Package ‘mdsOpt’

October 8, 2019

Title Searching for Optimal MDS Procedure for Metric and Interval-Valued Data

Version 0.4-3

Date 2019-09-18

Author Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Maintainer Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Depends R (>= 3.6.0), smacof, clusterSim, symbolicDA

Imports smds, animation, plotrix, spdep

Suggests testthat, R.rsp

VignetteBuilder R.rsp


License GPL (>= 2)

URL http://keii.ue.wroc.pl/mdsOpt

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-08 14:00:02 UTC
**data_lower_silesian**

The evaluation of tourist attractiveness of Lower Silesian counties

**Description**

The empirical study uses the statistical data presented in the article (Gryszel, Walesiak, 2014) and referring to the attractiveness level of 31 objects (29 Lower Silesian counties, pattern and antipattern object) The evaluation of tourist attractiveness of Lower Silesian counties was performed using 16 metric variables (measured on a ratio scale): x1 – beds in hotels per 1 km2 of a county area, x2 – number of nights spent daily by resident tourists per 1000 inhabitants of a county, x3 – number of nights spent daily by foreign tourists per 1000 inhabitants of a county, x4 – gas pollution emission in tons per 1 km2 of a county area, x5 – number of criminal offences and crimes against life and health per 1000 inhabitants of a county, x6 – number of property crimes per 1000 inhabitants of a county, x7 – number of historical buildings per 100 km2 of a county area, x8 – x9 – x10 – number of events as well as cultural and tourist ventures in a county, x11 – number of natural monuments calculated per 1 km2 of a county area, x12 – number of tourist economy entities per 1000 inhabitants of a county (natural and le-gal persons), x13 – expenditure of municipalities and counties on tourism, culture and national heritage protection as well as physical culture per 1 inhabitant of a county in PLN, x14 – viewers in cinemas per 1000 inhabitants of a county, x15 – museum visitors per 1000 inhabitants of a county, x16 – number of construction permits (hotels and accommodation buildings, commercial and service buildings, transport and communication buildings, civil and water engineering constructions) issued in a county in the years 2011-2012 per 1 km2 of a county area.

The statistical data were collected in 2012 and come from the Local Data Bank of the Central Statistical Office of Poland, the data for x7 variable only were obtained from the regional conservation officer.

**Format**

data.frame: 31 objects (29 counties, pattern and antipattern object, 16 variables. The coordinates of a pattern object cover the most preferred preference variable (stimulants, destimulants, nominants) values. The coordinates of an anti-pattern object cover the least preferred preference variable values.
**drawIsoquants**

**Source**


**Examples**

```r
print("uncomment to run - approximately 7 seconds runtime")
# uncomment to run - approximately 7 seconds runtime
# library(mdsOpt)
# metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
# metscale<-c("ratio","interval")
# metdist<-c("euclidean","GDM1")
# data(data_lower_silesian)
# res<-optSmacofSym_mMDS(data_lower_silesian,normalizations=metnor,
# distances=metdist,mdsmodels=metscale)
# print(findOptimalSmacofSym(res))
```

---

**drawIsoquants**  
**draw series of isoquants**

**Description**

function draw series of isoquants (a contour line drawn through the set of points at which the same quantity of output is produced while changing the quantities of two or more inputs)

**Usage**

```r
drawIsoquants(x,y=NULL,number=6,steps=NULL)
```

**Arguments**

- **x**
  - two dimensional point (center)
- **y**
  - optional - second point, used for calculations of step size if steps is null
- **number**
  - number of isoquants
- **steps**
  - distance between following isoquants starting from x, if length of this arguments is lower than number argument last item is repeated

**Author(s)**

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>  
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland  
References

Walesiak, M., (2016), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonometria, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.


Examples

#Example 1
library(mdsOpt)
library(smacof)
library(clusterSim)
data(data_lower_silesian)
z<-data.Normalization(data_lower_silesian, type="n1")
d<-dist.GDM(z, method="GDM1")
res <- smacofSym(delta=d,ndim=2,type="interval")
print("Objects configuration", quote=FALSE)
plot(res, plot.type="confplot")
r1<-res$conf[nrow(z),1]
r2<-res$conf[nrow(z),2]
r3<-res$conf[nrow(z)-1,1]
r4<-res$conf[nrow(z)-1,2]
arrows(r1,r2,r3,r4,length=0.1,col="black")
res_up<-as.matrix(dist(res$conf,method="euclidean"))
drawIsoquants(res$conf[nrow(z)-1,],steps=max(res_up)/6)
# or
# drawIsoquants(res$conf[nrow(z)-1,],steps=c(0.3,0.2),number=8)

#Example 2
library(mdsOpt)
library(smacof)
library(clusterSim)
data(data_lower_silesian)
z<-data.Normalization(data_lower_silesian, type="n1")
d<-dist.GDM(z, method="GDM1")
res<-smacofSym(delta=d,ndim=2,type="interval")
res1<-res$conf
#write.table(res1,"conf_2d.csv",dec="","",sep=";",col.names=NA,row.names=TRUE)
alfa<- 1.05*pi
a<- cos(alfa)
b<- -sin(alfa)
c<- sin(alfa)
d<- cos(alfa)
D<-array(c(a,b,c,d), c(2,2))
#res1<-read.csv2("conf_2d.csv", header=TRUE, row.names=1)
res1<-as.matrix(res1)
findOptimalIscalInterval

Selecting the optimal I-Scal multidimensional scaling procedure for interval-valued data

Description

Selecting the optimal multidimensional scaling procedure - I-Scal (by varying all combinations of normalization and optimization methods)

Usage

findOptimalIscalInterval(table, critical_stress = (max(as.numeric(gsub("", ",", table[, "I-STRESS"], fixed = T))) + min(as.numeric(gsub("", ",", table[, "I-STRESS"], fixed = T)))) / 2, critical_HHI = NA)

Arguments

table result from optSmacofSym_nMDS. Data frame ordered by increasing value of I-Stress fit measure with columns:
Normalization method
Optimization method
I-STRESS
HHI spb
critical_stress threshold value of I-Stress fit measure. Default - mid-range of I-Stress fit measures calculated for all MDS procedures
critical_HHI threshold value of Hirschman-Herfindahl HHI index. Only one parameter critical_stress or critical_HHI can be set, and the function finds the optimal value among the procedures for which the selected measure is lower or equal treshold value
findOptimalIscalInterval

Value

<table>
<thead>
<tr>
<th>Nr</th>
<th>number of row in table with optimal multidimensional scaling procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalization_method</td>
<td>normalization method used for optimal multidimensional scaling procedure</td>
</tr>
<tr>
<td>I_STRESS</td>
<td>value I-Stress fit measure for optimal multidimensional scaling procedure</td>
</tr>
<tr>
<td>HHI_spb</td>
<td>Herfindahl-Hirschman HHI index, calculated based on stress per box, for optimal multidimensional scaling procedure</td>
</tr>
</tbody>
</table>

Author(s)

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/mdsOpt

References


Walesiak, M. (2016), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonometria, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.


See Also

data.Normalization, interval_normalization, IMDS
findOptimalSmacofSym

Examples

```
print("uncomment to run - approximately a few seconds runtime")
#library(clusterSim)
#library(mdsOpt)
#data(data_symbolic_interval_polish_voivodships)
#x<-data_symbolic_interval_polish_voivodships
#metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
#methods<-c("MM","BFGS")
#w<-optIscalInterval(x,dataType="simple",normalizations=metnor,optMethods=methods,outDec=".")
#print(findOptimalIscalInterval(w))
```

findOptimalSmacofSym  Selecting the optimal multidimensional scaling (MDS) procedure

Description

Selecting the optimal multidimensional scaling procedure - metric MDS (by varying all combinations of normalization methods, distance measures, and metric MDS models) and nonmetric MDS (by varying all combinations of normalization methods and distance measures)

Usage

```
findOptimalSmacofSym(table, critical_stress=(max(as.numeric(gsub("",".",table[,"STRESS 1"],fixed=T)))+
  min(as.numeric(gsub("",".",table[,"STRESS 1"],fixed=T))))/2, critical_HHI=NA)
```

Arguments

- **table**: result from `optSmacofSym_nMDS` or `optSmacofSym_mMDS`. Data frame ordered by increasing value of Stress-1 fit measure or HHI index with columns:
  - Normalization method
  - Distance measure
  - MDS model
  - Spline degree (valid only for optSmacofSym_mMDS results)
  - STRESS 1
  - HHI spp
- **critical_stress**: threshold value of Kruskal’s Stress-1 fit measure. Default - mid-range of Kruskal’s Stress-1 fit measures calculated for all MDS procedures
- **critical_HHI**: threshold value of Hirschman-Herfindahl HHI index. Only one parameter critical_stress or critical_HHI can be set, and the function finds the optimal value among the procedures for which the selected measure is lower or equal threshold value
### Value

<table>
<thead>
<tr>
<th>Nr</th>
<th>number of row in table with optimal multidimensional scaling procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalization_method</td>
<td>normalization method used for optimal multidimensional scaling procedure</td>
</tr>
<tr>
<td>MDS_model</td>
<td>MDS model used for optimal multidimensional scaling procedure</td>
</tr>
<tr>
<td>Spline_degree</td>
<td>Additional spline.degree value for optimal procedure, if mspline model is used for simulation. For other models there is no value for this field</td>
</tr>
<tr>
<td>Distance_measure</td>
<td>distance measure used for optimal multidimensional scaling procedure</td>
</tr>
<tr>
<td>STRESS_1</td>
<td>value of Kruskal Stress-1 fit measure for optimal multidimensional scaling procedure</td>
</tr>
<tr>
<td>HHI_spp</td>
<td>Hirschman-Herfindahl HHI index, calculated based on stress per point, for optimal multidimensional scaling procedure</td>
</tr>
</tbody>
</table>

### Author(s)

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

http://keii.ue.wroc.pl/mdsOpt

### References


Walesiak, M. (2016b), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonometria, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.

See Also

data.Normalization, dist.GDM, dist.smacofSym

Examples

print("uncomment to run - approximately 7 seconds runtime")
# uncomment to run - approximately 7 seconds runtime
# library(mdsOpt)
# metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
# metscale<-c("ratio","interval")
# metdist<-c("euclidean","manhattan","maximum","seuclidean","GDM1")
# data(data_lower_silesian)
# res<-optSmacofSym_mMDS(data_lower_silesian,normalizations=metnor,
# distances=metdist,mdsmodels=metscale,outDec=".")
# print(findOptimalSmacofSym(res))

ispb Calculation of I-stress per box indices for multidimensional scaling procedure for interval-valued data

Description

Calculation of I-stress per box indices for multidimensional scaling procedure for interval-valued data

Usage

ispb(EIDM,idiss)

Arguments

EIDM the interval-valued dissimilarity matrix IDM (an object of class "array": IDM[1,]: the lower dissimilarity matrix; IDM[2,]: the upper dissimilarity matrix) in reduced space
idiss the primary interval-valued dissimilarity matrix

Value

The vector of i-stress per box percentage values

Author(s)

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/mdsOpt
References


See Also
data.Normalization, interval_normalization, IMDS

Examples

library(mdsOpt)
library(clusterSim)
library(smds)
data(data_symbolic_interval_polish_voivodships)
x1<-data_symbolic_interval_polish_voivodships[,,1]
y1<-data_symbolic_interval_polish_voivodships[,,2]
norm_type="n2"
normalized<-interval_normalization(x=x1,y=y1,dataType="separate_tables",type=norm_type)
x<-normalized$simple[,,1]
y<-normalized$simple[,,2]
my.idiss<-idistBox(X=(x+y)/2,R=(y-x)/2)
#Apply the hyperbox model via the MM algorithm
cmat<-(my.idiss[2, , ] + my.idiss[1, , ])/2
iniX<-cmdscale(as.dist(cmat), k = 2)
n=dim(my.idiss)[2]
iniR<-matrix(rep(1,n * 2), nrow = n, ncol = 2)
res.mm_box<-IMDS(IDM=my.idiss, p=2,model="box",opt.method="MM", ini=list(iniX,iniR))
plot(res.mm_box)
title(main="box_MM")
windows()
spb<-ispb(res.mm_box$EIDM,my.idiss)
w<-sort(spb,decreasing=TRUE)
print(spb)
names(w)<-order(spb,decreasing = TRUE)
plot(w, xlab="Object", ylab="spb in percents")
text(w,pos=1,names(w))
optIscalInterval

Selecting the optimal multidimensional scaling procedure for interval-valued data

Description

Selecting the optimal multidimensional scaling procedure by varying all combinations of normalization and optimization methods

Usage

optIscalInterval(x, dataType="simple", normalizations=NULL, optMethods=NULL, outputCsv="", outputCsv2="", y=NULL, outDec="", stressDigits=6, HHIDigits=2, ...)

Arguments

x
interval-valued data table or matrix or dataset
dataType
Type of symbolic data table passed to function:
'sda' - full symbolicDA format object;
'simple' - three dimensional array with lower and upper bound of intervals in third dimension;
'separate_tables' - lower bound of intervals in x, upper bound of intervals in y (formula y=... needed in argument list);
'rows' - lower and upper bound of intervals in neighbouring rows;
'columns' - lower and upper bound of intervals in neighbouring columns

normalizations
optional, vector of normalization methods that should be used in procedure

optMethods
optional, vector of optimization methods

outputCsv
optional, name of csv file with results

outputCsv2
optional, name of csv (comma as decimal point sign) file with results

y
matrix or dataset with upper bounds of intervals if argument dataType is equal to "separate_tables"

outDec
decimal sign used in returned table

stressDigits
Number of decimal digits for displaying I-Stress value

HHIDigits
Number of decimal digits for displaying HHI spb value

... arguments passed to smds I-scal implementation (function IMDS), like p, maxit, eps and others
Details

Parameter normalizations may be the subset of the following values:
"n1","n2","n3","n3a","n4","n5","n5a","n6","n6a",
"n7","n8","n9","n9a","n10","n11","n12","n12a","n13"
(e.g. normalizations=c("n1","n2","n3","n5","n5a",
"n8","n9","n9a","n11","n12a","n13"))

if normalizations is set to "n0" no normalization is applied

Parameter optMethods may be the subset of the following values (IMDS):
("MM","BFGS")

Value

Data frame ordered by increasing value of Stress-1 fit measure with columns:

<table>
<thead>
<tr>
<th>Normalization method</th>
<th>Opt method</th>
<th>Spline degree</th>
<th>I-STRESS</th>
<th>HHI spb</th>
</tr>
</thead>
<tbody>
<tr>
<td>normalization method used for p-th multidimensional scaling procedure</td>
<td>Optimization method used IMDS I-Scal implementatiomn</td>
<td>Additional spline.degree value if mspline model is used for simulation, for other models there is no value in this cell</td>
<td>value of I-Stress fit measure for p-th multidimensional scaling procedure</td>
<td>Hirschman-Herfindahl HHI index calculated based on stress per boc for p-th multidimensional scaling procedure</td>
</tr>
</tbody>
</table>

Author(s)

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/mdsOpt

References


Walesiak, M. (2014), Przegląd formuł normalizacji wartości zmiennych oraz ich własności w statystycznej analizie wielowymiarowej [Data Normalization in Multivariate Data Analysis. An Overview

Walesiak, M. (2016), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonometria, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.


See Also
data.Normalization, interval_normalization, IMDS

Examples

# print("uncomment to run - approximately 7 seconds runtime")
# library(mdsOpt)
# library(clusterSim)
# data(data_symbolic_interval_polish_voivodships)
# x<-data_symbolic_interval_polish_voivodships
# metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
# methods<-c("MM","BFGS")
# res<-optIscalInterval(x,dataType="simple",normalizations=metnor,optMethods=methods,outDec=".")
# Istress<-as.numeric(gsub("","",res[,"I-STRESS"],fixed=TRUE))
# hhi<-as.numeric(gsub("","",res[,"HHI spb"],fixed=TRUE))
# t<-findOptimalIscalInterval(res)
# cs<-(min(Istress)+max(Istress))/2 # critical I-stress
# print(t)

optSmacofSymInterval

Selecting the optimal multidimensional scaling procedure for interval-valued data

Description

Selecting the optimal multidimensional scaling procedure by varying all combinations of normalization methods, distance measures for interval-valued data, and metric MDS models

Usage

optSmacofSymInterval(x, dataType="simple", normalizations=NULL, distances=NULL, mdsmodels=NULL, spline.degrees=c(2), outputCsv="", outputCsv2="", y=NULL, outDec="", stressDigits=6, HHIDigits=2,...)
Arguments

x  interval-valued data table or matrix or dataset
dataType  Type of symbolic data table passed to function:
' sda' - full symbolicDA format object;
'simple' - three dimensional array with lower and upper bound of intervals in third dimension;
'separate_tables' - lower bound of intervals in x, upper bound of intervals in y;
'rows' - lower and upper bound of intervals in neighbouring rows;
'columns' - lower and upper bound of intervals in neighbouring columns
normalizations  optional, vector of normalization methods that should be used in procedure
distances  optional, vector of distance measures (Hausdorff, Ichino-Yaguchi) that should be used in procedure
mdsmodels  optional, vector of multidimensional models (ratio, interval, mspline) that should be used in procedure
spline.degrees  optional, vector (e.g. 2:4) of spline.degree parameter values that should be used in procedure for mspline model
outputCsv  optional, name of csv file with results
outputCsv2  optional, name of csv (comma as decimal point sign) file with results
y  matrix or dataset with upper bounds of intervals if argument dataType is equal to "separate_tables"
outDec  decimal sign used in returned table
stressDigits  Number of decimal digits for displaying Stress 1 value
HHIDigits  Number of decimal digits for displaying HHI spp value
...  arguments passed to smacofSym, like ndim, itmax, eps and others

Details

Parameter normalizations may be the subset of the following values:
"n1","n2","n3","n3a","n4","n5","n5a","n6","n6a",
"n7","n8","n9","n9a","n10","n11","n12","n12a","n13"
(e.g. normalizations=c("n1","n2","n3","n5","n5a",
"n8","n9","n9a","n11","n12a"))
if normalizations is set to 'n0' no normalization is applied

Parameter distances may be the subset of the following values:
"H_q1","H_q2","U_2_q1","U_2_q2" (In following order: Hausdorff distance with q=1, Euclidean Hausdorff distance with q=2, Ichino-Yaguchi distance with q=1; Euclidean Ichino-Yaguchi distance with q=2)
(e.g. distances=c("H_q1","U_2_q1"))

Parameter mdsmodels may be the subset of the following values (metric MDS):
"ratio","interval","mspline" (e.g. c("ratio","interval"))
**Value**

Data frame ordered by increasing value of Stress-1 fit measure with columns:

- **Normalization method**: normalization method used for p-th multidimensional scaling procedure
- **MDS model**: MDS model used for p-th multidimensional scaling procedure
- **Spline degree**: Additional spline.degree value if mspline model is used for simulation, for other models there is no value in this cell
- **Distance measure**: distance measures for interval-valued data used for p-th multidimensional scaling procedure
- **STRESS 1**: value of Kruskal Stress-1 fit measure for p-th multidimensional scaling procedure
- **HHI spp**: Hirschman-Herfindahl HHI index calculated based on stress per point for p-th multidimensional scaling procedure

**Author(s)**

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland


**References**


**See Also**

data.Normalization, interval_normalization, dist.Symbolic, smacofSym
Examples

```r
# Uncomment to run
# library(mdsOpt)
# library(clusterSim)
# data(data_symbolic_interval_polish_voivodships)
# metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
# metscale<-c("ratio","interval","mspline")
# metdist<-c("H_q1","H_q2","U_2_q1","U_2_q2")
# res<optSmacofSymInterval(data_symbolic_interval_polish_voivodships,dataType="simple",
# normalizations=metnor,distances=metdist,mdsmodels=metscale,spline.degrees=c(2,3),outDec="."
# stress<as.numeric(gsub("","",res[,"STRESS 1"],fixed=TRUE))
# hhi<as.numeric(gsub("","",res[,"HHI spp"],fixed=TRUE))
# t<findOptimalSmacofSym(res)
# cs<-(min(stress)+max(stress))/2 # critical stress
# plot(stress[-t$Nr],hhi[-t$Nr], xlab="Stress-1", ylab="HHI",type="n",font.lab=3)
# text(stress[-t$Nr],hhi[-t$Nr],labels=(1:nrow(res))[-t$Nr])
# abline(v=cs,col="red")
# points(stress[t$Nr],hhi[t$Nr], cex=5,col="red")
# text(stress[t$Nr],hhi[t$Nr],labels=(1:nrow(res))[t$Nr],col="red")
# print(t)
```

---

**optSmacofSym_mMDS**  
*Selecting the optimal multidimensional scaling procedure - metric MDS*

**Description**

Selecting the optimal multidimensional scaling procedure by varying all combinations of normalization methods, distance measures, and metric MDS models

**Usage**

```r
optSmacofSym_mMDS(x,normalizations=NULL,distances=NULL,
mdsmodels=NULL,weights=NULL,spline.degrees=c(2),
outputCsv="",outputCsv2="",outDec=".",
stressDigits=6,HHIDigits=2,...)
```

**Arguments**

- `x` matrix or dataset
- `normalizations` optional, vector of normalization methods that should be used in procedure
- `distances` optional, vector of distance measures (manhattan, Euclidean, Chebyshev, squared Euclidean, GDM1) that should be used in procedure
- `mdsmodels` optional, vector of multidimensional models (ratio, interval, mspline) that should be used in procedure
- `spline.degrees` optional, vector (e.g. 2:4) of spline.degree parameter values that should be used in procedure for mspline model
weights	onoptional, variable weights used in distance calculation. Each weight takes value from interval $[0; 1]$ and sum of weights equals one
outputCsv
	optional, name of csv file with results
outputCsv2
	optional, name of csv (comma as decimal point sign) file with results
outDec
	decimal sign used in returned table
stressDigits
	Number of decimal digits for displaying Stress 1 value
HHIDigits
	Number of decimal digits for displaying HHI spp value
...
arguments passed to smacofSym, like ndim, itmax, eps and others

Details

Parameter normalizations may be the subset of the following values:
")n1","n2","n3","n3a","n4","n5","n5a","n6","n6a",
")n7","n8","n9","n9a","n10","n11","n12","n12a","n13"
(e.g. normalizations=c("n1","n2","n3","n5","n5a",
")n8","n9","n9a","n11","n12a"))
if normalizations is set to "n0" no normalization is applied

Parameter distances may be the subset of the following values:
")euclidean","manhattan","maximum","seuclidean","GDM1"
(e.g. distances=c("euclidean","manhattan"))

Parameter mdsmodels may be the subset of the following values (metric MDS):
")ratio","interval","mspline" (e.g. c("ratio","interval"))

Value

Data frame ordered by increasing value of Stress-1 fit measure with columns:

Normalization method
	normalization method used for p-th multidimensional scaling procedure
MDS model
	MDS model used for p-th multidimensional scaling procedure
Spline degree
	Additional spline.degree value if mspline model is used for simulation, for other models there is no value in this cell
Distance measure
	distance measure used for p-th multidimensional scaling procedure
STRESS 1
	value of Kruskal Stress-1 fit measure for p-th multidimensional scaling procedure
HHI spp
	Hirschman-Herfindahl HHI index calculated based on stress per point for p-th multidimensional scaling procedure

Author(s)

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/mdsOpt
References


Walesiak, M. (2016b), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonometria, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.


See Also

data.Normalization, dist.GDM, dist, smacofSym

Examples

print("uncomment to run - approximately 30 seconds runtime")
# uncomment to run - approximately 30 seconds runtime
# library(mdsOpt)
# metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
# metscale<-c("ratio","interval","mspline")
# metdist<-c("euclidean","manhattan","seuclidean","maximum","GDM1")
# data(data_lower_silesian)
# res<-optSmacofSym_mMDS(data_lower_silesian,normalizations=metnor,distances=metdist,
# mdsmodels=metscale, spline.degrees=c(2:3),outDec=".")
# stress<-as.numeric(gsub("",","",res[,"STRESS 1"],fixed=TRUE))
# hhi<-as.numeric(gsub("",","",res[,"HHI spp"],fixed=TRUE))
# cs<-(min(stress)+max(stress))/2 # critical stress
# t<-findOptimalSmacofSym(res,critical_stress=cs)
# print(t)
# plot(stress[-t$Nr],hhi[-t$Nr], xlab="Stress-1", ylab="HHI",type="n",font.lab=3)
# text(stress[-t$Nr],hhi[-t$Nr],labels=(1:nrow(res))[-t$Nr])
# abline(v=cs,col="red")
# points(stress[t$Nr],hhi[t$Nr], cex=5,col="red")
# text(stress[t$Nr],hhi[t$Nr],labels=(1:nrow(res))[t$Nr],col="red")

## optSmacofSym_nMDS

**Selecting the optimal multidimensional scaling procedure - nonmetric MDS**

### Description

Selecting the optimal multidimensional scaling procedure by varying all combinations of normalization methods and distance measures

### Usage

```r
optSmacofSym_nMDS(x, normalizations=NULL, distances=NULL, mdsmodels=c("ordinal"), weights=NULL, outputCsv="", outputCsv2="", outDec="", stressDigits=6, HHIDigits=2, ...)
```

### Arguments

- **x**: matrix or dataset
- **normalizations**: optional, vector of normalization methods that should be used in procedure
- **distances**: optional, vector of distance measures (manhattan, Euclidean, Chebyshew, squared Euclidean, GDM1) that should be used in procedure
- **mdsmodels**: "ordinal" (nonmetric MDS)
- **weights**: optional, variable weights used in distance calculation. Each weight takes value from interval [0; 1] and sum of weights equals one
- **outputCsv**: optional, name of csv file with results
- **outputCsv2**: optional, name of csv (comma as decimal point sign) file with results
- **outDec**: decimal sign used in returned table
- **stressDigits**: Number of decimal digits for displaying Stress 1 value
- **HHIDigits**: Number of decimal digits for displaying HHI spp value
- ... arguments passed to smacofSym

### Details

Parameter normalizations may be the subset of the following values:

"n1","n2","n3","n3a","n4","n5","n5a","n6","n6a",

"n7","n8","n9","n9a","n10","n11","n12","n12a","n13"

(e.g. normalizations=c("n1","n2","n3","n5","n5a",...)
"n8","n9","n9a","n11","n12a")
if normalizations is set to "n0" no normalization is applied
Parameter distances may be the subset of the following values:
"euclidean", "manhattan", "maximum", "seuclidean", "GDM1"
(e.g. distances=c("euclidean", "manhattan"))
Parameter mdsmodels "ordinal" MDS model (nonmetric MDS)

Value
Data frame ordered by increasing value of Stress-1 fit measure with columns:

Normalization method
normalization method used for p-th multidimensional scaling procedure
MDS model
"ordinal" MDS model (nonmetric MDS) for p-th multidimensional scaling procedure
Distance measure
distance measure used for p-th multidimensional scaling procedure
STRESS 1
value of Kruskal Stress-1 fit measure for p-th multidimensional scaling procedure
HHI spp
Hirschman-Herfindahl HHI index calculated based on stress per point for p-th multidimensional scaling procedure

Author(s)
Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/mdsOpt

References

Walesiak, M. (2016b), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonometria, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.


See Also
data.Normalization, dist.GDM, dist, smacofSym

Examples

```
print("uncomment to run - approximately 30 seconds runtime")

# uncomment to run - approximately 30 seconds runtime#
# library(mdsOpt)
# metnor<-c("n1","n2","n3","n5","n5a","n8","n9","n9a","n11","n12a")
# metscale<-"ordinal"
# metdist<-c("euclidean","manhattan","maximum","seuclidean","GDM1")
# data(data_lower_silesian)
# res<-optSmacofSym_nMDS(data_lower_silesian,normalizations=metnor,
# distances=metdist,mdsmodels=metscale)
# stress<-as.numeric(gsub("\",","",res[,]"STRESS 1"],fixed=TRUE))
# hhi<-as.numeric(gsub("\",","",res[,]"HHI spp"],fixed=TRUE))
# cs<-(min(stress)+max(stress))/2 # critical stress
# t<-findOptimalSmacofSym(res,critical_stress=cs)
# print(t)
# plot(stress[-t$Nr],hhi[-t$Nr], xlab="Stress-1", ylab="HHI",type="n",font.lab=3)
# text(stress[-t$Nr],hhi[-t$Nr],labels=(1:nrow(res))[-t$Nr])
# abline(v=cs,col="red")
# points(stress[t$Nr],hhi[t$Nr], cex=5,col="red")
# text(stress[t$Nr],hhi[t$Nr],labels=(1:nrow(res))[t$Nr],col="red")
```

rotation2dAnimation Creaes video by FFmpeg with animation of dataset rotated

Description

This function opens a graphics device to record the images produced in the code expr, then uses FFmpeg to convert these images to a video.
Usage

rotation2dAnimation(conf2d,
ani.interval=0.2,
ani.nmax=361,
ani.width=500,
ani.height=500,
ani.video.name="mds_rotate.mp4",
angle.start=-\pi,
angle.stop=\pi,
angle.step=\pi/180)

Arguments

conf2d two dimensional dataset or matrix
ani.video.name the file name of the output video (e.g. 'animation.mp4' or 'animation.avi')
ani.interval interval between animation frames
ani.nmax maximal number of frames
ani.width width of movie
ani.height height of movie
angle.start starting angle for animation
angle.stop end angle for animation
angle.step step of animation in radians

Details

This function uses \texttt{system} to call FFmpeg to convert the images to a single video. The command line used in this function is:

\texttt{ffmpeg -y -r \langle1/interval\rangle -i \langleimg.name\rangle%d.<ani.type> other.opts video.name}

where interval comes from \texttt{ani.options('interval')}, and \texttt{ani.type} is from \texttt{ani.options('ani.type')}.

For more details on the numerous options of FFmpeg, please see the reference.

Some Linux systems may use the alternate software 'avconv' instead of 'ffmpeg'. The package will attempt to determine which command is present and set \texttt{ani.options('ffmpeg')} to an appropriate default value. This can be overridden by passing in the \texttt{ffmpeg} argument.

Value

An integer indicating failure (-1) or success (0) of the converting (refer to \texttt{system}).

Author(s)

Marek Walesiak <marek.walesiak@ue.wroc.pl>, Andrzej Dudek <andrzej.dudek@ue.wroc.pl>
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland
http://keii.ue.wroc.pl/mdsOpt
rotation2dAnimation

References

Walesiak, M. (2016), Visualization of Linear Ordering Results for Metric Data with the Application of Multidimensional Scaling, Ekonomist, 2(52), 9-21. Available at: http://dx.doi.org/10.15611/ekt.2016.2.01.


http://ffmpeg.org/documentation.html

See Also

Other utilities: im.convert, saveGIF, saveHTML, saveLatex, saveSWF

Examples

#uncomment to run - approximate time 12 seconds
#library(mdsOpt)
#library(smacof)
#library(animation)
#library(spdep)
#library(clusterSim)
#data(data_lower_silesian)
#z<-data.Normalization(data_lower_silesian, type="n1")
#d<-dist.GDM(z, method="GDM1")
#res<-smacofSym(delta=d,ndim=2,type="interval")
#konf<-as.matrix(res$conf)
#rotation2dAnimation(conf2d=konf,angle.start=-0,angle.stop=2*pi)
Index

*Topic 2D rotation
  rotation2dAnimation, 21
*Topic I-stress
  findOptimalIscalInterval, 5
  optIscalInterval, 11
*Topic data set
  data_lower_silesian, 2
*Topic distance measures for interval-valued data
  findOptimalIscalInterval, 5
  optIscalInterval, 11
  optSmacofSymInterval, 13
*Topic distance measures
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
*Topic drawing isoquants
  drawIsoquants, 3
*Topic interval-valued data
  findOptimalIscalInterval, 5
  ispb, 9
  optIscalInterval, 11
  optSmacofSymInterval, 13
*Topic isoquants
  drawIsoquants, 3
*Topic metric MDS
  findOptimalIscalInterval, 5
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
  optIscalInterval, 11
  optSmacofSymInterval, 13
*Topic metric data
  findOptimalIscalInterval, 5
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
*Topic multidimensional scaling
  findOptimalIscalInterval, 5
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
  optIscalInterval, 11
  optSmacofSymInterval, 13
*Topic nonmetric MDS
  findOptimalIscalInterval, 5
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
  optIscalInterval, 11
  optSmacofSymInterval, 13
*Topic optimization methods
  findOptimalIscalInterval, 5
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
  optIscalInterval, 11
  optSmacofSymInterval, 13
*Topic stress per box
  findOptimalIscalInterval, 5
  ispb, 9
  optIscalInterval, 11
*Topic variable normalization methods
  findOptimalIscalInterval, 5
  findOptimalSmacofSym, 7
  findOptimalSmacofSym_mMDS, 16
  findOptimalSmacofSym_nMDS, 19
  optIscalInterval, 11
  optSmacofSymInterval, 13

ani.options, 22
data_Normalization, 6, 9, 10, 13, 15, 18, 21
data_lower_silesian, 2
dist, 9, 18, 21
dist_GDM, 9, 18, 21
dist_Symbolic, 15
drawIsoquants, 3

findOptimalIscalInterval, 5
findOptimalSmacofSym, 7
im_convert, 23
IMDS, 6, 10, 13
interval_normalization, 6, 10, 13, 15
ispb, 9
INDEX

optIscalInterval, 11
optSmacofSym_nMDS, 7, 16
optSmacofSym_nMDS, 5, 7, 19
optSmacofSymInterval, 13
rotation2dAnimation, 21

saveGIF, 23
saveHTML, 23
saveLatex, 23
saveSWF, 23
smacofSym, 9, 15, 18, 21
system, 22