric SMACOF
Number of objects: 14
Stress-1 value: 0.266
Number of iterations: 193
```

```
set.seed(123)
stress_one <- NULL
low_stress_mds <- NULL

for(i in 1:100) {
  oceny_mds <- smacofSym(oceny,
                         type = "interval",
                         init = "random")
  stress_one[i] <- oceny_mds$stress
  if (i == 1) low_stress_mds <- oceny_mds
  if (low_stress_mds$stress >
oceny_mds$stress ) {
      low_stress_mds <- oceny_mds
    }
}
```

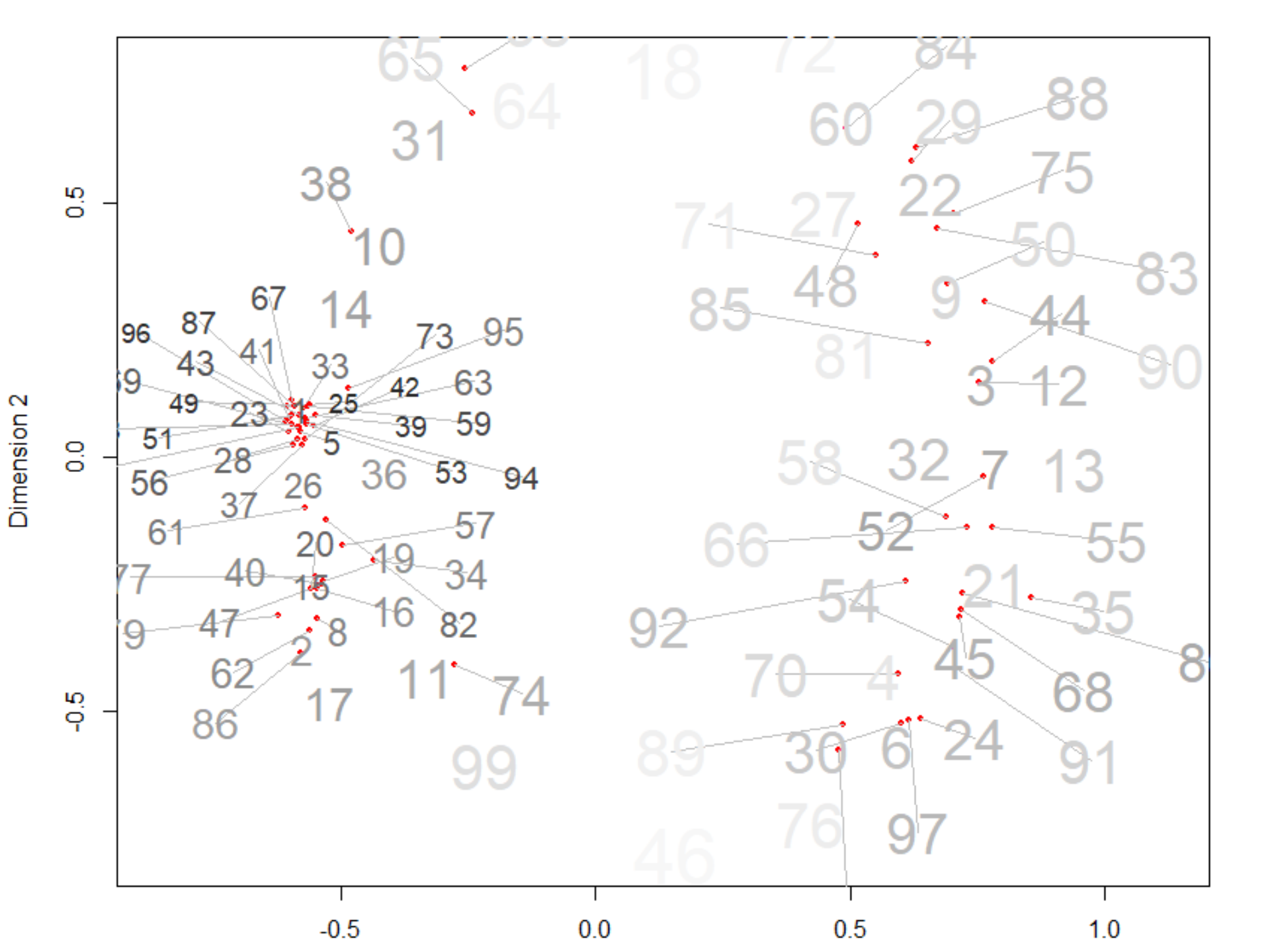

Histogram of stress_one
Frequency
0
5
10
15
20
25
30
35
0.20
0.25
0.30
0.35
stress_one

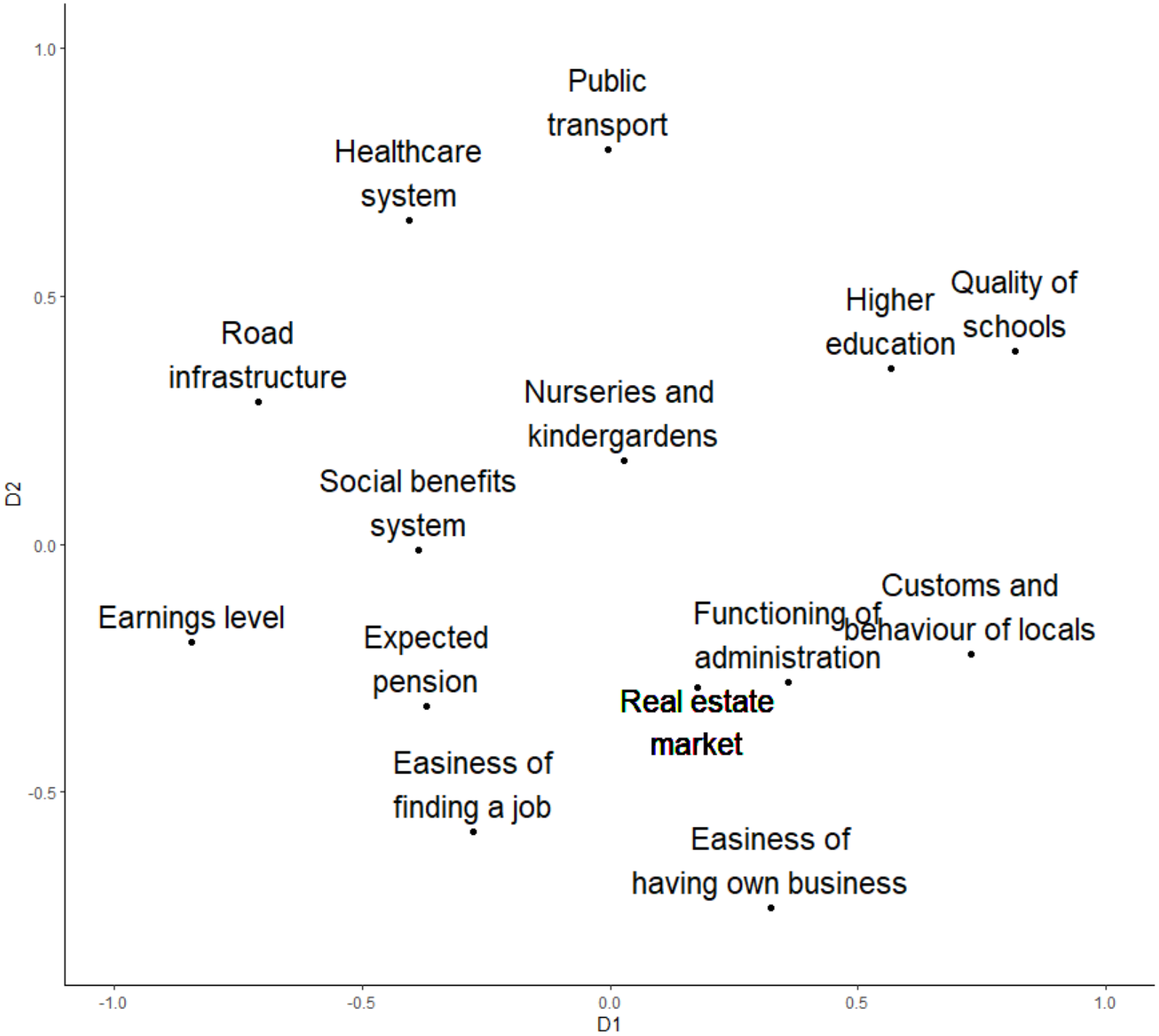
```
> low_stress_mds
Call: smacofSym(delta = oceny, type =
"interval", init = "random")

Model: Symmetric SMACOF
Number of objects: 14
Stress-1 value: 0.211
Number of iterations: 78
```

```
> low_stress_mds
Call: smacofSym(delta = oceny, type =
"interval", init = "random")

Model: Symmetric SMACOF
Number of objects: 14
Stress-1 value: 0.211
Number of iterations: 78

>smacofSym(delta = oceny, ,
          type = "interval",
          init = „torgerson")
…
Stress-1 value: 0.211
…
```

icExplore function tests multiple solutions and compares the map for you.

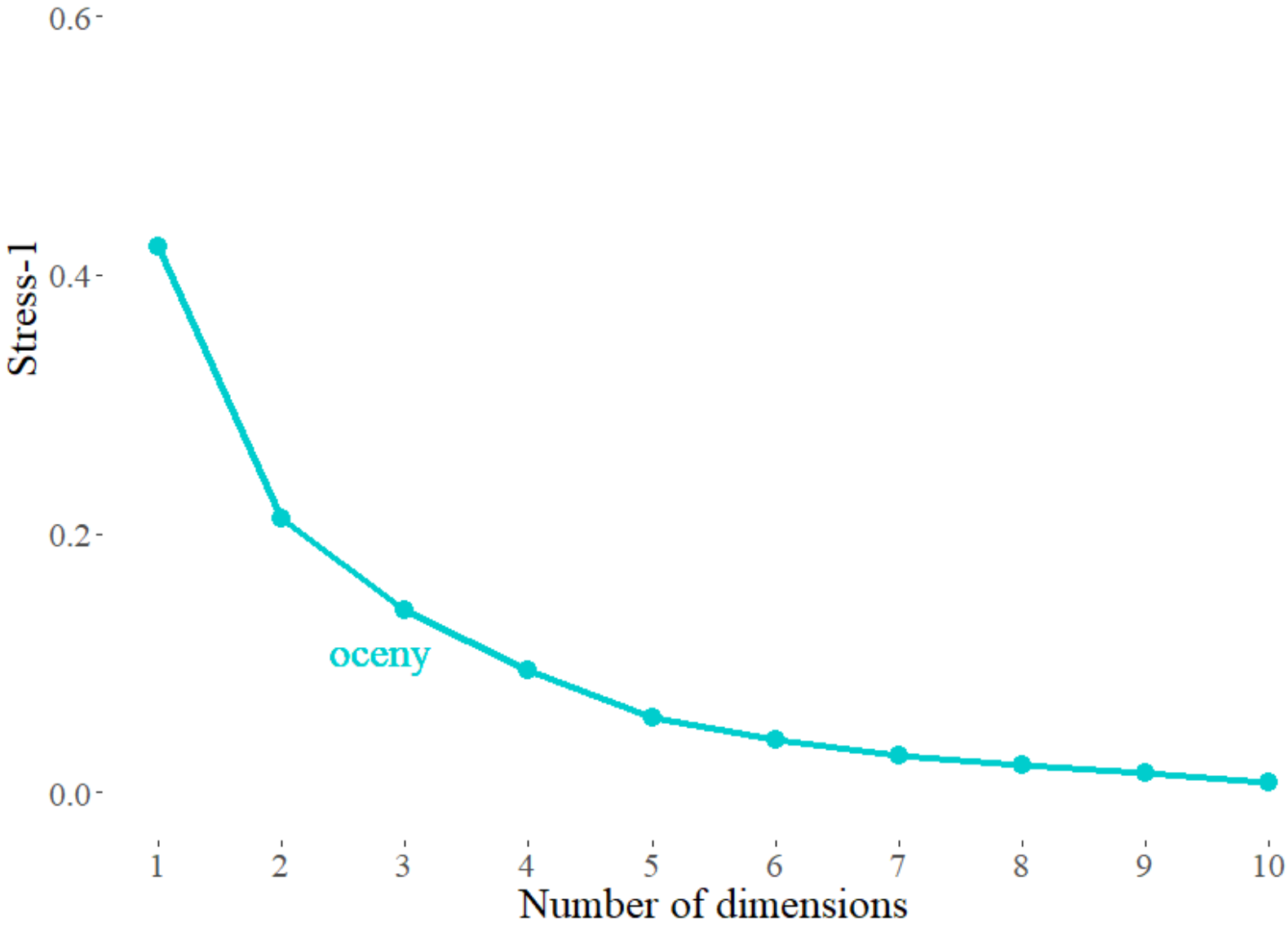
```
> set.seed(123)
> icExplore(oceny,  type = "interval",nrep =
100, ndim = 2)

Call: icExplore(delta = oceny, nrep = 100,
ndim = 2, type = "interval")

Number of replications: 100

Best stress value: 0.2115
Average stress value:  0.2824
Stress quantiles:
  2.5%    25%    50%    75%  97.5%
0.2115 0.2434 0.3005 0.3163 0.3297
```


Dimension 2
0.5
0.0
-0.5
-0.5
0.0
0.5
1.0


Public transport
Healthcare system
Quality of schools
Higher education
Road infrastructure
Nurseries and kindergardens
Social benefits system
Customs and behaviour of locals
Earnings level
Functioning of administration
Expected pension
Real estate market
Easiness of finding a job
Easiness of having own business
1.0
0.5
0.0
-0.5
D2
-1.0
-0.5
0.0
0.5
1.0
D1

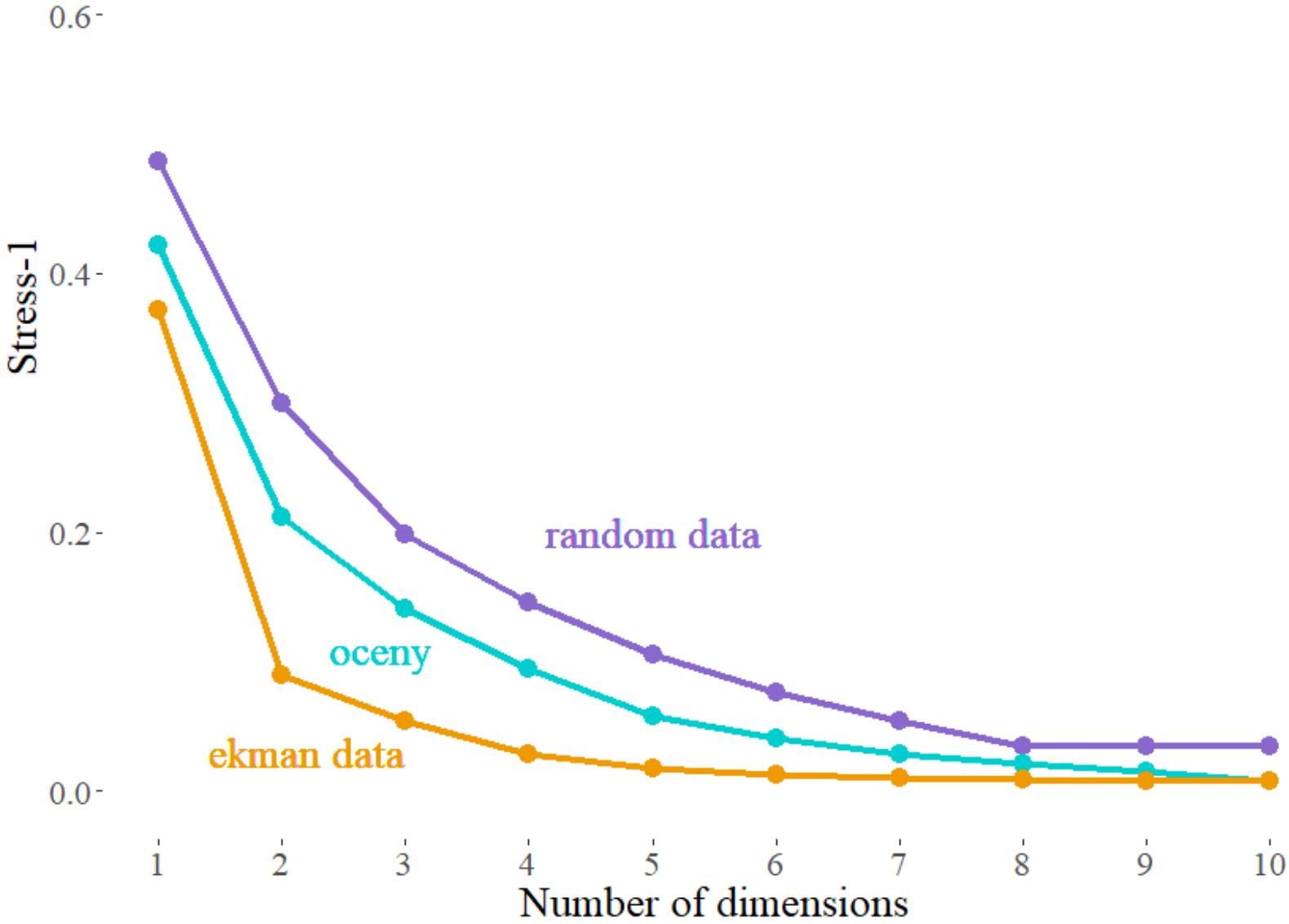
How many dimensions should my perception map have?

```
stress_of_oceny <- NULL

for(i in 1:10) {

 stress_of_oceny <- c(stress_of_oceny,
                      smacofSym(oceny,
                      type = "interval",
                      ndim=i)$stress)

}
```


0.6
0.4
0.2
0.0
Stress-1
oceny
1
2
3
4
5
6
7
8
9
10
Number of dimensions


0.6
0.4
0.2
0.0
Stress-1
oceny
ekman data
1
2
3
4
5
6
7
8
9
10
Number of dimensions

How good is my solution?


0.6
0.4
0.2
0.0
Stress-1
random data
oceny
ekman data
1
2
3
4
5
6
7
8
9
10
Number of dimensions

```
> mds_oceny <-
      smacofSym(oceny, type = "interval",ndim=2)
```

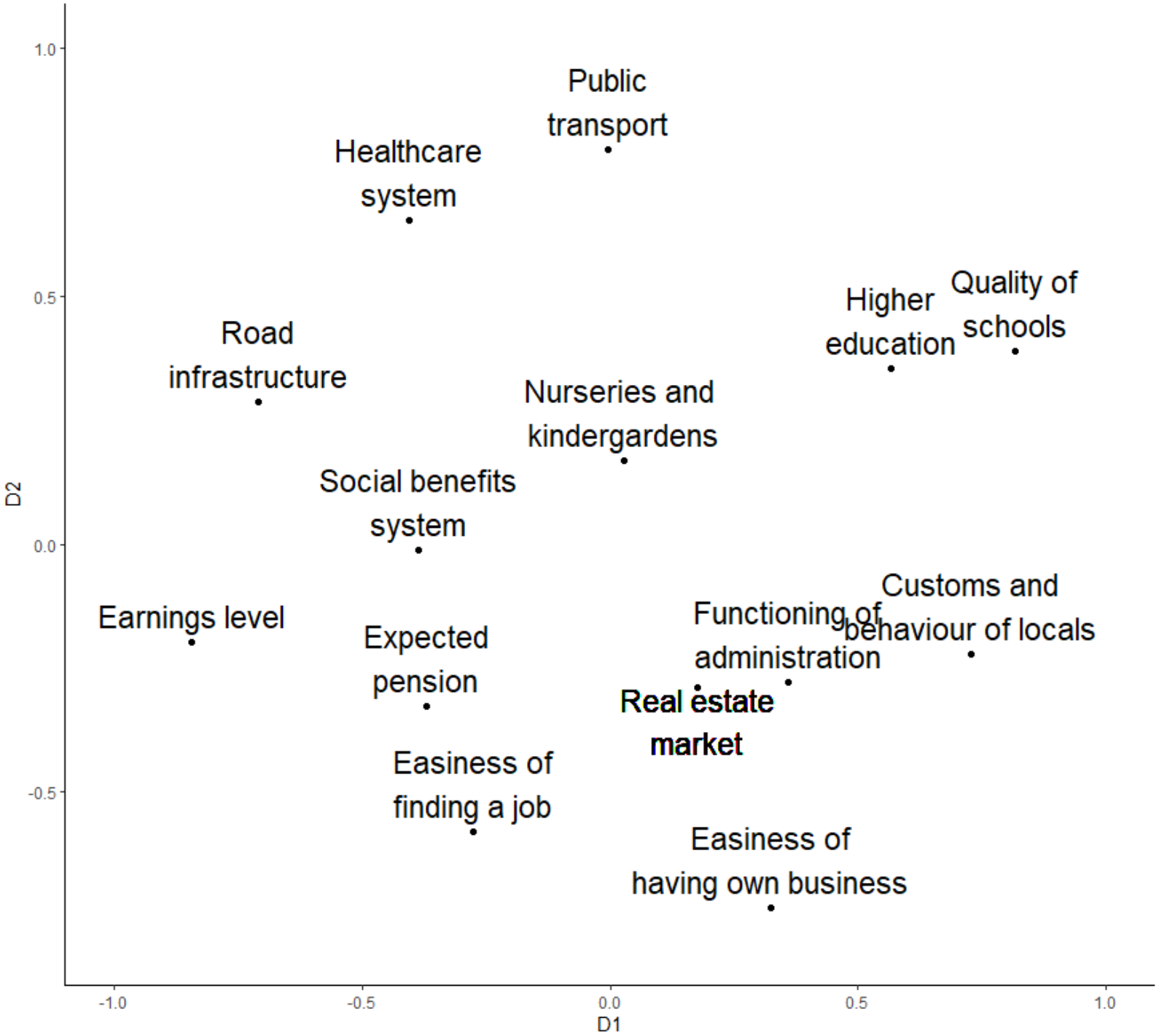
`> permtest(mds_oceny, nrep = 100, verbose = FALSE)`

```
Call: permtest.smacof(object = mds_oceny, nrep =
100, verbose = FALSE)

SMACOF Permutation Test
Number of objects: 14
Number of replications (permutations): 100

Observed stress value: 0.211
```

p-value: <0.001

Summary

```
> icExplore(oceny,type = "interval",
                ndim = 2, nrep = 100)

> mds_oceny <- smacofSym(oceny,
                    type = "interval",
                    init = "torgerson",
                    ndim = 2)

> permtest(mds_oceny, nrep = 100,
                verbose = FALSE)

> plot(mds_oceny)
```


Public transport
Healthcare system
Quality of schools
Higher education
Road infrastructure
Nurseries and kindergardens
Social benefits system
Customs and behaviour of locals
Earnings level
Functioning of administration
Expected pension
Real estate market
Easiness of finding a job
Easiness of having own business
1.0
0.5
0.0
-0.5
D2
-1.0
-0.5
0.0
0.5
1.0
D1

Thank you for your attention!

Barbara Jancewicz
barbara@jancewicz.net

More reading

Borg I, Groenen PJF (2005). *Modern Multidimensional Scaling: Theory and Applications*. 2nd edition. Springer, New York.

Borg I, Groenen PJF, Mair P (2018). *Applied Multidimensional Scaling and Unfolding*. 2ndedition. Springer, New York

Cox TF, Cox MAA (1991). *Multidimensional Scaling*. 2nd edition. Chapman & Hall/CRC, Boca Raton

De Leeuw J, Mair P (2009b). "Multidimensional Scaling Using Majorization: SMACOF in R." *Journal of Statistical Software*, 31(3), 1–30.

Package 'smacof'

June 6, 2019

Type Package

Title Multidimensional Scaling

Version 2.0-0

Date 2019-06-05

Description
Implements the following approaches for multidimensional scaling (MDS) based on stress minimization using majorization (smacof): ratio/interval/ordinal/spline MDS on symmetric dissimilarity matrices, MDS with external constraints on the configuration, individual differences scaling (idioscal, indscal), MDS with spherical restrictions, and ratio/interval/ordinal/spline unfolding (circular restrictions, row-conditional). Various tools and extensions like jackknife MDS, bootstrap MDS, permutation tests, MDS biplots, gravity models, unidimensional scaling, drift vectors (asymmetric MDS), classical scaling, and Procrustes are implemented as well.

Imports graphics, stats, polynom, Hmisc, colorspace, nnls, grDevices, MASS, weights, ellipse, wordcloud, candisc, parallel, foreach, doParallel

Depends R (>= 3.2.0), plotrix

License GPL-3

Suggests knitr, prefmod, MPsychoR, calibrate

VignetteBuilder knitr

LazyData yes

LazyLoad yes

ByteCompile yes

NeedsCompilation yes

Author Patrick Mair [aut, cre], Jan De Leeuw [aut], Patrick J. F. Groenen [aut], Ingwer Borg [ctb]

Maintainer Patrick Mair <mair@fas.harvard.edu>

Repository CRAN