Package 'MCI'

January 20, 2025

Type Package

Title Multiplicative Competitive Interaction (MCI) Model

Version 1.3.3

Date 2017-10-10

Author Thomas Wieland

Maintainer Thomas Wieland <thomas.wieland.geo@googlemail.com>

Description

Market area models are used to analyze and predict store choices and market areas concerning retail and service locations. This package implements two market area models (Huff Model, Multiplicative Competitive Interaction Model) into R, while the emphases lie on 1.) fitting these models based on empirical data via OLS regression and nonlinear techniques and 2.) data preparation and processing (esp. interaction matrices and data preparation for the MCI Model).

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-10 16:55:06 UTC

Contents

MCI-package   *Multiplicative Competitive Interaction (MCI) Model*

Description

The *Huff model* (Huff 1962, 1963, 1964) is the most popular *spatial interaction model* for retailing
and services and belongs to the family of *probabilistic market area models*. The basic idea of the
model, derived from the *Luce choice axiom*, is that consumer decisions are not deterministic but
probabilistic. Thus, the decision of customers for a shopping location in a competitive environ-
ment cannot be predicted exactly. The results of the model are probabilities for these decisions
(*interaction probabilities*), which can be interpreted as market shares of the regarded locations (j)
in the customer origins (i), p_{ij} (*local market shares*). The model results can be regarded as an
equilibrium solution (consumer equilibrium) with logically consistent market shares ($0 < p_{ij} < 1$,
$\sum_{j=1}^{n} p_{ij} = 1$). From a theoretical perspective, the model is based on an utility function with two
explanatory variables ("attraction" of the locations, transport costs between origins and locations),
which are weighted by an exponent: $U_{ij} = A_j^{\gamma} d_{ij}^{-\lambda}$. The probability is calculated as the utility
quotient: $p_{ij} = U_{ij} / \sum_{j=1}^{n} U_{ij}$. The *distance decay function* reflecting the disutility of transport
costs can also be exponential or logistic. The model can also be used for the estimation of market
areas based on location sales or total patronage using nonlinear optimization algorithms. When the
"real" local market shares were observed, the model can be parametrized by using the *Multiplicative
Competitive Interaction (MCI) Model*.

The *Multiplicative Competitive Interaction (MCI) Model* (Nakanishi/Cooper 1974, 1982) is an
econometric model for analyzing market shares and/or market areas in a competitive environment

where the market is divided in i submarkets (e.g. groups of customers, time periods or geographical regions) and served by j suppliers (e.g. firms, brands or locations). The dependent variable of the model is p_{ij}, the market shares of j in i, which are *logically consistent* ($0 < p_{ij} < 1$, $\sum_{j=1}^{n} p_{ij} = 1$). The market shares depend on the attraction/utility of the alternative j in the choice situation/submarket i, A_{ij} or U_{ij}. The model is nonlinear (multiplicative attractivity/utility function with exponential weighting) but can be transformed to be estimated by OLS (ordinary least squares) regression using the multi-step *log-centering transformation*. Before the *log-centering transformation* can be applied, which is required for fitting the model, also a re-arrangement of the raw data (e.g. household surveys) in an *interaction matrix* is necessary. An *interaction matrix* is a special case of table where every row is an i x j combination and the market shares of j in i (p_{ij}) are saved in a new column (Linear table, the opposite of crosstable). The MCI model is a special case of market share model (which fulfills the requirement of logical consistency in the output), but can especially be used as a market area model (or *spatial MCI model*) in retail location analysis since it is an econometric approach to estimate the parameters of the *Huff model* mentioned above.

The functions in this package include fitting the MCI model, MCI shares simulations, the *log-centering transformation* of MCI datasets, creation of *interaction matrices* from empirical raw data and several tools for data preparation. Additionally, the package provides applications for the *Huff model*, including a nonlinear optimization algorithm to estimate market areas on condition that total market areas (customers, sales) of the stores/locations are known.

Author(s)

Thomas Wieland

Maintainer: Thomas Wieland <thomas.wieland.geo@googlemail.com>

References

Cooper, L. G./Nakanishi, M. (2010): "Market-Share Analysis: Evaluating competitive marketing effectiveness". Boston, Dordrecht, London : Kluwer (first published 1988). E-book version from 2010: http://www.anderson.ucla.edu/faculty/lee.cooper/MCI_Book/BOOKI2010.pdf

Cliquet, G. (2006): "Retail Location Models". In: Cliquet, G. (ed.): *Geomarketing. Models and Strategies in Spatial Marketing*. London : ISTE. p. 137-163.

Guessefeldt, J. (2002): "Zur Modellierung von raeumlichen Kaufkraftstroemen in unvollkommenen Maerkten". In: *Erdkunde*, **56**, 4, p. 351-370.

Huff, D. L. (1962): "Determination of Intra-Urban Retail Trade Areas". Los Angeles : University of California.

Huff, D. L. (1963): "A Probabilistic Analysis of Shopping Center Trade Areas". In: *Land Economics*, **39**, 1, p. 81-90.

Huff, D. L. (1964): "Defining and Estimating a Trading Area". In: *Journal of Marketing*, **28**, 4, p. 34-38.

Huff, D. L./Batsell, R. R. (1975): "Conceptual and Operational Problems with Market Share Models of Consumer Spatial Behavior". In: *Advances in Consumer Research*, **2**, p. 165-172.

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Loeffler, G. (1998): "Market areas - a methodological reflection on their boundaries". In: *GeoJournal*, **45**, 4, p. 265-272.

Nakanishi, M./Cooper, L. G. (1974): "Parameter Estimation for a Multiplicative Competitive Interaction Model - Least Squares Approach". In: *Journal of Marketing Research*, **11**, 3, p. 303-311.

Nakanishi, M./Cooper, L. G. (1982): "Simplified Estimation Procedures for MCI Models". In: *Marketing Science*, **1**, 3, p. 314-322.

Wieland, T. (2013): "Einkaufsstaettenwahl, Einzelhandelscluster und raeumliche Versorgungsdisparitaeten - Modellierung von Marktgebieten im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten". In: Schrenk, M./Popovich, V./Zeile, P./Elisei, P. (eds.): REAL CORP 2013. Planning Times. Proceedings of 18th International Conference on Urban Planning, Regional Development and Information Society. Schwechat. p. 275-284. `http://www.corp.at/archive/CORP2013_98.pdf`

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. `http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf`

DIY1   *Distance matrix for DIY stores*

Description

Preliminary stage of an interaction matrix: Distance matrix for 19 origins and six DIY (do-it-yourself) stores (i = 19 submarkets x j = 6 suppliers) in a German research area.

Usage

```
data("DIY1")
```

Format

A data frame with 114 observations on the following 3 variables.

i_origin a factor with 19 levels representing the origins

j_destination a factor with six levels representing the DIY stores

t_ij_min a numeric vector containing the travel time (in minutes) from the origins to the stores

Source

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

References

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

Examples

```
data(DIY1)
data(DIY2)
data(DIY3)
# Loading the three DIY store datasets

DIY_alldata <- merge (DIY1, DIY2, by.x = "j_destination", by.y = "j_destination")
# Add store data to distance matrix

huff_DIY <- huff.shares (DIY_alldata, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, lambda = -2)
# Calculating Huff local market shares
# Gamma = 1, Lambda = -2

huff_DIY <- merge (huff_DIY, DIY3, by.x = "i_origin", by.y = "district")
# Add data for origins

huff_DIY_total <- shares.total (huff_DIY, "i_origin", "j_destination", "p_ij",
"population")
# Calculating total market areas (=sums of customers)

colnames(DIY3) <- c("district", "pop")
# Change column name to "pop" (must be other name)

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10)
# Iterative search for the best lambda value using bisection
# Output: gamma and lambda

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
```

```
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10, output = "iterations", show_proc = TRUE)
# Same procedure, output: single iterations

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "compare", iterations = 10, output = "iterations", show_proc = TRUE, plotVal = TRUE)
# Using compare method, output: single iterations and plot
```

DIY2   *DIY store information*

Description

The six DIY stores in a German research area, their corresponding DIY chain and sales area.

Usage

```
data("DIY2")
```

Format

A data frame with 6 observations on the following 3 variables.

j_destination a factor with six levels representing the DIY stores

j_chain a factor with five levels containing the store chain

A_j_salesarea_sqm a numeric vector for the sales area of the DIY stores in sqm

Source

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellana-lytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fall-beispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handels-forschung*, **23**. 289 pages. Mannheim : MetaGIS.

References

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellana-lytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fall-beispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handels-forschung*, **23**. 289 pages. Mannheim : MetaGIS.

Examples

```
data(DIY1)
data(DIY2)
data(DIY3)
# Loading the three DIY store datasets

DIY_alldata <- merge (DIY1, DIY2, by.x = "j_destination", by.y = "j_destination")
# Add store data to distance matrix

huff_DIY <- huff.shares (DIY_alldata, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, lambda = -2)
# Calculating Huff local market shares
# Gamma = 1, Lambda = -2

huff_DIY <- merge (huff_DIY, DIY3, by.x = "i_origin", by.y = "district")
# Add data for origins

huff_DIY_total <- shares.total (huff_DIY, "i_origin", "j_destination", "p_ij",
"population")
# Calculating total market areas (=sums of customers)

colnames(DIY3) <- c("district", "pop")
# Change column name to "pop" (must be other name)

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10)
# Iterative search for the best lambda value using bisection
# Output: gamma and lambda

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10, output = "iterations", show_proc = TRUE)
# Same procedure, output: single iterations

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "compare", iterations = 10, output = "iterations", show_proc = TRUE, plotVal = TRUE)
# Using compare method, output: single iterations and plot
```

DIY3 *Data for origins (DIY store customers' places of residence)*

Description

The 19 origins and the resident population.

Usage

```
data("DIY3")
```

Format

A data frame with 19 observations on the following 2 variables.

district a factor with 19 levels representing the origins

population a numeric vector containing the resident population (2012)

Source

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellana-lytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fall-beispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handels-forschung*, **23**. 289 pages. Mannheim : MetaGIS.

References

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellana-lytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fall-beispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handels-forschung*, **23**. 289 pages. Mannheim : MetaGIS.

Examples

```
data(DIY1)
data(DIY2)
data(DIY3)
# Loading the three DIY store datasets

DIY_alldata <- merge (DIY1, DIY2, by.x = "j_destination", by.y = "j_destination")
# Add store data to distance matrix

huff_DIY <- huff.shares (DIY_alldata, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, lambda = -2)
# Calculating Huff local market shares
# Gamma = 1, Lambda = -2

huff_DIY <- merge (huff_DIY, DIY3, by.x = "i_origin", by.y = "district")
# Add data for origins

huff_DIY_total <- shares.total (huff_DIY, "i_origin", "j_destination", "p_ij",
"population")
# Calculating total market areas (=sums of customers)
```

```
colnames(DIY3) <- c("district", "pop")
# Change column name to "pop" (must be other name)

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10)
# Iterative search for the best lambda value using bisection
# Output: gamma and lambda

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10, output = "iterations", show_proc = TRUE)
# Same procedure, output: single iterations

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "compare", iterations = 10, output = "iterations", show_proc = TRUE, plotVal = TRUE)
# Using compare method, output: single iterations and plot
```

Freiburg1	*Distance matrix for grocery stores in Freiburg*

Description

Preliminary stage of an interaction matrix: Distance matrix for all statistical 42 districts and all 63 grocery stores (i = 42 submarkets x j = 63 suppliers) in Freiburg (Germany) including the size of the grocery stores.

Usage

```
data("Freiburg1")
```

Format

A data frame with 2646 observations on the following 4 variables.

district a numeric vector representing the 42 statistical districts of Freiburg

store a numeric vector identifying the store code of the mentioned grocery store in the study area

salesarea a numeric vector for the sales area of the grocery stores in sqm

distance a numeric vector for the distance from the places of residence (statistical districts) to the grocery stores in km

Source

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

References

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

Examples

```
data(Freiburg1)
data(Freiburg2)
data(Freiburg3)
# Loads the data

huff_mat <- huff.shares (Freiburg1, "district", "store", "salesarea", "distance")
# Market area estimation using the Huff Model with standard parameters
# (gamma = 1, lambda = -2)

huff_mat_pp <- merge (huff_mat, Freiburg2)
# Adding the purchasing power data for the city districts

huff_total <- shares.total (huff_mat_pp, "district", "store", "p_ij", "ppower")
# Total expected sales and shares

huff_total_control <- merge (huff_total, Freiburg3, by.x = "suppliers_single",
by.y = "store")

model.fit(huff_total_control$annualsales, huff_total_control$sum_E_j, plotVal = TRUE)
```

Freiburg2 *Statistical districts of Freiburg*

Description

The 42 statistical districts of Freiburg (Germany) and the estimated annual purchasing power for groceries, based on average expenditures and population.

Usage

```
data("Freiburg2")
```

Format

A data frame with 42 observations on the following 2 variables.

district a numeric vector representing the 42 statistical districts of Freiburg

ppower a numeric vector containing the estimated absolute value of annual purchasing power for groceries in the district in EUR

Source

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

References

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

Examples

```
data(Freiburg1)
data(Freiburg2)
data(Freiburg3)
# Loads the data

huff_mat <- huff.shares (Freiburg1, "district", "store", "salesarea", "distance")
# Market area estimation using the Huff Model with standard parameters
# (gamma = 1, lambda = -2)

huff_mat_pp <- merge (huff_mat, Freiburg2)
# Adding the purchasing power data for the city districts

huff_total <- shares.total (huff_mat_pp, "district", "store", "p_ij", "ppower")
# Total expected sales and shares

huff_total_control <- merge (huff_total, Freiburg3, by.x = "suppliers_single",
by.y = "store")

model.fit(huff_total_control$annualsales, huff_total_control$sum_E_j, plotVal = TRUE)
```

Freiburg3   *Grocery stores in Freiburg*

Description

The 63 grocery stores in Freiburg (Germany) and the estimated annual sales in EUR.

Usage

```
data("Freiburg3")
```

Format

A data frame with 63 observations on the following 2 variables.

store a numeric vector identifying the store code of the mentioned grocery store in the study area

annualsales a numeric vector containing the estimated annual sales of the store in EUR

Source

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebens-
mitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen
des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht.
Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goet-
tingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

References

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebens-
mitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen
des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht.
Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goet-
tingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

Examples

```
data(Freiburg1)
data(Freiburg2)
data(Freiburg3)
# Loads the data

huff_mat <- huff.shares (Freiburg1, "district", "store", "salesarea", "distance")
# Market area estimation using the Huff Model with standard parameters
# (gamma = 1, lambda = -2)

huff_mat_pp <- merge (huff_mat, Freiburg2)
# Adding the purchasing power data for the city districts

huff_total <- shares.total (huff_mat_pp, "district", "store", "p_ij", "ppower")
```

```
# Total expected sales and shares

huff_total_control <- merge (huff_total, Freiburg3, by.x = "suppliers_single",
by.y = "store")

model.fit(huff_total_control$annualsales, huff_total_control$sum_E_j, plotVal = TRUE)
```

---

geom   *Geometric mean*

Description

Computes the geometric mean of a numeric vector.

Usage

```
geom(x)
```

Arguments

x   A numeric vector

Value

The value of the geometric mean.

Author(s)

Thomas Wieland

Examples

```
numvec <- c(10,15,20,25,30)
# Creates a numeric vector "numvec"
mean(numvec)
# Mean of numvec
geom(numvec)
# Geometric mean of numvec
```

---

grocery1   *Grocery store choices in Goettingen*

Description

Results from a POS survey in Goettingen (Germany) from June 2015 (raw data). Amongst other things, the participants were asked about their last grocery shopping trip (store choice and expenditures) and their place of residence (ZIP code). The survey dataset contains 179 cases/interviewed individuals. The survey is *not* representative and should be regarded as an example.

Usage

```
data("grocery1")
```

Format

A data frame with 179 observations on the following 5 variables.

interview_nr a numeric vector, interview/individual identifier

store_code a factor with 32 levels (ALDI1, ALDI3, ..., EDEKA1, ... REWE1, ...), identifying the store code of the mentioned grocery store in the study area, data from Wieland (2011)

store_chain a factor with 11 levels (Aldi, Edeka, Kaufland, ...) for the store chain of the grocery stores in the study area, data from Wieland (2011)

trip_expen a numeric vector containing the individual trip expenditures at the last visited grocery store

plz_submarket a factor with 7 levels (PLZ_37073, PLZ_37075, ...) representing the individuals' place of residence based on the five-digit ZIP codes in the study area

Source

Wieland, T. (2011): "Nahversorgung mit Lebensmitteln in Goettingen 2011 - Eine Analyse der Angebotssituation im Goettinger Lebensmitteleinzelhandel unter besonderer Beruecksichtigung der Versorgungsqualitaet". *Goettinger Statistik Aktuell*, **35**. Goettingen. http://www.goesis.goettingen.de/pdf/Aktuell35.pdf

Primary empirical sources: POS (point of sale) survey in the authors' course ("Seminar Angewandte Geographie 1: Stadtentwicklung und Citymarketing an einem konkreten Fallbeispiel", University of Goettingen/Institute of Geography, June 2015), own calculations

See Also

[grocery2](#)

Examples

```
data(grocery1)
# Loads the data
ijmatrix.create (grocery1, "plz_submarket", "store_code")
# Creates an interaction table with local market shares
```

grocery2	*Grocery store market areas in Goettingen*

Description

Market areas of grocery stores in Goettingen, generated from a POS survey in Goettingen (Germany) from June 2015. The survey dataset contains 224 cases ($i = 7$ submarkets x $j = 32$ suppliers). The data is the result of a survey that is *not* representative (see *grocery1*) and also biased due to the data preparation. The data should be regarded as an example.

Usage

```
data("grocery2")
```

Format

A data frame with 224 observations on the following 8 variables.

plz_submarket a factor with 7 levels (PLZ_37073, PLZ_37075, ...) representing the submarkets (places of residence based on the five-digit ZIP codes) in the study area

store_code a factor with 32 levels (ALDI1, ALDI3, ..., EDEKA1, ... REWE1, ...), identifying the store code of the mentioned grocery store in the study area, data from Wieland (2011)

store_chain a factor with 11 levels (Aldi, Edeka, ..., Kaufland, ...) for the store chain of the grocery stores in the study area, data from Wieland (2011)

store_type a factor with 3 levels for the store type (Biosup = bio-supermarkt, Disc = discounter, Sup = supermarket)

salesarea_qm a numeric vector for the sales area of the grocery stores in sqm, data from Wieland (2011)

pricelevel_euro a numeric vector for the price level of the grocery chain (standardized basket in EUR), based on the data from DISQ (2015)

dist_km a numeric vector for the distance from the places of residence (ZIP codes) to the grocery stores in km

p_ij_obs a numeric vector for the empirically observed (and corrected) market shares (p_{ij}) of the stores in the submarkets

Source

DISQ (Deutsches Institut fuer Servicequalitaet) (2015) "Discounter guenstig, Vollsortimenter serviceorientiert. Studie Lebensmittelmaerkte (15.10.2015)". http://disq.de/2015/20151015-Lebensmittelmaerkte.html

Wieland, T. (2011): "Nahversorgung mit Lebensmitteln in Goettingen 2011 - Eine Analyse der Angebotssituation im Goettinger Lebensmitteleinzelhandel unter besonderer Beruecksichtigung der Versorgungsqualitaet". *Goettinger Statistik Aktuell*, **35**. Goettingen. http://www.goesis.goettingen.de/pdf/Aktuell35.pdf

Primary empirical sources: POS (point of sale) survey in the authors' course ("Seminar Angewandte Geographie 1: Stadtentwicklung und Citymarketing an einem konkreten Fallbeispiel", University of Goettingen/Institute of Geography, June 2015), own calculations

See Also

grocery1

Examples

```
data(grocery2)
# Loads the data
mci.transmat (grocery2, "plz_submarket", "store_code", "p_ij_obs", "dist_km", "salesarea_qm")
# Applies the log-centering transformation to the dataset using the function mci.transmat
```

---

huff.attrac   *Local optimization of attraction values in the Huff Model*

Description

This function optimizes the attraction values of suppliers/location in a given Huff interaction matrix to fit empirically observed total values (e.g. annual sales) and calculates market shares/market areas

Usage

```
huff.attrac(huffdataset, origins, locations, attrac, dist,
lambda = -2, dtype = "pow", lambda2 = NULL,
localmarket_dataset, origin_id, localmarket,
location_dataset, location_id, location_total,
tolerance = 5, output = "matrix", show_proc = FALSE,
check_df = TRUE)
```

Arguments

huffdataset	an interaction matrix which is a data.frame containing the origins, locations and the explanatory variables (attraction, transport costs)
origins	the column in the interaction matrix huffdataset containing the origins (e.g. ZIP codes)
locations	the column in the interaction matrix huffdataset containing the locations (e.g. store codes)
attrac	the column in the interaction matrix huffdataset containing the attraction variable (e.g. sales area)
dist	the column in the interaction matrix huffdataset containing the transport costs (e.g. travelling time or street distance)
lambda	a single numeric value of λ for the (exponential) weighting of distance (transport costs, default: -2)
dtype	Type of distance weighting function: "pow" (power function), "exp" (exponential function) or "logistic" (logistic function) (default: dtype = "pow")
lambda2	if dtype = "logistic" a second λ parameter is needed

localmarket_dataset

> A `data.frame` containing the origins saved in a column which has the same name as in `huffdataset` and another column containing the local market potential

origin_id   the column in the dataset `localmarket_dataset` containing the origins (e.g. statistical districts, ZIP codes)

localmarket the column in the dataset `localmarket_dataset` containing the local market potential (e.g. purchasing power, number of customers)

location_dataset

> A `data.frame` containing the suppliers/locations and their observed total values

location_id the column in the dataset `location_dataset` containing the locations (e.g. store codes), j, according to the codes in `huffdataset`

location_total the column in the dataset `location_dataset` containing the observed total values of suppliers/locations, $T_{j,obs}$ (e.g. annual sales, total number of customers)

tolerance   accepted value of absolute percentage error between observed ($T_{j,obs}$) and modeled total values ($T_{j,exp}$) to skip a local optimization of location/supplier j

output   Type of function output: `output = "matrix"` returns a Huff interaction matrix with the optimized attraction values and the expected market shares/market areas. If `output = "total"`, the old (observed) and the new (expected) total values are returned. If `output = "attrac"`, the optimized attraction values are returned.

show_proc   logical argument that indicates if the function prints messages about the state of process during the work (e.g. "Processing variable xyz ..." or "Variable xyz is regarded as dummy variable"). Default: `show_proc = FALSE` (messages off)

check_df   logical argument that indicates if the given dataset is checked for correct input, only for internal use, should not be deselected (default: `TRUE`)

Details

In many cases, only total empirical values of the suppliers/locations can be used for market area estimation. This function fits the *Huff model* not by estimating the parameters but by optimizing the attraction variable (transport cost weighting by λ is given) using an optimization algorithm based on the idea of the *local optimization of attraction* algorithm developed by Guessefeldt (2002) and other model fit approaches. This function consists of a single optimization of every supplier/location. Note that the best results can be achieved by repeating the algorithm while evaluating the results (see the function `huff.fit()`, which extends this algorithm to a given number of iterations).

Value

The function output can be controlled by the function argument `output`. If `output = "matrix"` the function returns a Huff interaction matrix with the optimized attraction values and the expected market shares/market areas. If `output = "total"`, the old (observed) and the new (expected) total values are returned. If `output = "attrac"`, the optimized attraction values are returned. All results are `data.frame`.

Author(s)

Thomas Wieland

References

Guessefeldt, J. (2002): "Zur Modellierung von raeumlichen Kaufkraftstroemen in unvollkommenen Maerkten". In: *Erdkunde*, **56**, 4, p. 351-370.

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebens-mitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

See Also

huff.fit, huff.shares, huff.decay

Examples

```
data(Freiburg1)
data(Freiburg2)
data(Freiburg3)
# Loading the three Freiburg datasets

# NOTE: This may take a while!
# huff.attrac(Freiburg1, "district", "store", "salesarea", "distance", lambda = -2, dtype= "pow",
# lambda2 = NULL, Freiburg2, "district", "ppower", Freiburg3, "store", "annualsales",
# tolerance = 5, output = "total")
# Local optimization of store attraction using the function huff.attrac()
# returns a data frame with total values (observed and expected after optimization)
```

---

huff.decay   *Distance decay function in the Huff model*

Description

This function estimates a distance decay function from observed data and compares different function types

Usage

```
huff.decay(dataset, x, y, plotFunc = TRUE)
```

Arguments

dataset	A data.frame containing the observed interaction data and the transport costs
x	A numeric vector containing the independent variable, the transport costs (e.g. traveling time or street distance)
y	A numeric vector containing the dependent variable, the interaction measure (e.g. local market shares, trip volume, visitors per capita)
plotFunc	logical argument that indicates if the curves are plotted (default: plotFunc = TRUE)

Details

The *distance decay function* is a classic concept in quantitative economic geography and describes the relationship between transport costs and trip volume between origins (i) and a destination (j). The dependent variable is an indicator of trip volume, such as local market shares or visitors per capita etc., which are explained by the transport costs between all i and the destination j, d_{ij}.

The non-linear modeling of transport costs is a key concept of the *Huff model* (see the function `huff.shares`). This function estimates and compares different types of possible *distance decay functions* (linear, power, exponential, logistic) based on observed interaction data.

Value

A `data.frame` containing the function parameters (`Intercept`, `Slope`), their p values in the regression function (`p Intercept`, `p Slope`) and fitting measures (`R-Squared`, `Adj. R-Squared`). Optionally, a plot of the four estimated functions and the observed data.

Author(s)

Thomas Wieland

References

Huff, D. L. (1962): "Determination of Intra-Urban Retail Trade Areas". Los Angeles : University of California.

Huff, D. L. (1963): "A Probabilistic Analysis of Shopping Center Trade Areas". In: *Land Economics*, **39**, 1, p. 81-90.

Huff, D. L. (1964): "Defining and Estimating a Trading Area". In: *Journal of Marketing*, **28**, 4, p. 34-38.

Isard, W. (1960): "Methods of Regional Analysis: an Introduction to Regional Science". Cambridge.

Kanhaeusser, C. (2007): "Modellierung und Prognose von Marktgebieten am Beispiel des Moebeleinzelhandels". In: Klein, R./Rauh, J. (eds.): *Analysemethodik und Modellierung in der geographischen Handelsforschung*. *Geographische Handelsforschung*, **13**. Passau. p. 75-110.

Loeffler, G. (1998): "Market areas - a methodological reflection on their boundaries". In: *GeoJournal*, **45**, 4, p. 265-272.

See Also

huff.shares, huff.attrac, huff.fit, mci.fit

Examples

```
# Market area analysis based on the POS survey in shopping1 #

data(shopping1)
# The survey dataset
data(shopping2)
# Dataset with distances and travel times
```

```
shopping1_adj <- shopping1[(shopping1$weekday != 3) & (shopping1$holiday != 1)
& (shopping1$survey != "pretest"),]
# Removing every case from tuesday, holidays and the ones belonging to the pretest

ijmatrix_POS <- ijmatrix.create(shopping1_adj, "resid_code", "POS", "POS_expen")
# Creates an interaction matrix based on the observed frequencies (automatically)
# and the POS expenditures (Variable "POS_expen" separately stated)

ijmatrix_POS_data <- merge(ijmatrix_POS, shopping2, by.x="interaction", by.y="route",
all.x = TRUE)
# Adding the distances and travel times

ijmatrix_POS_data$freq_ij_abs_cor <- var.correct(ijmatrix_POS_data$freq_ij_abs,
corr.mode = "inc", incby = 0.1)
# Correcting the absolute values (frequencies) by increasing by 0.1

data(shopping3)
ijmatrix_POS_data_residdata <- merge(ijmatrix_POS_data, shopping3)
# Adding the information about the origins (places of residence) stored in shopping3

ijmatrix_POS_data_residdata$visitper1000 <- (ijmatrix_POS_data_residdata$
freq_ij_abs_cor/ijmatrix_POS_data_residdata$resid_pop2015)*1000
# Calculating the dependent variable
# visitper1000: surveyed customers per 1.000 inhabitants of the origin

ijmatrix_POS_data_residdata <-
ijmatrix_POS_data_residdata[(!is.na(ijmatrix_POS_data_residdata$
visitper1000)) & (!is.na(ijmatrix_POS_data_residdata$d_time)),]
# Removing NAs (data for some outlier origins and routes not available)

ijmatrix_POS_data_residdata_POS1 <-
ijmatrix_POS_data_residdata[ijmatrix_POS_data_residdata$POS=="POS1",]
# Dataset for POS1 (town centre)

ijmatrix_POS_data_residdata_POS2 <-
ijmatrix_POS_data_residdata[ijmatrix_POS_data_residdata$POS=="POS2",]
# Dataset for POS2 (out-of-town shopping centre)

huff.decay(ijmatrix_POS_data_residdata_POS1, "d_km", "visitper1000")
huff.decay(ijmatrix_POS_data_residdata_POS1, "d_time", "visitper1000")
huff.decay(ijmatrix_POS_data_residdata_POS2, "d_km", "visitper1000")
huff.decay(ijmatrix_POS_data_residdata_POS2, "d_time", "visitper1000")
```

---

huff.fit   *Fitting the Huff model using local optimization of attractivity*

Description

This function fits the Huff model with a given interaction matrix by optimizing the attractivity values
of suppliers/locations iteratively and calculates the market shares/market areas

Usage

```
huff.fit(huffdataset, origins, locations, attrac, dist, lambda = -2, dtype = "pow",
lambda2 = NULL, localmarket_dataset, origin_id, localmarket, location_dataset,
location_id, location_total, tolerance = 5, iterations = 3, output = "total",
show_proc = FALSE, check_df = TRUE)
```

Arguments

huffdataset	an interaction matrix which is a data.frame containing the origins, locations and the explanatory variables (attraction, transport costs)
origins	the column in the interaction matrix huffdataset containing the origins (e.g. ZIP codes)
locations	the column in the interaction matrix huffdataset containing the locations (e.g. store codes)
attrac	the column in the interaction matrix huffdataset containing the attraction variable (e.g. sales area)
dist	the column in the interaction matrix huffdataset containing the transport costs (e.g. travelling time or street distance)
lambda	a single numeric value of λ for the (exponential) weighting of distance (transport costs, default: -2)
dtype	Type of distance weighting function: "pow" (power function), "exp" (exponential function) or "logistic" (logistic function) (default: dtype = "pow")
lambda2	if dtype = "logistic" a second λ parameter is needed
localmarket_dataset	
	A data.frame containing the origins saved in a column which has the same name as in huffdataset and another column containing the local market potential
origin_id	the column in the dataset localmarket_dataset containing the origins (e.g. statistical districts, ZIP codes)
localmarket	the column in the dataset localmarket_dataset containing the local market potential (e.g. purchasing power, number of customers)
location_dataset	
	A data.frame containing the suppliers/locations and their observed total values
location_id	the column in the dataset location_dataset containing the locations (e.g. store codes), j, according to the codes in huffdataset
location_total	the column in the dataset location_dataset containing the observed total values of suppliers/locations, $T_{j,obs}$ (e.g. annual sales, total number of customers)
tolerance	accepted value of absolute percentage error between observed ($T_{j,obs}$) and modeled total values ($T_{j,exp}$) to skip a local optimization of location/supplier j
iterations	a single numeric value for the desired number of iterations
output	Type of function output: output = "matrix" returns a Huff interaction matrix with the optimized attractivity values and the expected market shares/market areas. If output = "total", the old (observed) and the new (expected) total values are returned. If output = "diag", the function returns the diagnosis results (fitting measures)

show_proc	logical argument that indicates if the function prints messages about the state of process during the work (e.g. "Processing variable xyz ..." or "Variable xyz is regarded as dummy variable"). Default: show_proc = FALSE (messages off)
check_df	logical argument that indicates if the given dataset is checked for correct input, only for internal use, should not be deselected (default: TRUE)

Details

In many cases, only total empirical values of the suppliers/locations can be used for market area estimation. This function fits the *Huff model* not by estimating the parameters but by optimizing the attraction variable (transport cost weighting by λ is given) using an optimization algorithm based on the idea of the *local optimization of attraction* algorithm developed by Guessefeldt (2002) and other model fit approaches. The fitting process in the huff.fit includes of given number of (m) iterations, while the fit gets better with every iteration. The algorithm results can be evaluated by several diagnosis criteria which have been frequently used to evaluate *Huff model* results: Besides the sum of squared residuals, the function also calculates a *Pseudo-R-squared* measure and the *MAPE (mean average percentage error)*, both used by De Beule et al. (2014), and the *global error* used by Klein (1988).

Value

The function output can be controlled by the function argument output. If output = "matrix" the function returns a Huff interaction matrix with the optimized attractivity values and the expected market shares/market areas. If output = "total", the old (observed) and the new (expected) total values are returned. If output = "diag", the diagnosis results (fitting measures) are returned. All results are data.frame.

Note

Note that the iterations can be time-consuming and depend on the number of suppliers/locations. Use show_proc = TRUE for monitoring the iteration process.

Author(s)

Thomas Wieland

References

De Beule, M./Van den Poel, D./Van de Weghe, N. (2014): "An extended Huff-model for robustly benchmarking and predicting retail network performance". In: *Applied Geography*, **46**, 1, p. 80-89.

Guessefeldt, J. (2002): "Zur Modellierung von raeumlichen Kaufkraftstroemen in unvollkommenen Maerkten". In: *Erdkunde*, **56**, 4, p. 351-370.

Klein, R. (1988): "Der Lebensmittel-Einzelhandel im Raum Verden. Raeumliches Einkaufsverhalten unter sich wandelnden Bedingungen". *Flensburger Arbeitspapiere zur Landeskunde und Raumordnung*, **6**. Flensburg.

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht.

Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. <http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf>

See Also

[huff.attrac](), [huff.shares](), [huff.decay]()

Examples

```
data(Freiburg1)
data(Freiburg2)
data(Freiburg3)
# Loading the three Freiburg datasets

# NOTE: This may take a while!
# huff_total_opt2 <- huff.fit(Freiburg1, "district", "store", "salesarea", "distance",
# lambda = -2, dtype= "pow", lambda2 = NULL, Freiburg2, "district", "ppower",
# Freiburg3, "store", "annualsales", tolerance = 1, iterations = 2, output = "total",
# show_proc = TRUE)
# 2 iterations of the optimization algorithm with an accepted difference of +/- 1 %
# Output of total sales/shares, stored in dataset huff_total_opt10

# model.fit(huff_total_opt2$total_obs, huff_total_opt2$sum_E_j, plotVal = TRUE)
# total_obs = observed total values, originally from dataset Freiburg3
# sum_E_j = expected total values
```

---

huff.lambda	*Fitting the distance parameter lambda in the Huff model*

Description

This function estimates a distance decay parameter from observed total store/location data (e.g. complete annual turnovers) using bisection or "trial and error"

Usage

```
huff.lambda(huffdataset, origins, locations, attrac, dist, gamma = 1, atype = "pow",
gamma2 = NULL, lambda_startv = -1, lambda_endv = -3, dtype = "pow",
localmarket_dataset, origin_id, localmarket,
location_dataset, location_id, location_total,
method = "bisection", iterations = 10, output = "matrix",
plotVal = FALSE, show_proc = FALSE, check_df = TRUE)
```

Arguments

huffdataset	an interaction matrix which is a data.frame containing the origins, locations and the explanatory variables (attraction, transport costs)
origins	the column in the interaction matrix huffdataset containing the origins (e.g. ZIP codes)

locations	the column in the interaction matrix huffdataset containing the locations (e.g. store codes)
attrac	the column in the interaction matrix huffdataset containing the attraction variable (e.g. sales area)
dist	the column in the interaction matrix huffdataset containing the transport costs (e.g. travelling time or street distance)
gamma	a single numeric value of γ for the exponential weighting of the attraction variable (default: 1)
atype	Type of attraction weighting function: atype = ″pow″ (power function), atype = ″exp″ (exponential function) or atype = ″logistic″ (default: atype = ″pow″)
gamma2	if atype = ″logistic″ a second γ parameter is needed
lambda_startv	Start value for λ search
lambda_endv	End value for λ search
dtype	Type of distance weighting function: dtype = ″pow″ (power function), dtype = ″exp″ (exponential function) or dtype = ″logistic″ (default: dtype = ″pow″)
localmarket_dataset	
	A data.frame containing the origins saved in a column which has the same name as in huffdataset and another column containing the local market potential
origin_id	the column in the dataset localmarket_dataset containing the origins (e.g. statistical districts, ZIP codes)
localmarket	the column in the dataset localmarket_dataset containing the local market potential (e.g. purchasing power, number of customers)
location_dataset	
	A data.frame containing the suppliers/locations and their observed total values
location_id	the column in the dataset location_dataset containing the locations (e.g. store codes), j, according to the codes in huffdataset
location_total	the column in the dataset location_dataset containing the observed total values of suppliers/locations, $T_{j,obs}$ (e.g. annual sales, total number of customers)
method	If method = ″bisection″ (default value), the optimal λ is found by bisection. If method = ″compare″, several values are tried and the best one is found ("trial and error")
iterations	a single numeric value for the desired number of iterations
output	If output = ″iterations″, the results for every single iteration is shown. If output = ″total″, total sales and market shares (or total market area) of the suppliers are shown.
plotVal	If plotVal = TRUE, the function shows a simple plot of λ values against residuals
show_proc	logical argument that indicates if the function prints messages about the state of process during the work (e.g. "Processing variable xyz ..."). Default: show_proc = FALSE (messages off)
check_df	logical argument that indicates if the given dataset is checked for correct input, only for internal use, should not be deselected (default: TRUE)

Details

In many cases, only total empirical values of the suppliers/locations (e.g. annual turnover) can be used for market area estimation. This function fits the *Huff model* by estimating the λ parameter iteratively using an optimization algorithm based on the idea of Klein (1988). The fitting process in the huff.lambda includes of given number of iterations, while the fit gets better with every iteration, measured using the sum of squared residuals of observed vs. expected total values. The iterative optimization can be done via bisection (see Kaw et al. 2011, ch. 03.03) or "trial and error" (see Fuelop et al. 2011).

Value

The function output can be controlled by the function argument output. If output = "iterations", the results for every single iteration is shown (data.frame). If output = "total", total sales and market shares (or total market area) of the suppliers are shown (data.frame). The default output is a list with γ and λ.

Note

Note that the iterations can be time-consuming and depend on the number of suppliers/locations. Use show_proc = TRUE for monitoring the iteration process.

Author(s)

Thomas Wieland

References

Fuelop, G./Kopetsch, T./Schoepe, P. (2011): "Catchment areas of medical practices and the role played by geographical distance in the patient's choice of doctor". In: *The Annals of Regional Science*, **46**, 3, p. 691-706.

Kaw, A. K./Kalu, E. E./Nguyen, D. (2011): "Numerical Methods with Applications". http://nm.mathforcollege.com/topics/textbook_index.html

Klein, R. (1988): "Der Lebensmittel-Einzelhandel im Raum Verden. Raeumliches Einkaufsverhalten unter sich wandelnden Bedingungen". *Flensburger Arbeitspapiere zur Landeskunde und Raumordnung*, **6**. Flensburg.

See Also

huff.attrac, huff.shares, huff.decay, huff.fit

Examples

```
data(DIY1)
data(DIY2)
data(DIY3)
# Loading the three DIY store datasets

DIY_alldata <- merge (DIY1, DIY2, by.x = "j_destination", by.y = "j_destination")
# Add store data to distance matrix
```

```
huff_DIY <- huff.shares (DIY_alldata, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, lambda = -2)
# Calculating Huff local market shares
# Gamma = 1, Lambda = -2

huff_DIY <- merge (huff_DIY, DIY3, by.x = "i_origin", by.y = "district")
# Add data for origins

huff_DIY_total <- shares.total (huff_DIY, "i_origin", "j_destination", "p_ij",
"population")
# Calculating total market areas (=sums of customers)

colnames(DIY3) <- c("district", "pop")
# Change column name to "pop" (must be other name)

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10)
# Iterative search for the best lambda value using bisection
# Output: gamma and lambda

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "bisection", iterations = 10, output = "iterations", show_proc = TRUE)
# Same procedure, output: single iterations

huff.lambda (huff_DIY, "i_origin", "j_destination", "A_j_salesarea_sqm",
"t_ij_min", gamma = 1, atype = "pow", gamma2 = NULL,
lambda_startv = -1, lambda_endv = -2.5, dtype= "pow",
DIY3, "district", "pop", huff_DIY_total, "suppliers_single", "sum_E_j",
method = "compare", iterations = 10, output = "iterations", show_proc = TRUE, plotVal = TRUE)
# Using compare method, output: single iterations and plot
```

---

huff.shares   *Huff model market share/market area simulations*

Description

Calculating market areas/local market shares using the probabilistic market area model by Huff

Usage

```
huff.shares(huffdataset, origins, locations, attrac, dist, gamma = 1, lambda = -2,
atype = "pow", dtype = "pow", gamma2 = NULL, lambda2 = NULL, check_df = TRUE)
```

Arguments

huffdataset	an interaction matrix which is a data.frame containing the origins, locations and the explanatory variables (attraction, transport costs)
origins	the column in the interaction matrix huffdataset containing the origins (e.g. ZIP codes)
locations	the column in the interaction matrix huffdataset containing the locations (e.g. store codes)
attrac	the column in the interaction matrix huffdataset containing the attraction variable (e.g. sales area)
dist	the column in the interaction matrix huffdataset containing the transport costs (e.g. travelling time or street distance)
gamma	a single numeric value of γ for the exponential weighting of the attraction variable (default: 1)
lambda	a single numeric value of λ for the (exponential) weighting of distance (transport costs, default: -2)
atype	Type of attraction weighting function: atype = "pow" (power function), atype = "exp" (exponential function) or atype = "logistic" (default: atype = "pow")
dtype	Type of distance weighting function: dtype = "pow" (power function), dtype = "exp" (exponential function) or dtype = "logistic" (default: dtype = "pow")
gamma2	if atype = "logistic" a second γ parameter is needed
lambda2	if dtype = "logistic" a second λ parameter is needed
check_df	logical argument that indicates if the given dataset is checked for correct input, only for internal use, should not be deselected (default: TRUE)

Details

This function computes the market shares from a given interaction matrix and given weighting parameters. The result matrix can be processed by the function shares.total() to calculate the total values (e.g. annual sales) and shares.

Value

Returns the input interaction matrix including the calculated shares (p_ij) as data.frame.

Author(s)

Thomas Wieland

References

Huff, D. L. (1962): "Determination of Intra-Urban Retail Trade Areas". Los Angeles : University of California.

Huff, D. L. (1963): "A Probabilistic Analysis of Shopping Center Trade Areas". In: *Land Economics*, **39**, 1, p. 81-90.

Huff, D. L. (1964): "Defining and Estimating a Trading Area". In: *Journal of Marketing*, **28**, 4, p. 34-38.

Loeffler, G. (1998): "Market areas - a methodological reflection on their boundaries". In: *GeoJournal*, **45**, 4, p. 265-272.

Wieland, T. (2015): "Nahversorgung im Kontext raumoekonomischer Entwicklungen im Lebensmitteleinzelhandel - Konzeption und Durchfuehrung einer GIS-gestuetzten Analyse der Strukturen des Lebensmitteleinzelhandels und der Nahversorgung in Freiburg im Breisgau". Projektbericht. Goettingen : GOEDOC, Dokumenten- und Publikationsserver der Georg-August-Universitaet Goettingen. http://webdoc.sub.gwdg.de/pub/mon/2015/5-wieland.pdf

See Also

huff.attrac, huff.fit, huff.decay

Examples

```
data(Freiburg1)
data(Freiburg2)
# Loads the data

huff.shares (Freiburg1, "district", "store", "salesarea", "distance")
# Standard weighting (power function with gamma=1 and lambda=-2)
```

---

ijmatrix.create   *Interaction matrix with market shares*

Description

Creation of an interaction matrix with market shares (p_{ij}) of every supplier (j) in every submarket (i) based on the frequencies in the raw data (e.g. household or POS survey).

Usage

```
ijmatrix.create(rawdataset, submarkets, suppliers, ..., remNA = TRUE,
remSing = FALSE, remSing.val = 1, remSingSupp.val = 1,
correctVar = FALSE, correctVar.val = 1)
```

Arguments

rawdataset	a data.frame containing the raw data
submarkets	the column in the dataset containing the submarkets (e.g. ZIP codes)
suppliers	the column in the dataset containing the suppliers (e.g. store codes)
...	other numeric variables in the raw data which were observed and shall be used to calculate market shares (e.g. expenditures)
remNA	logical argument that indicates if NA values of the submarkets and suppliers are removed or not (default: remNA = TRUE, otherwise: remNA = FALSE, which means that NA are considered in the interaction matrix, recoded as factor 99999)

remSing	logical argument that indicates if singular instances of the submarkets and suppliers are removed or not (default: remSing = FALSE)
remSing.val	if remSing = TRUE: numeric value that indicates the limit value of singular instances in the submarket set to be removed from the data
remSingSupp.val	
	if remSing = TRUE: numeric value that indicates the limit value of singular instances in the supplier set to be removed from the data
correctVar	logical argument that indicates if the calculated market shares shall be corrected when they do not match the MCI standards ($p_{ij} = 0$) (default: correctVar = FALSE)
correctVar.val	if correctVar = TRUE: numeric value which is added to the absolute values before calculating market shares (default: 1)

Details

This function creates an interaction matrix for all i submarkets (e.g. geographical regions) and all j suppliers (e.g. store locations). The function calculates p_{ij} based on the frequencies and, optionally, further market shares calculated from other observed variables in the given raw dataset (e.g. expenditures from submarket i at supplier j).

Single observations with missing submarket or supplier (NA) are removed from the data automatically (unless remNA = FALSE). Optionally, singular instances (e.g. some submarkets or suppliers are only represented once or twice in the whole dataset) can also be removed (remSing = TRUE), where the limit values for extraction can be set by remSing.val and remSingSupp.val (e.g. remSing.val = 2 and remSingSupp.val = 1 removes every submarket from the interaction matrix which was observed ≤ 2 and every supplier observed ≤ 1).

Value

An interaction matrix which is a data.frame containing the i x j combinations ('interaction'), the submarkets (column is named as in raw data), the suppliers (column is named as in raw data), the observed absolute frequencies of every j in every i ('freq_ij_abs'), the observed absolute frequencies in every i ('freq_i_total') and the observed market shares of every j in every i ('p_ij_obs'). If additional variables are stated (e.g. expenditures) which shall be turned into (local) market shares, the output interaction matrix contains absolute values for every interaction, total values for every i submarket and market shares (p_{ij}) for these variables, too, which are automatically named based on the given variable name (e.g. the market shares based on a raw data variable called expen is named p_ij_obs_expen). The first three variables of the output matrix are factors, the calculated values are numeric.

Author(s)

Thomas Wieland

References

Cooper, L. G./Nakanishi, M. (2010): "Market-Share Analysis: Evaluating competitive marketing effectiveness". Boston, Dordrecht, London : Kluwer (first published 1988). E-book version from 2010: http://www.anderson.ucla.edu/faculty/lee.cooper/MCI_Book/BOOKI2010.pdf

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

ijmatrix.shares, ijmatrix.crosstab

Examples

```
# Creating an interaction matrix based on the POS survey in grocery1 #

data(grocery1)
# Loads the data
ijmatrix.create (grocery1, "plz_submarket", "store_code")
# Creates an interaction matrix with local market shares based on frequencies

mynewmcidata <- ijmatrix.create (grocery1, "plz_submarket", "store_code")
# Save results directly in a new dataset

ijmatrix.create (grocery1, "plz_submarket", "store_code", "trip_expen")
# Creates an interaction matrix with local market shares based on frequencies
# and expenditures (Variable "trip_expen")



# MCI analysis for the grocery store market areas based on the POS survey in shopping1 #

data(shopping1)
# Loading the survey dataset
data(shopping2)
# Loading the distance/travel time dataset
data(shopping3)
# Loading the dataset containing information about the city districts
data(shopping4)
# Loading the grocery store data

shopping1_KAeast <- shopping1[shopping1$resid_code %in%
shopping3$resid_code[shopping3$KA_east == 1],]
# Extracting only inhabitants of the eastern districts of Karlsruhe

ijmatrix_gro_adj <- ijmatrix.create(shopping1_KAeast, "resid_code",
"gro_purchase_code", "gro_purchase_expen", remSing = TRUE, remSing.val = 1,
remSingSupp.val = 2, correctVar = TRUE, correctVar.val = 0.1)
# Removing singular instances/outliers (remSing = TRUE) incorporating
# only suppliers which are at least obtained three times (remSingSupp.val = 2)
# Correcting the values (correctVar = TRUE)
```

```
# by adding 0.1 to the absolute values (correctVar.val = 0.1)

ijmatrix_gro_adj <- ijmatrix_gro_adj[(ijmatrix_gro_adj$gro_purchase_code !=
"REFORMHAUSBOESER") & (ijmatrix_gro_adj$gro_purchase_code != "WMARKT_DURLACH")
& (ijmatrix_gro_adj$gro_purchase_code != "X_INCOMPLETE_STORE"),]
# Remove non-regarded observations

ijmatrix_gro_adj_dist <- merge (ijmatrix_gro_adj, shopping2, by.x="interaction",
by.y="route")
# Include the distances and travel times (shopping2)
ijmatrix_gro_adj_dist_stores <- merge (ijmatrix_gro_adj_dist, shopping4,
by.x = "gro_purchase_code", by.y = "location_code")
# Adding the store information (shopping4)

mci.transvar(ijmatrix_gro_adj_dist_stores, "resid_code", "gro_purchase_code",
"p_ij_obs")
# Log-centering transformation of one variable (p_ij_obs)

ijmatrix_gro_transf <- mci.transmat(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# Log-centering transformation of the interaction matrix

mcimodel_gro_trips <- mci.fit(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# MCI model for the grocery store market areas
# shares: "p_ij_obs", explanatory variables: "d_time", "salesarea_qm"

summary(mcimodel_gro_trips)
# Use like lm
```

---

ijmatrix.crosstab   *Converting interaction matrix with market shares to crosstable*

Description

This function converts a given interaction matrix with (local) market shares to a crosstable where
the rows are the submarkets and the columns contain the market shares

Usage

```
ijmatrix.crosstab(mcidataset, submarkets, suppliers, shares)
```

Arguments

mcidataset	The interaction matrix containing the submarkets/origins, suppliers/locations and local market shares
submarkets	the column in the dataset containing the submarkets (e.g. ZIP codes)
suppliers	the column in the dataset containing the suppliers (e.g. store codes)
shares	the column in the dataset containing the local market shares

Details

In many cases the results of a market area analysis shall be visualized in a map, e.g. by pie charts or contour lines which belongs to the standard map types in Geographical Information Systems (GIS). An interaction matrix can not be processed directly in a GIS due to its linear character. This function converts an interaction matrix into a special kind of crosstable where the rows contain the origins i and the local market shares p_{ij} are represented by the columns. The submarkets/origins ID (rows) can be joined directly to the geodata (e.g. point shapefile) while the columns can be used for visualization.

Value

A `data.frame` containing i rows and $j+1$ columns (suppliers/locations and one column containing the submarkets/origins).

Author(s)

Thomas Wieland

References

Cooper, L. G./Nakanishi, M. (2010): "Market-Share Analysis: Evaluating competitive marketing effectiveness". Boston, Dordrecht, London : Kluwer (first published 1988). E-book version from 2010: http://www.anderson.ucla.edu/faculty/lee.cooper/MCI_Book/BOOKI2010.pdf

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

ijmatrix.create, ijmatrix.shares

Examples

```
data(grocery2)
# Loads the data

grocery2_cross <- ijmatrix.crosstab(grocery2, "plz_submarket", "store_code", "p_ij_obs")
# Converts the market shares in the grocery2 dataset to a crosstable
```

---

ijmatrix.shares *Market shares in interaction matrix*

Description

Calculating market shares in an interaction matrix based on the observations of the regarded variable.

Usage

```
ijmatrix.shares(rawmatrix, submarkets, suppliers, observations,
varname_total = "freq_i_total", varname_shares = "p_ij_obs")
```

Arguments

rawmatrix	a data.frame containing the submarkets, suppliers and the observed data
submarkets	the column in the dataset containing the submarkets (e.g. ZIP codes)
suppliers	the column in the dataset containing the suppliers (e.g. store codes)
observations	the column with the regarded variable (e.g. frequencies, expenditures, turnovers)
varname_total	character value, name of the variable for the total absolute values of the i submarkets in the output (default: varname_total = "freq_i_total")
varname_shares	character value, name of the variable for the market shares p_{ij} in the output (default: varname_shares = "p_ij_obs")

Details

This function calculates the market shares of every j in every i (p_{ij}) based on an existing interaction matrix.

Value

The input interaction matrix which is a data.frame with a new column 'p_ij_obs' (or another stated name in the argument varname_shares) or, if used after ijmatrix.create, an update of the columns 'freq_i_total' and 'p_ij_obs' (or different stated names in the arguments varname_total and/or varname_shares).

Author(s)

Thomas Wieland

References

Cooper, L. G./Nakanishi, M. (2010): "Market-Share Analysis: Evaluating competitive marketing effectiveness". Boston, Dordrecht, London : Kluwer (first published 1988). E-book version from 2010: http://www.anderson.ucla.edu/faculty/lee.cooper/MCI_Book/BOOKI2010.pdf

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

ijmatrix.create

Examples

```
data(grocery1)
# Loads the data

mymcidata <- ijmatrix.create (grocery1, "plz_submarket", "store_code")
# Creates an interaction matrix with market shares based on the frequencies
# of visited grocery stores and saves results directly in a new dataset
mymcidata$freq_ij_corr <- var.correct(mymcidata$freq_ij_abs, 1)
# Corrects the frequency variable (no zero or negative values allowed)
mymcidata_shares <- ijmatrix.shares(mymcidata, "plz_submarket", "store_code", "freq_ij_corr")
# Calculates market shares based on the corrected frequencies
# and saves the results as a new dataset
```

---

lm.beta   *Beta regression coefficients*

Description

Calculating the standardized (beta) regression coefficients of linear models

Usage

```
lm.beta(linmod, dummy.na = TRUE)
```

Arguments

linmod	A lm object (linear regression model) with more than one independent variable
dummy.na	logical argument that indicates if dummy variables should be ignored when calculating the beta weights (default: TRUE). Note that beta weights of dummy variables do not make any sense

Details

Standardized coefficients (beta coefficients) show how many standard deviations a dependent variable will change when the regarded independent variable is increased by a standard deviation. The β values are used in multiple linear regression models to compare the real effect (power) of the independent variables when they are measured in different units. Note that β values do not make any sense for dummy variables since they cannot change by a standard deviation.

Value

A `list` containing all independent variables and the corresponding standardized coefficients.

Author(s)

Thomas Wieland

References

Backhaus, K./Erichson, B./Plinke, W./Weiber, R. (2016): "Multivariate Analysemethoden: Eine anwendungsorientierte Einfuehrung". Berlin: Springer.

Examples

```
x1 <- runif(100)
x2 <- runif(100)
# random values for two independent variables (x1, x2)
y <- runif(100)
# random values for the dependent variable (y)
testmodel <- lm(y~x1+x2)
# OLS regression
summary(testmodel)
# summary
lm.beta(testmodel)
# beta coefficients
```

---

mci.fit   *Fitting the MCI model*

Description

This function fits the MCI model based on a given MCI interaction matrix.

Usage

```
mci.fit(mcidataset, submarkets, suppliers, shares, ..., origin = TRUE, show_proc = FALSE)
```

Arguments

mcidataset	an interaction matrix which is a data.frame containing the submarkets, suppliers, the observed market shares (p_{ij}) and the explanatory variables
submarkets	the column in the interaction matrix mcidataset containing the submarkets
suppliers	the column in the interaction matrix mcidataset containing the suppliers
shares	the column in the interaction matrix mcidataset containing the observed market shares (p_{ij}), numeric
...	the column(s) of the explanatory variable(s) (at least one), numeric and positive (or dummy [1,0])
origin	logical argument that indicates if an intercept is included in the model or it is a regression through the origin (default origin = TRUE)
show_proc	logical argument that indicates if the function prints messages about the state of process during the work (e.g. "Processing variable xyz ..." or "Variable xyz is regarded as dummy variable"). Default: show_proc = FALSE (messages off)

Details

The function transforms the input dataset (MCI interaction matrix) to regression-ready-data with the *log-centering transformation* by Nakanishi/Cooper (1974) and, therefore, the data is fitted by a linear regression model. The return of the function mci.fit() can be treated exactly like the output of the lm() function. The column in the interaction matrix mcidataset containing the shares is the 4th parameter of the function (shares). The further arguments (...) are the columns with the explanatory variables (attractivity/utility values of the j alternatives, characteristics of the i submarkets). The function identifies dummy variables which are not transformed (because they do not have to be). Normally, in MCI analyzes no intercept is included into the transformed linear model due to the requirement of logically consistent market shares as model results (see above), so the default is a regression through the origin (origin = TRUE). Note: If an intercept is included (origin = FALSE) (and also if dummy variables are used as explanatories), the *inverse log-centering transformation* by Nakanishi/Cooper (1982) has to be used for simulations.

Value

The function mci.fit() returns an object of class lm. The full information (estimates, significance, R-squared etc.) can be adressed by the function summary(). The explanatory variables are marked with a "_t" to indicate that they were transformed by log-centering transformation.

Author(s)

Thomas Wieland

References

Colome Perales, R. (2002): "Consumer Choice in Competitive Location Models". Barcelona.

Gonzalez-Benito, O./Greatorex, M./Munos-Gallego, P. A. (2000): "Assessment of potential retail segmentation variables - An approach based on a subjective MCI resource allocation model". In: *Journal of Retailing and Consumer Services*, **7**, 3, p. 171-179.

Hartmann, M. (2005): "Gravitationsmodelle als Verfahren der Standortanalyse im Einzelhandel". *Statistik Regional Electronic Papers*, **02/2005**. Halle.

Huff, D. L./Batsell, R. R. (1975): "Conceptual and Operational Problems with Market Share Models of Consumer Spatial Behavior". In: *Advances in Consumer Research*, **2**, p. 165-172.

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Nakanishi, M./Cooper, L. G. (1974): "Parameter Estimation for a Multiplicative Competitive Interaction Model - Least Squares Approach". In: *Journal of Marketing Research*, **11**, 3, p. 303-311.

Nakanishi, M./Cooper, L. G. (1982): "Simplified Estimation Procedures for MCI Models". In: *Marketing Science*, **1**, 3, p. 314-322.

Suarez-Vega, R./Gutierrez-Acuna, J. L./Rodriguez-Diaz, M. (2015): "Locating a supermarket using a locally calibrated Huff model". In: *International Journal of Geographical Information Science*, **29**, 2, p. 217-233.

Tihi, B./Oruc, N. (2012): "Competitive Location Assessment - the MCI Approach". In: *South East European Journal of Economics and Business*, **7**, 2, p. 35-49.

Wieland, T. (2013): "Einkaufsstaettenwahl, Einzelhandelscluster und raeumliche Versorgungsdisparitaeten - Modellierung von Marktgebieten im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten". In: Schrenk, M./Popovich, V./Zeile, P./Elisei, P. (eds.): REAL CORP 2013. Planning Times. Proceedings of 18th International Conference on Urban Planning, Regional Development and Information Society. Schwechat. p. 275-284. http://www.corp.at/archive/CORP2013_98.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

[mci.transmat](), [mci.transvar](), [mci.shares]()

Examples

```
# MCI analysis for the grocery store market areas based on the POS survey in shopping1 #

data(shopping1)
# Loading the survey dataset
data(shopping2)
# Loading the distance/travel time dataset
data(shopping3)
# Loading the dataset containing information about the city districts
data(shopping4)
# Loading the grocery store data

shopping1_KAeast <- shopping1[shopping1$resid_code %in%
shopping3$resid_code[shopping3$KA_east == 1],]
# Extracting only inhabitants of the eastern districts of Karlsruhe
```

```
ijmatrix_gro_adj <- ijmatrix.create(shopping1_KAeast, "resid_code",
"gro_purchase_code", "gro_purchase_expen", remSing = TRUE, remSing.val = 1,
remSingSupp.val = 2, correctVar = TRUE, correctVar.val = 0.1)
# Removing singular instances/outliers (remSing = TRUE) incorporating
# only suppliers which are at least obtained three times (remSingSupp.val = 2)
# Correcting the values (correctVar = TRUE)
# by adding 0.1 to the absolute values (correctVar.val = 0.1)

ijmatrix_gro_adj <- ijmatrix_gro_adj[(ijmatrix_gro_adj$gro_purchase_code !=
"REFORMHAUSBOESER") & (ijmatrix_gro_adj$gro_purchase_code != "WMARKT_DURLACH")
& (ijmatrix_gro_adj$gro_purchase_code != "X_INCOMPLETE_STORE"),]
# Remove non-regarded observations

ijmatrix_gro_adj_dist <- merge (ijmatrix_gro_adj, shopping2, by.x="interaction",
by.y="route")
# Include the distances and travel times (shopping2)
ijmatrix_gro_adj_dist_stores <- merge (ijmatrix_gro_adj_dist, shopping4,
by.x = "gro_purchase_code", by.y = "location_code")
# Adding the store information (shopping4)

mci.transvar(ijmatrix_gro_adj_dist_stores, "resid_code", "gro_purchase_code",
"p_ij_obs")
# Log-centering transformation of one variable (p_ij_obs)

ijmatrix_gro_transf <- mci.transmat(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# Log-centering transformation of the interaction matrix

mcimodel_gro_trips <- mci.fit(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# MCI model for the grocery store market areas
# shares: "p_ij_obs", explanatory variables: "d_time", "salesarea_qm"

summary(mcimodel_gro_trips)
# Use like lm
```

---

mci.shares	*MCI market share/market area simulations*

Description

This function calculates (local) market shares based on specified explanatory variables and their weighting parameters in a given MCI interaction matrix.

Usage

```
mci.shares(mcidataset, submarkets, suppliers, ..., mcitrans = "lc", interc = NULL)
```

Arguments

mcidataset	an interaction matrix which is a data.frame containing the submarkets, suppliers and the explanatory variables
submarkets	the column in the interaction matrix mcidataset containing the submarkets
suppliers	the column in the interaction matrix mcidataset containing the suppliers
...	the column(s) of the explanatory variable(s) (at least one), numeric and positive (or dummy [1,0]), and their weighting parameter(s). The parameter(s) must follow the particular variable(s): mcivariable1, parameter1, ...
mcitrans	defines if the regular multiplicative formula is used or the *inverse log-centering transformation* where the explanatory variables are MCI-transformed and linked by addition in an exponential function instead of multiplication. This transformation is necessary if an intercept is included in the model and/or if dummy variables are used as explanatories (default: mcitrans = "lc", which indicates the regular log-centering transformation)
interc	if mcitrans = "ilc": logical argument that indicates if an intercept is included in the model (default interc = NULL)

Details

In this function, the input dataset (MCI interaction matrix) is used for a calculation of (local) market shares (p_{ij}), based on (at least one) given explanatory variable(s) and (a) given weighting parameter(s). If an intercept is included in the model and/or if dummy variables are used as explanatories, the *inverse log-centering transformation* by Nakanishi/Cooper (1982) has to be used for simulations (mcitrans = "ilc").

Value

The function mci.shares() returns the input interaction matrix (data.frame) with new variables/columns, where the last one (p_ij) is the one of interest, containing the (local) market shares of the j suppliers in the i submarkets (p_{ij}).

Author(s)

Thomas Wieland

References

Huff, D. L./Batsell, R. R. (1975): "Conceptual and Operational Problems with Market Share Models of Consumer Spatial Behavior". In: *Advances in Consumer Research*, **2**, p. 165-172.

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Nakanishi, M./Cooper, L. G. (1974): "Parameter Estimation for a Multiplicative Competitive Interaction Model - Least Squares Approach". In: *Journal of Marketing Research*, **11**, 3, p. 303-311.

Nakanishi, M./Cooper, L. G. (1982): "Simplified Estimation Procedures for MCI Models". In: *Marketing Science*, **1**, 3, p. 314-322.

Wieland, T. (2013): "Einkaufsstaettenwahl, Einzelhandelscluster und raeumliche Versorgungsdisparitaeten - Modellierung von Marktgebieten im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten". In: Schrenk, M./Popovich, V./Zeile, P./Elisei, P. (eds.): REAL CORP 2013. Planning Times. Proceedings of 18th International Conference on Urban Planning, Regional Development and Information Society. Schwechat. p. 275-284. http://www.corp.at/archive/CORP2013_98.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

[mci.fit](), [mci.transmat](), [mci.transvar](), [shares.total]()

Examples

```
data(Freiburg1)
data(Freiburg2)
# Loads the data

mynewmatrix <- mci.shares(Freiburg1, "district", "store", "salesarea", 1, "distance", -2)
# Calculating shares based on two attractivity/utility variables

mynewmatrix_alldata <- merge(mynewmatrix, Freiburg2)
# Merge interaction matrix with district data (purchasing power)

shares.total (mynewmatrix_alldata, "district", "store", "p_ij", "ppower")
# Calculation of total sales
```

---

mci.shares.elast   *Market share elasticities*

Description

This function calculates the market share elasticities (point elasticity) with respect to an attraction/utility variable and its given weighting parameter.

Usage

```
mci.shares.elast (mcidataset, submarkets, suppliers, shares, mcivar, mciparam,
check_df = TRUE)
```

Arguments

mcidataset	an interaction matrix which is a data.frame containing the submarkets, suppliers, the market shares and a variable for the local market potential (e.g. purchasing power, number of customers, population)
submarkets	the column in the interaction matrix mcidataset containing the submarkets
suppliers	the column in the interaction matrix mcidataset containing the suppliers
shares	the column in the interaction matrix mcidataset containing the (local) market shares
mcivar	the column in the interaction matrix mcidataset containing the attraction/utility variable
mciparam	single value of the (empirically estimated) weighting parameter corresponding to the attraction/utility variable
check_df	logical argument that indicates if the input (dataset, column names) is checked (default: check_df = TRUE (should not be changed, only for internal use))

Details

Market-share elasticity is defined as the ratio of the relative change in the market share corresponding to a relative change in an explanatory (attraction/utility) variable, such as price or, in the context of retailing, driving time, sales area or price level. The elasticities calculated here are point elasticities (not arc elasticities): $e_{s_i} = (dp_i/dX_{ki}) * (X_{ki}/p_i)$, which are calculated for the MCI model via: $e_{s_i} = \beta_k * (1 - p_i)$, where β_k is the corresponding weighting parameter. If the (absolute) elasticity value is greater than one, the suppliers' market share is called elastic, if it is smaller than one, the share is unelastic. E.g. if the share elasticity of a products' price is -2, a relative price reduction of 5% results in a share increase of 10% (Cooper/Nakanishi 2010). Note that the elasticity depends on the empirical shares: The greater the actual share, the smaller is the elasticity.

Value

The function mci.shares.elast() returns the input interaction matrix (data.frame) with a new column containing the calculated share elasticities for every combination of i and j.

Author(s)

Thomas Wieland

References

Cooper, L. G./Nakanishi, M. (2010): "Market-Share Analysis: Evaluating competitive marketing effectiveness". Boston, Dordrecht, London : Kluwer (first published 1988). E-book version from 2010: http://www.anderson.ucla.edu/faculty/lee.cooper/MCI_Book/BOOKI2010.pdf

See Also

mci.fit, mci.shares

Examples

```
# MCI analysis for the grocery store market areas based on the POS survey in shopping1 #

data(shopping1)
# Loading the survey dataset
data(shopping2)
# Loading the distance/travel time dataset
data(shopping3)
# Loading the dataset containing information about the city districts
data(shopping4)
# Loading the grocery store data

shopping1_KAeast <- shopping1[shopping1$resid_code %in%
shopping3$resid_code[shopping3$KA_east == 1],]
# Extracting only inhabitants of the eastern districts of Karlsruhe

ijmatrix_gro_adj <- ijmatrix.create(shopping1_KAeast, "resid_code",
"gro_purchase_code", "gro_purchase_expen", remSing = TRUE, remSing.val = 1,
remSingSupp.val = 2, correctVar = TRUE, correctVar.val = 0.1)
# Removing singular instances/outliers (remSing = TRUE) incorporating
# only suppliers which are at least obtained three times (remSingSupp.val = 2)
# Correcting the values (correctVar = TRUE)
# by adding 0.1 to the absolute values (correctVar.val = 0.1)

ijmatrix_gro_adj <- ijmatrix_gro_adj[(ijmatrix_gro_adj$gro_purchase_code !=
"REFORMHAUSBOESER") & (ijmatrix_gro_adj$gro_purchase_code != "WMARKT_DURLACH")
& (ijmatrix_gro_adj$gro_purchase_code != "X_INCOMPLETE_STORE"),]
# Remove non-regarded observations

ijmatrix_gro_adj_dist <- merge (ijmatrix_gro_adj, shopping2, by.x="interaction",
by.y="route")
# Include the distances and travel times (shopping2)
ijmatrix_gro_adj_dist_stores <- merge (ijmatrix_gro_adj_dist, shopping4,
by.x = "gro_purchase_code", by.y = "location_code")
# Adding the store information (shopping4)

mci.transvar(ijmatrix_gro_adj_dist_stores, "resid_code", "gro_purchase_code",
"p_ij_obs")
# Log-centering transformation of one variable (p_ij_obs)

ijmatrix_gro_transf <- mci.transmat(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# Log-centering transformation of the interaction matrix

mcimodel_gro_trips <- mci.fit(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# MCI model for the grocery store market areas
# shares: "p_ij_obs", explanatory variables: "d_time", "salesarea_qm"

summary(mcimodel_gro_trips)
# Use like lm
```

```
# Calculating market share elasticity:
ijmatrix_gro_adj_dist_stores_elas1 <- mci.shares.elast (ijmatrix_gro_adj_dist_stores,
"resid_code", "gro_purchase_code", "p_ij_obs", "d_time", -1.2443)
# Share elasticities of driving time

ijmatrix_gro_adj_dist_stores_elas2 <- mci.shares.elast (ijmatrix_gro_adj_dist_stores,
"resid_code", "gro_purchase_code", "p_ij_obs", "salesarea_qm", 0.9413)
# Share elasticities of sales area of the stores
```

---

mci.transmat	*Log-centering transformation of an MCI interaction matrix*

Description

This function applies the *log-centering transformation* to the variables in a given MCI interaction matrix.

Usage

```
mci.transmat(mcidataset, submarkets, suppliers, mcivariable1, ..., show_proc = FALSE)
```

Arguments

mcidataset	an interaction matrix which is a data.frame containing the submarkets, suppliers and the regarded variables (e.g. the observed market shares, p_{ij}, and the explanatory variables)
submarkets	the column in the interaction matrix mcidataset containing the submarkets
suppliers	the column in the interaction matrix mcidataset containing the suppliers
mcivariable1	the column of the first variable to be transformed, numeric and positive (or dummy [1,0])
...	the columns of other variables to be transformed, numeric and positive (or dummy [1,0])
show_proc	logical argument that indicates if the function prints messages about the state of process during the work (e.g. "Processing variable xyz ..." or "Variable xyz is regarded as dummy variable"). Default: show_proc = FALSE (messages off)

Details

This function transformes the input dataset (MCI interaction matrix) to regression-ready data with the *log-centering transformation* by Nakanishi/Cooper (1974). The resulting data.frame can be fitted with the lm() function (to combine these two steps in one, use mci.fit()). The *log-centering transformation* can be regarded as the key concept of the MCI model because it enables the model to be estimated by OLS (ordinary least squares) regression. The function identifies dummy variables which are not transformed (because they do not have to be).

Value

Returns a new `data.frame` with regression-ready data where the input variables are transformed by the the *log-centering transformation*. The names of the input variables are passed to the new `data.frame` marked with a "_t" to indicate that they were transformed (e.g. "shares_t" is the transformation of "shares").

Author(s)

Thomas Wieland

References

Huff, D. L./Batsell, R. R. (1975): "Conceptual and Operational Problems with Market Share Models of Consumer Spatial Behavior". In: *Advances in Consumer Research*, **2**, p. 165-172.

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Nakanishi, M./Cooper, L. G. (1974): "Parameter Estimation for a Multiplicative Competitive Interaction Model - Least Squares Approach". In: *Journal of Marketing Research*, **11**, 3, p. 303-311.

Wieland, T. (2013): "Einkaufsstaettenwahl, Einzelhandelscluster und raeumliche Versorgungsdisparitaeten - Modellierung von Marktgebieten im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten". In: Schrenk, M./Popovich, V./Zeile, P./Elisei, P. (eds.): REAL CORP 2013. Planning Times. Proceedings of 18th International Conference on Urban Planning, Regional Development and Information Society. Schwechat. p. 275-284. http://www.corp.at/archive/CORP2013_98.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

mci.fit, mci.transvar

Examples

```
# MCI analysis for the grocery store market areas in grocery2 #
data(grocery2)
# Loads the data
mci.transmat (grocery2, "plz_submarket", "store_code", "p_ij_obs", "dist_km", "salesarea_qm")
# Applies the log-centering transformation to the dataset using the function mci.transmat



# MCI analysis for the grocery store market areas based on the POS survey in shopping1 #

data(shopping1)
```

```
# Loading the survey dataset
data(shopping2)
# Loading the distance/travel time dataset
data(shopping3)
# Loading the dataset containing information about the city districts
data(shopping4)
# Loading the grocery store data

shopping1_KAeast <- shopping1[shopping1$resid_code %in%
shopping3$resid_code[shopping3$KA_east == 1],]
# Extracting only inhabitants of the eastern districts of Karlsruhe

ijmatrix_gro_adj <- ijmatrix.create(shopping1_KAeast, "resid_code",
"gro_purchase_code", "gro_purchase_expen", remSing = TRUE, remSing.val = 1,
remSingSupp.val = 2, correctVar = TRUE, correctVar.val = 0.1)
# Removing singular instances/outliers (remSing = TRUE) incorporating
# only suppliers which are at least obtained three times (remSingSupp.val = 2)
# Correcting the values (correctVar = TRUE)
# by adding 0.1 to the absolute values (correctVar.val = 0.1)

ijmatrix_gro_adj <- ijmatrix_gro_adj[(ijmatrix_gro_adj$gro_purchase_code !=
"REFORMHAUSBOESER") & (ijmatrix_gro_adj$gro_purchase_code != "WMARKT_DURLACH")
& (ijmatrix_gro_adj$gro_purchase_code != "X_INCOMPLETE_STORE"),]
# Remove non-regarded observations

ijmatrix_gro_adj_dist <- merge (ijmatrix_gro_adj, shopping2, by.x="interaction",
by.y="route")
# Include the distances and travel times (shopping2)
ijmatrix_gro_adj_dist_stores <- merge (ijmatrix_gro_adj_dist, shopping4,
by.x = "gro_purchase_code", by.y = "location_code")
# Adding the store information (shopping4)

mci.transvar(ijmatrix_gro_adj_dist_stores, "resid_code", "gro_purchase_code",
"p_ij_obs")
# Log-centering transformation of one variable (p_ij_obs)

ijmatrix_gro_transf <- mci.transmat(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# Log-centering transformation of the interaction matrix

mcimodel_gro_trips <- mci.fit(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# MCI model for the grocery store market areas
# shares: "p_ij_obs", explanatory variables: "d_time", "salesarea_qm"

summary(mcimodel_gro_trips)
# Use like lm
```

---

mci.transvar   *Log-centering transformation of one variable in an MCI interaction matrix*

Description

This function applies the *log-centering transformation* to a variable in a given MCI interaction matrix.

Usage

```
mci.transvar(mcidataset, submarkets, suppliers, mcivariable,
output_ij = FALSE, output_var = "numeric", show_proc = FALSE, check_df = TRUE)
```

Arguments

mcidataset	an interaction matrix which is a `data.frame` containing the submarkets, suppliers and the regarded variables (e.g. the observed market shares, p_{ij}, and the explanatory variables)
submarkets	the column in the interaction matrix `mcidataset` containing the submarkets
suppliers	the column in the interaction matrix `mcidataset` containing the suppliers
mcivariable	the column of the variable to be transformed, numeric and positive (or dummy [1,0])
output_ij	logical argument that indicates if the function output has to be a `data.frame` with three columns (submarkets, suppliers, transformed variable) or a vector only with the transformed values (default is `output_ij = FALSE`)
output_var	defines the mode of the function output if `output_ij = FALSE` (default is `output_var = "numeric"`, otherwise `"list"`)
show_proc	logical argument that indicates if the function prints messages about the state of process during the work (e.g. "Processing variable xyz ..." or "Variable xyz is regarded as dummy variable"). Default: `show_proc = FALSE` (messages off)
check_df	logical argument that indicates if the input (dataset, column names) is checked (default: `check_df = TRUE` (should not be changed, only for internal use))

Details

This function transforms one variable from the input dataset (MCI interaction matrix) to regression-ready data with the *log-centering transformation* by Nakanishi/Cooper (1974) (to transform a complete interaction matrix, use `mci.transmat()`, for transformation and fitting use `mci.fit()`). The *log-centering transformation* can be regarded as the key concept of the MCI model because it enables the model to be estimated by OLS (ordinary least squares) regression. The function identifies dummy variables which are not transformed (because they do not have to be).

Value

The format of the output can be controlled by the last two arguments of the function (see above). Either a new `data.frame` with the transformed input variable and the submarkets/suppliers or a vector with the transformed values only. The name of the input variable is passed to the new `data.frame` marked with a "_t" to indicate that it was transformed (e.g. "shares_t" is the transformation of "shares").

Author(s)

Thomas Wieland

References

Huff, D. L./Batsell, R. R. (1975): "Conceptual and Operational Problems with Market Share Models of Consumer Spatial Behavior". In: *Advances in Consumer Research*, **2**, p. 165-172.

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf

Nakanishi, M./Cooper, L. G. (1974): "Parameter Estimation for a Multiplicative Competitive Interaction Model - Least Squares Approach". In: *Journal of Marketing Research*, **11**, 3, p. 303-311.

Wieland, T. (2013): "Einkaufsstaettenwahl, Einzelhandelscluster und raeumliche Versorgungsdisparitaeten - Modellierung von Marktgebieten im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten". In: Schrenk, M./Popovich, V./Zeile, P./Elisei, P. (eds.): REAL CORP 2013. Planning Times. Proceedings of 18th International Conference on Urban Planning, Regional Development and Information Society. Schwechat. p. 275-284. http://www.corp.at/archive/CORP2013_98.pdf

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

mci.fit, mci.transmat

Examples

```
# MCI analysis for the grocery store market areas based on the POS survey in shopping1 #

data(shopping1)
# Loading the survey dataset
data(shopping2)
# Loading the distance/travel time dataset
data(shopping3)
# Loading the dataset containing information about the city districts
data(shopping4)
# Loading the grocery store data

shopping1_KAeast <- shopping1[shopping1$resid_code %in%
shopping3$resid_code[shopping3$KA_east == 1],]
# Extracting only inhabitants of the eastern districts of Karlsruhe

ijmatrix_gro_adj <- ijmatrix.create(shopping1_KAeast, "resid_code",
"gro_purchase_code", "gro_purchase_expen", remSing = TRUE, remSing.val = 1,
remSingSupp.val = 2, correctVar = TRUE, correctVar.val = 0.1)
# Removing singular instances/outliers (remSing = TRUE) incorporating
```

```
# only suppliers which are at least obtained three times (remSingSupp.val = 2)
# Correcting the values (correctVar = TRUE)
# by adding 0.1 to the absolute values (correctVar.val = 0.1)

ijmatrix_gro_adj <- ijmatrix_gro_adj[(ijmatrix_gro_adj$gro_purchase_code !=
"REFORMHAUSBOESER") & (ijmatrix_gro_adj$gro_purchase_code != "WMARKT_DURLACH")
& (ijmatrix_gro_adj$gro_purchase_code != "X_INCOMPLETE_STORE"),]
# Remove non-regarded observations

ijmatrix_gro_adj_dist <- merge (ijmatrix_gro_adj, shopping2, by.x="interaction",
by.y="route")
# Include the distances and travel times (shopping2)
ijmatrix_gro_adj_dist_stores <- merge (ijmatrix_gro_adj_dist, shopping4,
by.x = "gro_purchase_code", by.y = "location_code")
# Adding the store information (shopping4)

mci.transvar(ijmatrix_gro_adj_dist_stores, "resid_code", "gro_purchase_code",
"p_ij_obs")
# Log-centering transformation of one variable (p_ij_obs)

ijmatrix_gro_transf <- mci.transmat(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# Log-centering transformation of the interaction matrix

mcimodel_gro_trips <- mci.fit(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# MCI model for the grocery store market areas
# shares: "p_ij_obs", explanatory variables: "d_time", "salesarea_qm"

summary(mcimodel_gro_trips)
# Use like lm
```

---

model.fit   *Goodness of fit statistics for the Huff model*

Description

This function calculates several goodness of fit values to evaluate how good the observed values fit
the empirical observations.

Usage

```
model.fit(y_obs, y_exp, plotVal = FALSE)
```

Arguments

y_obs	Observed values of the dependent variable y
y_exp	Expected values of the dependent variable y (model results)
plotVal	Logical argument that indicates if the function plots a graph comparing observed and expected values

Details

This function computes several goodness of fit statistics to evaluate the results of non-linear fitting procedures for the Huff model (see the functions `huff.attrac` and `huff.fit`). Besides the sum of squared residuals, the function also calculates a *Pseudo-R-squared* measure and the *MAPE (mean average percentage error)*, both used by De Beule et al. (2014), and the *global error* used by Klein (1988).

Value

list:

`resids_sq_sum`	Sum of squared residuals
`pseudorsq`	Pseudo-R-squared
`globerr`	Global error
`mape`	Mean average percentage error

Author(s)

Thomas Wieland

References

De Beule, M./Van den Poel, D./Van de Weghe, N. (2014): "An extended Huff-model for robustly benchmarking and predicting retail network performance". In: *Applied Geography*, **46**, 1, p. 80-89.

Klein, R. (1988): "Der Lebensmittel-Einzelhandel im Raum Verden. Raeumliches Einkaufsverhalten unter sich wandelnden Bedingungen". *Flensburger Arbeitspapiere zur Landeskunde und Raumordnung*, **6**. Flensburg.

See Also

huff.fit

Examples

```
# Controlling the fit of a Huff Model market area estimation #

data(Freiburg1)
data(Freiburg2)
data(Freiburg3)
# Loads the data

huff_mat <- huff.shares (Freiburg1, "district", "store", "salesarea", "distance")
# Market area estimation using the Huff Model with standard parameters
# (gamma = 1, lambda = -2)

huff_mat_pp <- merge (huff_mat, Freiburg2)
# Adding the purchasing power data for the city districts

huff_total <- shares.total (huff_mat_pp, "district", "store", "p_ij", "ppower")
```

```
# Total expected sales and shares

huff_total_control <- merge (huff_total, Freiburg3, by.x = "suppliers_single",
by.y = "store")

model.fit(huff_total_control$annualsales, huff_total_control$sum_E_j, plotVal = TRUE)
# Observed vs. expected

# Results can be adressed directly:
huff_fit <- model.fit(huff_total_control$annualsales, huff_total_control$sum_E_j, plotVal = TRUE)
huff_fit$mape
```

---

shares.segm   *Segmentation of market areas by a criterion*

Description

This function segments the contents of an interaction matrix based on a criterion, such as distance or market penetration.

Usage

```
shares.segm(mcidataset, submarkets, suppliers, segmentation, observations,
..., check_df = TRUE)
```

Arguments

mcidataset	an interaction matrix which is a data.frame containing the submarkets, suppliers, the total values (e.g. shopping trips, expenditures) and a variable containing the criterion for the segmentation (e.g. distance, travel time, market penetration)
submarkets	the column in the interaction matrix mcidataset containing the submarkets
suppliers	the column in the interaction matrix mcidataset containing the suppliers
segmentation	The column in the interaction matrix mcidataset containing the variable which is to be used for segmentation
observations	The column in the interaction matrix mcidataset containing the total values which have to segmented
...	The stated limits of class segments (e.g. 0, 10, 20, 30)
check_df	logical argument that indicates if the input (dataset, column names) is checked (default: check_df = TRUE (should not be changed, only for internal use))

Details

For practical reasons, a market/market area can be zoned into segments based on a criterion (such as distance or travel time zones, zones of market penetration). Based on an existing interaction matrix, this function returns zones of a market/market area.

Value

Returns a new `data.frame` with the classification segments, the sum of the total observed values with respect to each class and the corresponding percentage.

Author(s)

Thomas Wieland

References

Berman, B. R./Evans, J. R. (2013): "Retail Management: A Strategic Approach". Pearson, 12 edition, 2013.

See Also

`shares.total`

Examples

```
# Market area segmentation based on the POS survey in shopping1 #

data(shopping1)
# The survey dataset
data(shopping2)
# Dataset with distances and travel times

shopping1_adj <- shopping1[(shopping1$weekday != 3) & (shopping1$holiday != 1)
& (shopping1$survey != "pretest"),]
# Removing every case from tuesday, holidays and the ones belonging to the pretest

ijmatrix_POS <- ijmatrix.create(shopping1_adj, "resid_code", "POS", "POS_expen")
# Creates an interaction matrix based on the observed frequencies (automatically)
# and the POS expenditures (Variable "POS_expen" separately stated)

ijmatrix_POS_data <- merge(ijmatrix_POS, shopping2, by.x="interaction", by.y="route",
all.x = TRUE)
# Adding the distances and travel times

ijmatrix_POS_data_segm_visit <- shares.segm(ijmatrix_POS_data, "resid_code", "POS",
"d_time", "freq_ij_abs", 0,10,20,30)
# Segmentation by travel time using the number of customers/visitors
# Parameters: interaction matrix (data frame), columns with origins and destinations,
# variable to divide in classes, absolute frequencies/expenditures, class segments

ijmatrix_POS_data_segm_exp <- shares.segm(ijmatrix_POS_data, "resid_code", "POS",
"d_time", "freq_ij_abs_POS_expen", 0,10,20,30)
# Segmentation by travel time using the POS expenditures
```

---

shares.total	*Total market shares/market areas*

Description

This function calculates the total sales and market shares (or total market area) of the suppliers based on a given interaction matrix which already contains (local) market shares.

Usage

```
shares.total(mcidataset, submarkets, suppliers, shares, localmarket,
plotChart = FALSE, plotChart.title = "Total sales", plotChart.unit = "sales",
check_df = TRUE)
```

Arguments

mcidataset	an interaction matrix which is a data.frame containing the submarkets, suppliers, the market shares and a variable for the local market potential (e.g. purchasing power, number of customers, population)
submarkets	the column in the interaction matrix mcidataset containing the submarkets
suppliers	the column in the interaction matrix mcidataset containing the suppliers
shares	the column in the interaction matrix mcidataset containing the the (local) market shares
localmarket	the column in the interaction matrix mcidataset containing the local market potential
plotChart	logical argument that indicates if the total values shall be visualized in a bar plot (default: plotChart = FALSE)
plotChart.title	
	If plotChart = TRUE: Title of the plot
plotChart.unit	If plotChart = TRUE: Unit of the plotted total values (e.g. a currency), used as plot subtitle
check_df	logical argument that indicates if the input (dataset, column names) is checked (default: check_df = TRUE (should not be changed, only for internal use))

Details

If (local) market shares are observed and estimated, respectively, it is possible to link them to a (local) market potential to estimate the total sales and shares of the given suppliers. In this function, the input dataset (interaction matrix with local market shares) is used for the calculation of total sales (or total number of customers) and total market shares of all j regarded suppliers. Optionally, the function also returns a simple bar plot of the total values.

Value

Returns a new data.frame with the total sales (sum_E_j) and the over-all market shares of the j suppliers (share_j).

Author(s)

Thomas Wieland

References

Huff, D. L./McCallum, D. (2008): "Calibrating the Huff Model Using ArcGIS Business Analyst". *ESRI White Paper*, September 2008. `https://www.esri.com/library/whitepapers/pdfs/calibrating-huff-model.pdf`

Nakanishi, M./Cooper, L. G. (1974): "Parameter Estimation for a Multiplicative Competitive Interaction Model - Least Squares Approach". In: *Journal of Marketing Research*, **11**, 3, p. 303-311.

Nakanishi, M./Cooper, L. G. (1982): "Simplified Estimation Procedures for MCI Models". In: *Marketing Science*, **1**, 3, p. 314-322.

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

See Also

`mci.fit`, `mci.transmat`, `mci.transvar`, `mci.shares`

Examples

```
data(Freiburg1)
data(Freiburg2)
# Loads the data

mynewmatrix <- mci.shares(Freiburg1, "district", "store", "salesarea", 1, "distance", -2)
# Calculating shares based on two attractivity/utility variables

mynewmatrix_alldata <- merge(mynewmatrix, Freiburg2)
# Merge interaction matrix with district data (purchasing power)

shares.total (mynewmatrix_alldata, "district", "store", "p_ij", "ppower")
# Calculation of total sales
```

---

shopping1	*Point-of-sale survey in Karlsruhe*

Description

The dataset contains a point-of-sale (POS) survey conducted at two retail supply locations (town centre and planned shopping centre) in the east of Karlsruhe (Germany) from May 2016 (raw data). Amongst other things, the participants were asked about their last shopping trip with respect to groceries, clothing als consumer electronics (store choice and expenditures) and their place of residence (ZIP code and city district, respectively). The survey dataset contains 434 cases/interviewed individuals. The survey is *not* representative and should be regarded as an example.

Usage

```
data("shopping1")
```

Format

A data frame with 434 observations on the following 29 variables.

POS a factor indicating the survey location: POS1 (town centre) or POS2 (shopping centre)

time a numeric vector containing the code for the time period the interview was conducted

date a POSIXct containing the date the interview was conducted

POS_traffic a numeric vector containing the code for the traffic mode the respondet used to come to the supply location

POS_stay a numeric vector containing the respondents' duration of stay at the supply location

POS_expen a numeric vector containing the respondents' expenditures at the supply location

POS1_freq a numeric vector containing the frequency of visiting the supply location POS1

POS2_freq a numeric vector containing the frequency of visiting the supply location POS2

gro_purchase_code a factor containing the destination of the last grocery shopping trip

gro_purchase_brand a factor containing the brand (store chain) of the destination of the last grocery shopping trip

gro_purchase_channel a factor containing the shopping channel of the destination of the last grocery shopping trip: ambulant, online and store

gro_purchase_expen a numeric vector containing the expenditures corresponding to the last grocery shopping trip

cloth_purchase_code a factor containing the destination of the last clothing shopping trip

cloth_purchase_brand a factor containing the brand (store chain) of the destination of the last clothing shopping trip

cloth_purchase_channel a factor containing the shopping channel of the destination of the last clothing shopping trip: mail order, online or store

cloth_purchase_expen a numeric vector containing the expenditures corresponding to the last clothing shopping trip

ce_purchase_code a factor containing the destination of the last shopping trip with respect to consumer electronics (CE)

ce_purchase_brand a factor containing the brand (store chain) of the destination of the last CE shopping trip

ce_purchase_channel a factor containing the shopping channel of the destination of the last CE shopping trip: online or store

ce_purchase_expen a numeric vector containing the expenditures corresponding to the last CE shopping trip

resid_PLZ a factor containing the customer origin (place of residence) as ZIP code

resid_name a factor containing the customer origin (place of residence) as name of the corresponding city or city district

resid_name_official a factor containing the customer origin (place of residence) as official names of the corresponding city or city district

resid_code a factor containing the customer origin (place of residence) as internal code

age_cat a numeric vector containing the age category of the respondent

sex a numeric vector containing the sex of the respondent

weekday a numeric vector containing the weekday where the interview took place

holiday a numeric vector containing a dummy variable which indicates whether the interview was conducted on a holiday or not

survey a factor reflecting the mode of survey: main is the main survey while pretest marks the cases from the pretest

Source

Primary empirical sources: POS (point of sale) survey in the authors' course ("Praktikum Empirische Sozialforschung: Stadtteilzentren als Einzelhandelsstandorte - Das Fallbeispiel Karlsruhe-Durlach", Karlsruhe Institute of Technology, Institute for Geography and Geoecology, May 2016), own calculations

See Also

shopping2, shopping3, shopping4

Examples

```
# Market area segmentation based on the POS survey in shopping1 #

data(shopping1)
# The survey dataset
data(shopping2)
# Dataset with distances and travel times

shopping1_adj <- shopping1[(shopping1$weekday != 3) & (shopping1$holiday != 1)
& (shopping1$survey != "pretest"),]
# Removing every case from tuesday, holidays and the ones belonging to the pretest

ijmatrix_POS <- ijmatrix.create(shopping1_adj, "resid_code", "POS", "POS_expen")
# Creates an interaction matrix based on the observed frequencies (automatically)
# and the POS expenditures (Variable "POS_expen" separately stated)

ijmatrix_POS_data <- merge(ijmatrix_POS, shopping2, by.x="interaction", by.y="route",
all.x = TRUE)
# Adding the distances and travel times

ijmatrix_POS_data_segm_visit <- shares.segm(ijmatrix_POS_data, "resid_code", "POS",
"d_time", "freq_ij_abs", 0,10,20,30)
# Segmentation by travel time using the number of customers/visitors
# Parameters: interaction matrix (data frame), columns with origins and destinations,
# variable to divide in classes, absolute frequencies/expenditures, class segments
```

```
ijmatrix_POS_data_segm_exp <- shares.segm(ijmatrix_POS_data, "resid_code", "POS",
"d_time", "freq_ij_abs_POS_expen", 0,10,20,30)
# Segmentation by travel time using the POS expenditures
```

---

shopping2   *Distance matrix for the point-of-sale survey in Karlsruhe*

Description

The dataset contains a distance matrix (OD matrix: Origins-Destinations matrix) including the street distance and the travel time from the customer origins to the shopping destinations, both stored in the dataset shopping1.

Usage

```
data("shopping2")
```

Format

A data frame with 3723 observations on the following 5 variables.

from a factor containing the customer origin (place of residence) as internal code

to a factor containing the shopping destination

d_km a numeric vector containing the street distance from the origins to the destinations in km

d_time a numeric vector containing the driving time from the origins to the destinations in km

route a factor containing the interaction/route code between origins and destinations (from-to)

Source

Primary empirical sources: POS (point of sale) survey in the authors' course ("Praktikum Empirische Sozialforschung: Stadtteilzentren als Einzelhandelsstandorte - Das Fallbeispiel Karlsruhe-Durlach", Karlsruhe Institute of Technology, Institute for Geography and Geoecology, May 2016), own calculations

The street distance and travel time was calculated using the package **ggmap**.

See Also

shopping1, shopping3, shopping4

Examples

```
# Market area segmentation based on the POS survey in shopping1 #

data(shopping1)
# The survey dataset
data(shopping2)
# Dataset with distances and travel times
```

```
shopping1_adj <- shopping1[(shopping1$weekday != 3) & (shopping1$holiday != 1)
& (shopping1$survey != "pretest"),]
# Removing every case from tuesday, holidays and the ones belonging to the pretest

ijmatrix_POS <- ijmatrix.create(shopping1_adj, "resid_code", "POS", "POS_expen")
# Creates an interaction matrix based on the observed frequencies (automatically)
# and the POS expenditures (Variable "POS_expen" separately stated)

ijmatrix_POS_data <- merge(ijmatrix_POS, shopping2, by.x="interaction", by.y="route",
all.x = TRUE)
# Adding the distances and travel times

ijmatrix_POS_data_segm_visit <- shares.segm(ijmatrix_POS_data, "resid_code", "POS",
"d_time", "freq_ij_abs", 0,10,20,30)
# Segmentation by travel time using the number of customers/visitors
# Parameters: interaction matrix (data frame), columns with origins and destinations,
# variable to divide in classes, absolute frequencies/expenditures, class segments

ijmatrix_POS_data_segm_exp <- shares.segm(ijmatrix_POS_data, "resid_code", "POS",
"d_time", "freq_ij_abs_POS_expen", 0,10,20,30)
# Segmentation by travel time using the POS expenditures
```

---

shopping3   *Market area data for the point-of-sale survey in Karlsruhe*

Description

The dataset contains information about

Usage

```
data("shopping3")
```

Format

A data frame with 70 observations on the following 5 variables.

resid_name a factor containing the customer origin (place of residence) as name of the corresponding city or city district

resid_name_offical a factor containing the customer origin (place of residence) as official names of the corresponding city or city district

resid_pop2015 a numeric vector containing the population size of the area

KA_east a numeric vector containing a dummy variable indicating whether the area belongs to the east of Karlsruhe or not

resid_code a factor containing the customer origin (place of residence) as internal code

Source

Primary empirical sources: POS (point of sale) survey in the authors' course ("Praktikum Empirische Sozialforschung: Stadtteilzentren als Einzelhandelsstandorte - Das Fallbeispiel Karlsruhe-Durlach", Karlsruhe Institute of Technology, Institute for Geography and Geoecology, May 2016), own calculations

Stadt Karlsruhe, Amt fuer Stadtentwicklung (2016): "Die Karlsruher Bevoelkerung im Dezember 2015". Stadt Karlsruhe.

See Also

shopping1, shopping2, shopping4

Examples

```
# Market area analysis based on the POS survey in shopping1 #

data(shopping1)
# The survey dataset
data(shopping2)
# Dataset with distances and travel times

shopping1_adj <- shopping1[(shopping1$weekday != 3) & (shopping1$holiday != 1)
& (shopping1$survey != "pretest"),]
# Removing every case from tuesday, holidays and the ones belonging to the pretest

ijmatrix_POS <- ijmatrix.create(shopping1_adj, "resid_code", "POS", "POS_expen")
# Creates an interaction matrix based on the observed frequencies (automatically)
# and the POS expenditures (Variable "POS_expen" separately stated)

ijmatrix_POS_data <- merge(ijmatrix_POS, shopping2, by.x="interaction", by.y="route",
all.x = TRUE)
# Adding the distances and travel times

ijmatrix_POS_data$freq_ij_abs_cor <- var.correct(ijmatrix_POS_data$freq_ij_abs,
corr.mode = "inc", incby = 0.1)
# Correcting the absolute values (frequencies) by increasing by 0.1

data(shopping3)
ijmatrix_POS_data_residdata <- merge(ijmatrix_POS_data, shopping3)
# Adding the information about the origins (places of residence) stored in shopping3

ijmatrix_POS_data_residdata$visitper1000 <- (ijmatrix_POS_data_residdata$
freq_ij_abs_cor/ijmatrix_POS_data_residdata$resid_pop2015)*1000
# Calculating the dependent variable
# visitper1000: surveyed customers per 1.000 inhabitants of the origin

ijmatrix_POS_data_residdata <-
ijmatrix_POS_data_residdata[(!is.na(ijmatrix_POS_data_residdata$
visitper1000)) & (!is.na(ijmatrix_POS_data_residdata$d_time)),]
# Removing NAs (data for some outlier origins and routes not available)
```

```
ijmatrix_POS_data_residdata_POS1 <-
ijmatrix_POS_data_residdata[ijmatrix_POS_data_residdata$POS=="POS1",]
# Dataset for POS1 (town centre)

ijmatrix_POS_data_residdata_POS2 <-
ijmatrix_POS_data_residdata[ijmatrix_POS_data_residdata$POS=="POS2",]
# Dataset for POS2 (out-of-town shopping centre)

huff.decay(ijmatrix_POS_data_residdata_POS1, "d_km", "visitper1000")
huff.decay(ijmatrix_POS_data_residdata_POS1, "d_time", "visitper1000")
huff.decay(ijmatrix_POS_data_residdata_POS2, "d_km", "visitper1000")
huff.decay(ijmatrix_POS_data_residdata_POS2, "d_time", "visitper1000")
```

---

shopping4   *Grocery store data for the point-of-sale survey in Karlsruhe*

Description

This dataset contains information about the regarded grocery stores in the east of Karlsruhe, based on the POS survey stored in the dataset shopping1 and the related information in shopping2..

Usage

```
data("shopping4")
```

Format

A data frame with 11 observations on the following 4 variables.

location_code a factor containing the grocery store codes

salesarea_qm a numeric vector containing the sales area of the stores in sqm

storetype_dc a numeric vector containing a dummy variable that indicates if the store is a discounter or not

store_chain a factor containing the store chain

Source

Primary empirical sources: POS (point of sale) survey in the authors' course ("Praktikum Empirische Sozialforschung: Stadtteilzentren als Einzelhandelsstandorte - Das Fallbeispiel Karlsruhe-Durlach", Karlsruhe Institute of Technology, Institute for Geography and Geoecology, May 2016), own calculations

Mapping of grocery stores in the east of Karlsruhe in June 2016 with additional research

See Also

shopping1, shopping2, shopping3

Examples

```
# MCI analysis for the grocery store market areas based on the POS survey in shopping1 #

data(shopping1)
# Loading the survey dataset
data(shopping2)
# Loading the distance/travel time dataset
data(shopping3)
# Loading the dataset containing information about the city districts
data(shopping4)
# Loading the grocery store data

shopping1_KAeast <- shopping1[shopping1$resid_code %in%
shopping3$resid_code[shopping3$KA_east == 1],]
# Extracting only inhabitants of the eastern districts of Karlsruhe

ijmatrix_gro_adj <- ijmatrix.create(shopping1_KAeast, "resid_code",
"gro_purchase_code", "gro_purchase_expen", remSing = TRUE, remSing.val = 1,
remSingSupp.val = 2, correctVar = TRUE, correctVar.val = 0.1)
# Removing singular instances/outliers (remSing = TRUE) incorporating
# only suppliers which are at least obtained three times (remSingSupp.val = 2)
# Correcting the values (correctVar = TRUE)
# by adding 0.1 to the absolute values (correctVar.val = 0.1)

ijmatrix_gro_adj <- ijmatrix_gro_adj[(ijmatrix_gro_adj$gro_purchase_code !=
"REFORMHAUSBOESER") & (ijmatrix_gro_adj$gro_purchase_code != "WMARKT_DURLACH")
& (ijmatrix_gro_adj$gro_purchase_code != "X_INCOMPLETE_STORE")),]
# Remove non-regarded observations

ijmatrix_gro_adj_dist <- merge (ijmatrix_gro_adj, shopping2, by.x="interaction",
by.y="route")
# Include the distances and travel times (shopping2)
ijmatrix_gro_adj_dist_stores <- merge (ijmatrix_gro_adj_dist, shopping4,
by.x = "gro_purchase_code", by.y = "location_code")
# Adding the store information (shopping4)

mci.transvar(ijmatrix_gro_adj_dist_stores, "resid_code", "gro_purchase_code",
"p_ij_obs")
# Log-centering transformation of one variable (p_ij_obs)

ijmatrix_gro_transf <- mci.transmat(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# Log-centering transformation of the interaction matrix

mcimodel_gro_trips <- mci.fit(ijmatrix_gro_adj_dist_stores, "resid_code",
"gro_purchase_code", "p_ij_obs", "d_time", "salesarea_qm")
# MCI model for the grocery store market areas
# shares: "p_ij_obs", explanatory variables: "d_time", "salesarea_qm"

summary(mcimodel_gro_trips)
# Use like lm
```

---

var.asdummy   *Creating dummy variables*

Description

This function creates a dataset of dummy variables based on an input character vector.

Usage

```
var.asdummy(x)
```

Arguments

x   A character vector

Details

In MCI analyzes (as in OLS regression models generally) only *quantitative* information (that means: numeric) is allowed. *Qualitative* information (e.g. brands, companies, retail chains) can be added using dummy variables [1,0]. This function transforms a character vector x with c characteristics to a set of c dummy variables whose column names correspond to these characteristics marked with "_DUMMY".

Value

A data.frame with dummy variables corresponding to the levels of the input variable.

Author(s)

Thomas Wieland

References

Nakanishi, M./Cooper, L. G. (1982): "Simplified Estimation Procedures for MCI Models". In: *Marketing Science*, **1**, 3, p. 314-322.

Tihi, B./Oruc, N. (2012): "Competitive Location Assessment - the MCI Approach". In: *South East European Journal of Economics and Business*, **7**, 2, p. 35-49.

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellana-lytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fall-beispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handels-forschung*, **23**. 289 pages. Mannheim : MetaGIS.

Examples

```
charvec <- c("Peter", "Paul", "Peter", "Mary", "Peter", "Paul")
# Creates a vector with three names (Peter, Paul, Mary)
var.asdummy(charvec)
# Returns a data frame with 3 dummy variables
# (Mary_DUMMY, Paul_DUMMY, Peter_DUMMY)

data(grocery2)
# Loads the data
dummyvars <- var.asdummy(grocery2$store_chain)
# Save the dummy variable set into a new dataset
mynewmcidata <- data.frame(grocery2, dummyvars)
# Add the dummy dataset to the input dataset
```

---

var.correct   *Correcting MCI input variables*

Description

This function corrects a numeric variable to match the MCI standards.

Usage

```
var.correct(x, corr.mode = "inc", incby = 1)
```

Arguments

x	a numeric vector
corr.mode	character value for the mode of variable correction: corr.mode = "inc" increases the values by the value of incby (default: incby = 1). If corr.mode = "incabs" the values are increased by the absolute value of their minimum + incby. If corr.mode = "zetas" a zeta squared transformation is applied to the numeric vector
incby	value to increase the values with (default = 1)

Details

In the MCI model, only numeric variables with values greater than zero are accepted (From the theoretical perspective, a zero or negative attractivity/utility is just as impossible as negative market shares. In the log-centering transformation, those values cannot be processed.). This function corrects a numeric variable with zero and/or negative values to match the MCI standards. The most frequent case is that some absolute values which shall be used to calculate market shares (e.g. observed frequencies or expenditures) are equal to zero and must be increased by 1. Alternatively, they can be increased automatically by the absolute value of their minimum + incby. Another option which is especially designed to transform interval scale data (such as scoring in consumer surveys) is to apply a zeta-squared transformation (Cooper/Nakanishi 1983) to the numeric vector (corr.mode = "zetas").

Value

Returns a numeric vector with the corrected values.

Author(s)

Thomas Wieland

References

Colome Perales, R. (2002): "Consumer Choice in Competitive Location Models". Barcelona.

Cooper, L.G./Nakanishi, M. (1983): "Standardizing Variables in Multiplicative Choice Models". In: *Journal of Consumer Research*, **10**, 1, p. 96-108.

Cooper, L. G./Nakanishi, M. (2010): "Market-Share Analysis: Evaluating competitive marketing effectiveness". Boston, Dordrecht, London : Kluwer (first published 1988). E-book version from 2010: http://www.anderson.ucla.edu/faculty/lee.cooper/MCI_Book/BOOKI2010.pdf

Hartmann, M. (2005): "Gravitationsmodelle als Verfahren der Standortanalyse im Einzelhandel". *Statistik Regional Electronic Papers*, **02/2005**. Halle.

Tihi, B./Oruc, N. (2012): "Competitive Location Assessment - the MCI Approach". In: *South East European Journal of Economics and Business*, **7**, 2, p. 35-49.

Wieland, T. (2015): "Raeumliches Einkaufsverhalten und Standortpolitik im Einzelhandel unter Beruecksichtigung von Agglomerationseffekten. Theoretische Erklaerungsansaetze, modellanalytische Zugaenge und eine empirisch-oekonometrische Marktgebietsanalyse anhand eines Fallbeispiels aus dem laendlichen Raum Ostwestfalens/Suedniedersachsens". *Geographische Handelsforschung*, **23**. 289 pages. Mannheim : MetaGIS.

Examples

```
var1 <- c(11, 17.5, 24.1, 0.9, 21.2, 0)
# a vector containing one zero value
var.correct(var1)
# returns a vector with input values increased by 1

var2 <- -5:5
# a vector containing zero and negative values
var.correct(var2, corr.mode = "incabs", incby = 1)
# returns a vector with minimum value equal to 1

var.correct(var2, corr.mode = "zetas")
# returns a vector only with positive values
# (zeta-squared transformation)
```

Index