Package 'MCI2'

January 3, 2019

Type Package

Title Market Area Models for Retail and Service Locations

Version 1.1.1

Date 2019-01-03

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 is a more user-friendly wrapper of the functions in the package 'MCI' (Wieland 2017) providing market area analysis using the Huff Model or the Multiplicative Competitive Interaction (MCI) Model. In 'MCI2', also a function for creating transport costs matrices is provided.

Depends R (>= 3.5.0)

Imports MCI, REAT, tmaptools, osrm, reshape

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-03 08:40:03 UTC

R topics documented:

`MCI2-package` *Market Area Models for Retail and Service Locations*

Description

This package provides several function for market area modeling with respect to retail and service locations (for a review, see Wieland 2017). The package **MCI2** has two purposes:

1) A user-friendly wrapper for the functions in the package **MCI** (Wieland 2017) concerning the Huff Model and the Multiplicative Competitive Interaction Model. For more information, see the **MCI** package documentation and the related RJ paper (Wieland 2017).

2) A user-friendly way to create transport costs matrices, including the usage of OSM (OpenStreetMap)-related APIs. The included function `tcmat.create` is a convenient wrapper of the functions `geocode_OSM` (for geocoding street addresses using OSM Nominatim) from the **tmaptools** package (Tennekes 2018), `osrmTable` (for creating travel time matrices between points while interfacing the table OSRM service) from the **osrm** package (Giraud 2018) and `dist.mat` (for creating airline distances between points) from the **REAT** package (Wieland 2018). For more information, see the related package documentations.

Author(s)

Thomas Wieland

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

References

Giraud, T. (2018): osrm: Interface Between R and the OpenStreetMap-Based Routing Service OSRM. R package version 3.1.1. https://CRAN.R-project.org/package=osrm.

Tennekes, M. (2018): tmaptools: Thematic Map Tools. R package version 2.0-1. https://CRAN.R-project.org/package=tmaptools.

Wieland, T. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

Wieland, T. (2018): REAT: Regional Economic Analysis Toolbox. R package version 2.1.0. https://CRAN.R-project.org/package=REAT.

HaslachDistricts *Freiburg-Haslach: districts*

Description

Statistical districts of Haslach and additional information

Usage

```
data("HaslachDistricts")
```

Format

A data frame with 4 observations on the following 5 variables.

`WO` a character vector containing the district code

`WO_Name` a character vector containing the district name

`WO_Einwohner` a numeric vector containing the district population

`WO_lonlat` a character vector containing the longitude/latitude of the district center

`WO_Adr` a character vector containing the street address of the district center

Source

Amt fuer Buergerservice und Informationsverarbeitung, Freiburg (2018): Einwohner nach Haupt- und Nebenwohnsitz 2017.

Amt fuer Buergerservice und Informationsverarbeitung, Freiburg (2018): Kleinraeumige Gliederung: Stadtbezirke Stadt Freiburg (shape-Dateien).

Own postprocessing.

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

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes
```

```
# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
```

```
origvar.data = HaslachDistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`HaslachStores` *Freiburg-Haslach: grocery stores*

Description

Main grocery stores in Freiburg-Haslach and additional information

Usage

```
data("HaslachStores")
```

Format

A data frame with 8 observations on the following 9 variables.

`LM` a character vector containing the store code

`LM_Name` a character vector containing the store name

`LM_NameZusatz` a character vector containing the additional store name

`LM_Betriebsform` a character vector containing the store format (in German, Nielsen typology)

`LM_Adr` a character vector containing the street address of the stores

`LM_Stadt` a character vector containing the corresponding city (Freiburg)

`LM_PLZ` a character vector containing the corresponding zip code

`LM_VKF` a numeric vector containing the store size (sqm)

`LM_Adr_zus` a character vector containing the complete street address (ready for geocoding)

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

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
```

```
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

HaslachSurvey *Freiburg-Haslach: customer survey*

Description

Part of a point of sale survey (n=235) in Freiburg-Haslach with respect to grocery shopping behavior.

Usage

```
data("HaslachSurvey")
```

Format

A data frame with 470 observations on the following 24 variables.

`DATUM` a numeric vector containing the datum code

`UHRZEIT` a numeric vector containing the time code

`BEFRSTANDORT2` a numeric vector containing the sample point code

`LMHAEUF` a numeric vector containing the weekly frequency of grocery shopping

`LM1` a numeric vector containing the store code of the last grocery shopping trip

`LM1_Text` a character vector containing the store code (character) of the last grocery shopping trip, corresponding to variable `LM1` in the `HaslachStores` dataset

`LM1A` a character vector containing other shopping destinations on grocery shopping trips (not coded)

`LM1E` a numeric vector containing the amount of purchases corresponding to the last grocery shopping trip

`LM2` a numeric vector containing the store code of the second to the last grocery shopping trip

LM2_Text a character vector containing the store code (character) of the second to the the last grocery shopping trip, corresponding to variable LM2 in the HaslachStores dataset

LM2A a character vector containing other shopping destinations on grocery shopping trips (not coded)

LM2E a numeric vector containing the amount of purchases corresponding to the second to the last grocery shopping trip

ZUFR_LM a numeric vector containing customer satisfaction scores (1=best, ... 6=worst) with respect to local grocery stores supply

ZUFR_LME a numeric vector containing customer satisfaction scores (1=best, ... 6=worst) with respect to local grocery stores accessibility

ZUFR_APO a numeric vector containing customer satisfaction scores (1=best, ... 6=worst) with respect to local pharmacies supply

ZUFR_EHSO a numeric vector containing customer satisfaction scores (1=best, ... 6=worst) with respect to other types of local retailing

ZUFR_BANK a numeric vector containing customer satisfaction scores (1=best, ... 6=worst) with respect to local bank supply

WOHNSTANDORT a numeric vector containing the district code

WO a character vector containing the district code, corresponding to variable WO in the HaslachDistricts dataset

ALTERKAT a numeric vector containing the code of age categoory

GESCHL a numeric vector containing the gender code

BERUF a numeric vector containing the code of the respondent's working status

HHPERS a numeric vector containing the household size

HHKIND a numeric vector containing the no. of children in the household

Source

Own survey (June 2018). Own postprocessing.

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0
```

```
# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`Haslach_coords_destinations` *Freiburg-Haslach: coordinates of destinations (grocery stores)*

Description

Coordinates of destinations (grocery stores) in Freiburg-Haslach and related information

Usage

```
data("Haslach_coords_destinations")
```

Format

A data frame with 8 observations on the following 4 variables.

`destinations.id` Destination ID (grocery stores)

`destinations.addr` street address of grocery stores

`destinations.x_lon` longitude

`destinations.y_lat` latitude

Source

Generated by the `tcmat.create` function.

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)
```

```
Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

Haslach_coords_origins

Freiburg-Haslach: coordinates of origins (districts)

Description

Coordinates of customer origins (districts) in Freiburg-Haslach and related information

Usage

```
data("Haslach_coords_origins")
```

Format

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

`origins.id` customer origin ID (districts)

`origins.addr` street address of customer origins

`origins.x_lon` longitude

`origins.y_lat` latitude

Source

Generated by the `tcmat.create` function.

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times
```

```
# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`Haslach_tcmatAirline` *Freiburg-Haslach: airline distances*

Description

Airline distances for customer origins (districts) and destinations (grocery stores) in Freiburg-Haslach

Usage

```
data("Haslach_tcmatAirline")
```

Format

A data frame with 32 observations on the following 4 variables.

`from` customer origin ID (districts)

`to` destination ID (grocery stores)

`from_to` interactions (customer origins x destinations)

`tc` transport costs values (here: airline distances)

Source

Generated by the `tcmat.create` function.

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)
```

```
data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`Haslach_tcmatDrvtime` *Freiburg-Haslach: car driving times*

Description

Car driving times for customer origins (districts) and destinations (grocery stores) in Freiburg-Haslach

Usage

```
data("Haslach_tcmatDrvtime")
```

Format

A data frame with 32 observations on the following 4 variables.

`from` customer origin ID (districts)

`to` destination ID (grocery stores)

`from_to` interactions (customer origins x destinations)

`tc` transport costs values (here: car driving times)

Source

Generated by the `tcmat.create` function.

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)
```

```
# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`huff` *Huff Model market simulation*

Description

Calculating local market shares and total market areas of locations using the probabilistic market area model by Huff

Usage

```
huff(origins.id, origins.pot, destinations.id, destinations.attrac, tcmat,
atype = "pow", gamma = 1, gamma2 = NULL, dtype = "pow", lambda = -2, lambda2 = NULL)
```

Arguments

`origins.id`	Vector of customer origins
`origins.pot`	Vector of corresponding customer potential of the origins
`destinations.id`	Vector of destinations (stores, locations)
`destinations.attrac`	Vector of corresponding attraction values of the destinations
`tcmat`	Object (`list`) created by the function `tcmat.create` (Transportation costs matrix)

atype	Type of attraction weighting function: atype = "pow" (power function), atype = "exp" (exponential function) or atype = "logistic" (default: atype = "pow")
gamma	a single numeric value of γ for the (exponential) weighting of the attraction variable (default: 1)
gamma2	if atype = "logistic" a second γ parameter is needed
dtype	Type of distance weighting function: dtype = "pow" (power function), dtype = "exp" (exponential function) or dtype = "logistic" (default: dtype = "pow")
lambda	a single numeric value of λ for the (exponential) weighting of distance (transport costs, default: -2)
lambda2	if dtype = "logistic" a second λ parameter is needed

Details

This function computes local market shares (p_{ij}) and total market areas (T_j) according to the Huff Model.

Value

A huffmodel list (invisible) containing the following components:

huffmat	Huff interaction matrix (data frame), also containing the local market shares (p_ij)
hufftotal	Total location market areas (data frame), also containing the total market areas (T_j)
params	A matrix containing the user-defined weighting functions and the corresponding weighting parameters
coords	A list containing the coordinates and additional information of the origins (coords_origins) and the destinations (coords_destinations), inherited from the tcmat object
tc.mode	A list containing information about the transportation costs matrix, inherited from the tcmat object

Note

The function is a wrapper of huff.shares and shares.total of the **MCI** package. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

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.

Wieland, T. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

huff.newdest, huff.updest, tcmat.create

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
```

```
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

huffmodel1 <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline)
# Huff Model with standard parameters
# uses given transport costs matrix (tcmat_haslach_airline)

huffmodel1a <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline,
atype = "pow", gamma = 0.9, dtype = "pow", lambda = -2.1)
# Alternative weighting parameters

## Not run:
huffmodel2 <- huff.newdest(huffmodel1, newdest.id = "LM_NEU",
newdest.addr = "Bettackerstrasse 3, Freiburg, Germany",
newdest.attract = 1500)
# Adding a new destination with a given street address
# Recalculation of the Huff Model
# Needs internet access and accesses OpenStreetMap server(s)
# If server not available, the function will produce an error

## End(Not run)

huffmodel3 <- huff.updest (huffmodel1, dest.id = "LM01",
dest.attrac = 1200)
# Update the attraction value of one grocery stores
# Recalculation of the Huff Model

## Not run:
model.export(huffmodel3, "interactionmatrix", "totals")
# export as CSV files

## End(Not run)
```

`huff.newdest` *Huff Model market simulation with a new destination*

Description

Adding a new location and (re-)calculating local market shares and total market areas of locations using the probabilistic market area model by Huff

Usage

```
huff.newdest(huffmodel, newdest.id, newdest.addr, newdest.attract,
newdestaddr.format = "stradr", distval = NULL)
```

Arguments

huffmodel	Object (list) created by the function huff (Huff Model)
newdest.id	Code of the new destination
newdest.addr	(Street) address of the new destination
newdest.attract	Attraction value of the new destination
newdestaddr.format	Address format (default: street address)
distval	Optional: numerical vector containing the transportation costs values with respect to the new destination and all origins in the Huff Model object (huffmodel)

Details

Adding a new destination (e.g. new store) to a given location system (Huff Model object). This function computes local market shares (p_{ij}) and total market areas (T_j) according to the Huff Model.

Value

A huffmodel list (invisible) containing the following components:

huffmat	Huff interaction matrix (data frame), also containing the local market shares (p_ij)
hufftotal	Total location market areas (data frame), also containing the total market areas (T_j)
params	A matrix containing the user-defined weighting functions and the corresponding weighting parameters
coords	A list containing the coordinates and additional information of the origins (coords_origins) and the destinations (coords_destinations), inherited from the tcmat object
tc.mode	A list containing information about the transportation costs matrix, inherited from the tcmat object

Note

The function is a wrapper of huff.shares and shares.total of the **MCI** package, also integrating the tcmat.create function. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

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.

Wieland, T. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

huff, huff.updest, tcmat.create

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
```

```
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

huffmodel1 <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline)
# Huff Model with standard parameters
# uses given transport costs matrix (tcmat_haslach_airline)

huffmodel1a <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline,
atype = "pow", gamma = 0.9, dtype = "pow", lambda = -2.1)
# Alternative weighting parameters

## Not run:
huffmodel2 <- huff.newdest(huffmodel1, newdest.id = "LM_NEU",
newdest.addr = "Bettackerstrasse 3, Freiburg, Germany",
newdest.attract = 1500)
# Adding a new destination with a given street address
# Recalculation of the Huff Model
# Needs internet access and accesses OpenStreetMap server(s)
# If server not available, the function will produce an error

## End(Not run)

huffmodel3 <- huff.updest (huffmodel1, dest.id = "LM01",
dest.attrac = 1200)
# Update the attraction value of one grocery stores
# Recalculation of the Huff Model

## Not run:
model.export(huffmodel3, "interactionmatrix", "totals")
# export as CSV files

## End(Not run)
```

`huff.optim` *Huff Model optimization*

Description

This function fits the Huff Model using a local optimization of attraction algorithm

Usage

```
huff.optim(huffmodel, location.dataset, location.id, location.total,
tolerance = 5, iterations = 3, show_proc = TRUE)
```

Arguments

`huffmodel`	Object (`huffmodel list`) object created by the function `huff` (Huff Model)
`location.dataset`	A `data frame` containing the destinations and the corresponding observed total market areas, $T_{j,obs}$ (e.g. annual turnover)
`location.id`	Name of the column in `location.dataset` containing the destination IDs
`location.total`	Name of the column in `location.dataset` containing the destinations' observed total market areas, $T_{j,obs}$ (e.g. annual turnover)
`tolerance`	Accepted value of absolute percentage error between observed ($T_{j,obs}$) and estimated total values ($T_{j,exp}$) to skip a local optimization of location j
`iterations`	a single numeric value for the desired number of iterations
`show_proc`	logical argument that indicates if the function prints messages about the state of process during the work

Details

In many cases, only total empirical market areas (e.g. annual turnover) of the destinations/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 introduced and explained in Wieland (2017a) and Wieland (2017b).

Value

A `huffmodel list` (invisible) containing the following components:

`huffmat`	Huff interaction matrix (`data frame`), also containing the local market shares (`p_ij`)
`hufftotal`	Total location market areas (`data frame`), also containing both the empirical and the estimated total market areas (`T_j` and `T_j_obs`, respectively) as well as their difference (`T_j_diff`) and the new attraction values (`A_opt`)
`diag`	A `data frame` containing several model diagnoses for each iteration

params	A matrix containing the user-defined weighting functions and the corresponding weighting parameters
coords	A list containing the coordinates and additional information of the origins (coords_origins) and the destinations (coords_destinations), inherited from the tcmat object
tc.mode	A list containing information about the transportation costs matrix, inherited from the tcmat object

Note

The function is a wrapper of huff.attrac from the **MCI** package. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

Author(s)

Thomas Wieland

References

Wieland, T. (2017a): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

Wieland, T. (2017b): "Raum- und standortoekonomische Optimierungsmodelle in Open-Source-Umgebungen - Implementation und Anwendungsmoeglichkeiten im Kontext der Einzelhandels- und Versorgungsforschung". In: Schrenk, M./Popovich, V./Zeile, P./Elisei, P./Beyer, C. (eds.): REAL CORP 2017. PANTA RHEI - A World in Constant Motion. Proceedings of 22nd International Conference on Urban Planning, Regional Development and Information Society. p. 463-473. https://programm.corp.at/cdrom2017/papers2017/CORP2017_98.pdf.

See Also

huff, huff.newdest, huff.updest

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0
```

```
# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

huffmodel1 <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline)
# Huff Model with standard parameters
# uses given transport costs matrix (tcmat_haslach_airline)

huffmod_total <- huffmodel1$hufftotal
# Total values from the Huff Model

huff.optim(huffmodel1, location.dataset = huffmod_total,
location.id = "j_dest", location.total = "T_j")
# Using calculated total market areas for optimization

# now adding random errors:
rnderr <- runif(8, min=-1000, max=1500)
huffmod_total2 <- huffmod_total
huffmod_total2$T_j <- huffmod_total2$T_j+rnderr

# New optimization:
# now saving as huffmodel list object
huffmodel_opt <- huff.optim(huffmodel1, location.dataset = huffmod_total2,
location.id = "j_dest", location.total = "T_j")

huffmodel_opt
# complete huffmodel list object
```

```
huffmodel_opt$huffmat
# local market shares
```

`huff.updest` *Huff Model market simulation with an updated destination*

Description

Updating an existing location and (re-)calculating local market shares and total market areas of locations using the probabilistic market area model by Huff

Usage

```
huff.updest(huffmodel, dest.id, dest.attrac)
```

Arguments

`huffmodel`	Object (`list`) created by the function `huff` (Huff Model)
`dest.id`	Code of the existing destination that has to be updated
`dest.attrac`	New attraction value of the existing destination

Details

Updating the attraction value of an existing destination (e.g. store) in a given location system (Huff Model object). This function computes local market shares (p_{ij}) and total market areas (T_j) according to the Huff Model.

Value

A `huffmodel list` (invisible) containing the following components:

`huffmat`	Huff interaction matrix (`data frame`), also containing the local market shares (`p_ij`)
`hufftotal`	Total location market areas (`data frame`), also containing the total market areas (`T_j`)
`params`	A `matrix` containing the user-defined weighting functions and the corresponding weighting parameters
`coords`	A `list` containing the coordinates and additional information of the origins (`coords_origins`) and the destinations (`coords_destinations`), inherited from the `tcmat` object
`tc.mode`	A `list` containing information about the transportation costs matrix, inherited from the `tcmat` object

Note

The function is a wrapper of `huff.shares` and `shares.total` of the **MCI** package. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

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.

Wieland, T. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

huff, huff.newdest, tcmat.create

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)
```

```
Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

huffmodel1 <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline)
# Huff Model with standard parameters
# uses given transport costs matrix (tcmat_haslach_airline)

huffmodel1a <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline,
atype = "pow", gamma = 0.9, dtype = "pow", lambda = -2.1)
# Alternative weighting parameters

## Not run:
huffmodel2 <- huff.newdest(huffmodel1, newdest.id = "LM_NEU",
newdest.addr = "Bettackerstrasse 3, Freiburg, Germany",
newdest.attract = 1500)
# Adding a new destination with a given street address
# Recalculation of the Huff Model
# Needs internet access and accesses OpenStreetMap server(s)
# If server not available, the function will produce an error

## End(Not run)

huffmodel3 <- huff.updest (huffmodel1, dest.id = "LM01",
dest.attrac = 1200)
# Update the attraction value of one grocery stores
# Recalculation of the Huff Model

## Not run:
model.export(huffmodel3, "interactionmatrix", "totals")
# export as CSV files
```

```
## End(Not run)
```

---

mci *Fitting the MCI model*

Description

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

Usage

```
mci(mcimat, shares, ..., no.intercept = TRUE,
mci.weighting = FALSE, mci.weights = NULL,
show_proc = FALSE)
```

Arguments

mcimat	Object (list) created by the function mcimat.create (Creating an MCI interaction matrix based on raw data)
shares	Column containing the local market shares
...	Explanatory variables (partial utilites)
no.intercept	Logical argument that indicates if an intercept is included into the model
mci.weighting	Logical argument that indicates if weighted least squares (WLS) should be used for fitting the model
mci.weights	If mci.weighting = TRUE: optional weighting vector for the WLS fitting. If mci.weights = NULL, the reciprocals of the residuals are used as weightings
show_proc	Logical argument that indicates if the function prints messages about the state of process during the work

Details

This function calculates a Multiplicative Competitive Interaction (MCI) Model based on a given interaction matrix.

Value

A mcimodel list (invisible) containing the following components:

regdata	Log-centering transformed interaction matrix (data frame)
mcimodel_coef	A matrix containing the regression results (parameters, std. errors, t statistics,...)
mcimodel_stat	A matrix containing the regression model diagnostics (R-squared, adj. R-squared, F statistic,...)

Note

The function is a wrapper of `mci.transmat` and `mci.fit` of the **MCI** package. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

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. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

`mcimat.create`

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
```

```
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`mci.sim` *MCI model market simulation*

Description

Market simulation using the MCI model

Usage

```
mci.sim(mcimodel, origins.pot, ...,
mcitrans = "lc", interc = NULL)
```

Arguments

mcimodel	Object (list) created by the function mci
origins.pot	The column representing the customer potential in the origins in the interaction matrix in mcimodel
...	Explanatory variables (partial utilites) and their corresponding weighting parameters (variable1, parameter1, variable2, parameter2, ...)
mcitrans	Type of MCI transformation: Log-centering transformation (mcitrans = "lc"), or, e.g. in the case of using dummy variables, inverse log-centering transformation (mcitrans = "ilc")
interc	Intercept to be included

Details

This function calculates a market simulation using the Multiplicative Competitive Interaction (MCI) Model based on a given MCI model.

Value

A mcimodel list (invisible) containing the following components:

mcimat	MCI interaction matrix (data frame), also containing the local market shares (p_ij)
mcitotal	Total location market areas (data frame), also containing the total market areas (T_j)

Note

The function is a wrapper of mci.shares and shares.total of the **MCI** package. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

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. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

mci, mcimat.create

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times
```

```
# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)

if (!require("MCI")) install.packages("MCI")
# Needed for function var.asdummy from package MCI

# Integration of chains as dummy variables
chaindummies <- var.asdummy(HaslachStores$LM_Name)
HaslachStores <- cbind(HaslachStores, chaindummies)

mcimat_haslach2 <- mcimat.create (rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), remDest = "LM00", corObserved = 0.1,
origvar.data = HaslachDistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")
```

```
mcimodel_haslach2 <- mci (mcimat = mcimat_haslach2, shares = "p_ij_obs_LME",
"d_ij", "LM_VKF", "Aldi_DUMMY", "Edeka_DUMMY", "Netto_DUMMY",
"Lidl_DUMMY", "Real_DUMMY")
# same analysis as above, but including chain dummies

mcisim_haslach2 <- mci.sim (mcimodel_haslach2, origins.pot = "WO_Einwohner",
"d_ij", -3.1627045, "LM_VKF", 4.5001070,
"Aldi_DUMMY", 0.6012442, "Edeka_DUMMY", 0.4078891, "Netto_DUMMY", -0.4179182,
"Lidl_DUMMY", 1.3272989, "Real_DUMMY", -2.8156859,
mcitrans = "ilc", interc = NULL)
# MCI simulation based on mcimodel_haslach2
# "d_ij", -3.1627045, "LM_VKF", 4.5001070, "Aldi_DUMMY", 0.6012442, ...
# = explanatory variables and their corresponding weighting parameters
# mcitrans = "ilc": inverse log-centering transformation
```

`mcimat.create` *Creation of a MCI interaction matrix*

Description

Creation of an interaction matrix with local market shares (p_{ij}) of each location (j) in each customer origin (i) based on the frequencies in the raw data (e.g. household or POS survey).

Usage

```
mcimat.create(rawdata, origins.id, destinations.id, ..., tcmat,
origvar.data = NULL, origvardata.id = NULL,
destvar.data = NULL, destvardata.id = NULL,
remOrig = NULL, remDest = NULL, corObserved = 0,
remNA = TRUE)
```

Arguments

`rawdata`	Raw data (`data.frame`), e.g. customer survey
`origins.id`	Vector of customer origins
`destinations.id`	Vector of destinations (stores, locations)
`...`	other numeric variables in the raw data which were observed and shall be used to calculate market shares (e.g. expenditures)
`tcmat`	Object (`list`) created by the function `tcmat.create` (Transportation costs matrix)
`origvar.data`	Optional: additional data (`data.frame`) concerning the customer origins
`origvardata.id`	Optional: customer origins in the additional origins data
`destvar.data`	Optional: additional data (`data.frame`) concerning the destinations
`destvardata.id`	Optional: destinations in the additional destinations data
`remOrig`	Optional: vector of origins to be removed from the analysis

remDest	Optional: vector of destinations to be removed from the analysis
corObserved	numeric value which is added to the absolute values before calculating market shares (default: 0)
remNA	Logical argument that indicates if NA values of the origins and destinations are removed or not

Details

This function creates a Multiplicative Competitive Interaction (MCI) Model interaction matrix for further use in the function mci.

Value

A mcimat list (invisible) containing the following components:

mcimat	MCI interaction matrix (data frame), also containing the local market shares (p_ij)
coords	A list containing the coordinates and additional information of the origins (coords_origins) and the destinations (coords_destinations), inherited from the tcmat object
tc.mode	A list containing information about the transportation costs matrix, inherited from the tcmat object
mci.cormode	A list containing information about the processing mode of the interaction matrix (removed origins/destinations etc.)

Note

The function is a wrapper of ijmatrix.create from the **MCI** package. For further information see the **MCI** documentation and the corresponding RJ paper (Wieland 2017).

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. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

mci, tcmat.create

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
```

```
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

# Creating a MCI interaction matrix
# based on raw data of a survey (HaslachSurvey_prepared)
# and a tcmat list object
mcimat_haslach <- mcimat.create(rawdata = HaslachSurvey_prepared,
origins.id = "Wohnort", destinations.id = "LM", "LME",
tcmat = tcmat_haslach_airline,
remOrig = c("SBXXX", "SB613"), corObserved = 0.1,
origvar.data = HaslachDdistricts, origvardata.id = "WO",
destvar.data = HaslachStores, destvardata.id = "LM")

# MCI model based on empirical local market shares
# two explanatory variables: distance (d_ij), store size (LM_VKF)
mcimodel_haslach <- mci (mcimat_haslach, "p_ij", "d_ij", "LM_VKF",
show_proc = TRUE)
```

`model.export` *Output Huff Model results*

Description

Output Huff Model results as .csv file(s)

Usage

```
model.export(huffmodel, mat.filename, total.filename,
decimal = ".", colsep = ";", mat.ascrosstab = TRUE)
```

Arguments

`huffmodel`	Object (`list`) created by the function `huff` (Huff Model)
`mat.filename`	File name for the interaction matrix output (no suffix)
`total.filename`	File name for the total market areas output (no suffix)
`decimal`	the string to use for decimal points (as in `write.table`)
`colsep`	the field separator string (as in `write.table`)
`mat.ascrosstab`	Logical argument that indicates if the interaction matrix should be stored in the form of a crosstab (for further use in GIS)

Value

Two .csv files (interaction matrix, total market areas)

Note

The function is a wrapper of `ijmatrix.crosstab` of the **MCI** package and `write.table` from **utils**. For further information see the package documentations and the corresponding RJ paper (Wieland 2017).

Author(s)

Thomas Wieland

Examples

```
# Compilation of tcmat list from existing datasets:
# (Results from the tcmat.create function)
data(Haslach_tcmatAirline)
# airline distances
data(Haslach_coords_origins)
# Coordinates of origins
data(Haslach_coords_destinations)
# Coordinates of destinationes

# Component "tc.mode":
Airline_tc.mode <- list()
Airline_tc.mode$tc.type = "airline"
Airline_tc.mode$tc.unit = "km"
Airline_tc.mode$tc.constant = 0

# tcmat with airline distances
# Compilation as a list:
tcmat_haslach_airline <- list(tcmat = Haslach_tcmatAirline,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Airline_tc.mode)

Drvtime_tc.mode <- list()
Drvtime_tc.mode$tc.type = "street"
Drvtime_tc.mode$tc.unit = "min"
Drvtime_tc.mode$tc.constant = 0

data(Haslach_tcmatDrvtime)
# car driving times

# tcmat with car driving times
# Compilation as a list:
tcmat_haslach_drvtime <- list(tcmat = Haslach_tcmatDrvtime,
coords_origins = Haslach_coords_origins,
coords_destinations = Haslach_coords_destinations,
tc.mode = Drvtime_tc.mode)

data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)
```

```
huffmodel1 <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline)
# Huff Model with standard parameters
# uses given transport costs matrix (tcmat_haslach_airline)

huffmodel1a <- huff (HaslachDistricts$WO, HaslachDistricts$WO_Einwohner,
HaslachStores$LM, HaslachStores$LM_VKF,
tcmat = tcmat_haslach_airline,
atype = "pow", gamma = 0.9, dtype = "pow", lambda = -2.1)
# Alternative weighting parameters

## Not run:
huffmodel2 <- huff.newdest(huffmodel1, newdest.id = "LM_NEU",
newdest.addr = "Bettackerstrasse 3, Freiburg, Germany",
newdest.attract = 1500)
# Adding a new destination with a given street address
# Recalculation of the Huff Model
# Needs internet access and accesses OpenStreetMap server(s)
# If server not available, the function will produce an error

## End(Not run)

huffmodel3 <- huff.updest (huffmodel1, dest.id = "LM01",
dest.attrac = 1200)
# Update the attraction value of one grocery stores
# Recalculation of the Huff Model

## Not run:
model.export(huffmodel3, "interactionmatrix", "totals")
# export as CSV files

## End(Not run)
```

`rawdata.prep` *Preparation of survey raw data*

Description

Preparation of survey raw data for further use in a MCI interaction matrix.

Usage

```
rawdata.prep(cols.below1, cols.below2, cols.keep, colnames.new)
```

Arguments

`cols.below1`	First list of columns in a survey data frame to arrange one below the other (e.g. destination IDs)

cols.below2	Second list of columns in a survey data frame to arrange one below the other (e.g. obtained destination purchases)
cols.keep	Single column in a survey data frame that must be kept for all data (e.g. IDs of customer origins)
colnames.new	Character vector of new column names

Details

This function re-orders survey raw data for further use in the function mcimat.create. Sometimes, in a household survey, two or three destinations/stores are obtained instead of one, always arranged in different columns. This function re-orders survey raw data in single columns for further use in the function mcimat.create.

Value

A data frame containing three columns named corresponding to colnames.new

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.

Wieland, T. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

See Also

mcimat.create

Examples

```
data(HaslachSurvey)
# survey raw data (Store choices and purchases)

# Preparing raw data (HaslachSurvey)
HaslachSurvey_prepared <- rawdata.prep (cols.below1 =
list(HaslachSurvey$LM1_Text, HaslachSurvey$LM2_Text),
cols.below2 = list(HaslachSurvey$LM1E, HaslachSurvey$LM2E),
cols.keep = HaslachSurvey$WO,
colnames.new = c("LM", "LME", "Wohnort"))
# "WO" and "Wohnort" = origin ID
# "LM1_Text", "LM2_Text" and "LM" = destination IDs (grocery stores)
# "LM1E", "LM2E" and "LME" = grocery store purchases

HaslachSurvey_prepared
```

`tcmat.create` *Creation of a transport costs matrix*

Description

Creation of a transport costs matrix (airline distance, car travel time)

Usage

```
tcmat.create(origins.id, origins.addr,
destinations.id, destinations.addr,
tc.type = "airline", tc.unit = "km",
or.addr.format = "stradr", de.addr.format = "stradr",
tc.constant = 0, show_proc = FALSE)
```

Arguments

`origins.id`	ID of the origins
`origins.addr`	Address of the origins, either street addresses (e.g. "Markgrafenstrasse 68, Freiburg, Germany") or coordinates (X_LON;X_LAT, e.g.: "7,82218;47,99387")
`destinations.id`	ID of the destinations
`destinations.addr`	Address of the destinations, either street addresses (e.g. "Markgrafenstrasse 68, Freiburg, Germany") or coordinates (X_LON;X_LAT, e.g.: "7,82218;47,99387")
`tc.type`	Type of transportat costs: `"airline"` (airline distance) or `"street"` (car travel time)
`tc.unit`	If `tc.type = "airline"`: unit of distance ("m", "km" or "miles")
`or.addr.format`	Origins address format. Default: `"stradr"` (street addresses))
`de.addr.format`	Destinations address format. Default: `"stradr"` (street addresses))
`tc.constant`	Adding a constant to the calculated/queried values (default: 0)
`show_proc`	Logical argument that indicates if the function prints messages about the state of process during the work

Details

Creation of a `tcmat list` object (Transport costs matrix) for further use in `huff` and `mcimat.create`. Included geocoding of street addresses (from `geocode_OSM` in **tmaptools**) and travel time query (from `ormsTable` in **osrm**) or the calculation of airline distances (from `dist.mat` in **REAT**).

Value

A `tcmat list` (invisible) containing the following components:

`coords_origins`
: A `data.frame` containing the coordinates (lat, lon) of the origins

`coords_destinations`
: A `data.frame` containing the coordinates (lat, lon) of the destinations

`tcmat`
: A `data.frame` containing the transport costs matrix

`tc.mode`
: A `list` containing information about the transportation costs matrix

Note

This function is wrapper of the functions `geocode_OSM` (for geocoding street addresses using OSM Nominatim) from the **tmaptools** package (Tennekes 2018), `osrmTable` (for creating travel time matrices between points while interfacing the table OSRM service) from the **osrm** package (Giraud 2018) and `dist.mat` (for creating airline distances between points) from the **REAT** package (Wieland 2018). The OSM-related functions rely on the usage of a running OSRM service! For more information, see the related package documentations.

Author(s)

Thomas Wieland

References

Giraud, T. (2018): osrm: Interface Between R and the OpenStreetMap-Based Routing Service OSRM. R package version 3.1.1. https://CRAN.R-project.org/package=osrm.

Tennekes, M. (2018): tmaptools: Thematic Map Tools. R package version 2.0-1. https://CRAN.R-project.org/package=tmaptools.

Wieland, T. (2017): "Market Area Analysis for Retail and Service Locations with MCI". In: *The R Journal*, **9**, 1, p. 298-323. https://journal.r-project.org/archive/2017/RJ-2017-020/RJ-2017-020.pdf.

Wieland, T. (2018): REAT: Regional Economic Analysis Toolbox. R package version 2.1.0. https://CRAN.R-project.org/package=REAT.

See Also

`huff`, `mcimat.create`

Examples

```
data(HaslachSurvey)
# survey raw data (Store choices and purchases)
data(HaslachDistricts)
# IDs and information about customer origins
data(HaslachStores)
# IDs and information about destinations (grocery stores)
```

```
## Not run:
tcmat_haslach_airline <- tcmat.create(origins.id = HaslachDistricts$WO,
origins.addr = HaslachDistricts$WO_Adr, destinations.id = HaslachStores$LM,
destinations.addr = HaslachStores$LM_Adr_zus,
tc.type = "airline", tc.unit = "km",
addr.format = "stradr", tc.constant = 0, show_proc = TRUE)
# Creation of a transport costs matrix with airline distances
# saving as list object "tcmat_haslach_airline"
# Needs internet access and accesses OpenStreetMap server(s)
# If server not available, the function will produce an error

tcmat_haslach_drvtime <- tcmat.create(origins.id = HaslachDistricts$WO,
origins.addr = HaslachDistricts$WO_Adr, destinations.id = HaslachStores$LM,
destinations.addr = HaslachStores$LM_Adr_zus,
tc.type = "street", addr.format = "stradr",
tc.constant = 0, show_proc = TRUE)
# Creation of a transport costs matrix with car driving times
# saving as list object "tcmat_haslach_drvtime"
# Needs internet access and accesses OpenStreetMap server(s)
# If server not available, will produce an error

## End(Not run)
```

`to.tcmat` *Creation of a transport costs matrix*

Description

Creation of a `tcmat list` object from a given transport costs table

Usage

```
to.tcmat(dataset, colname.from, colname.to, colname.tc)
```

Arguments

`dataset`	A `data frame` containing the transport costs table
`colname.from`	Name of the column containing the origin IDs
`colname.to`	Name of the column containing the destination IDs
`colname.tc`	Name of the column containing the transport costs (e.g. airline distances, travel times)

Details

Creation of a `tcmat list` object (Transport costs matrix) from a given table for further use in `huff` and `mcimat.create`.

Value

A `tcmat list` (invisible) containing the following components:

`coords_origins`
: A `data.frame`

`coords_destinations`
: A `data.frame`

`tcmat`
: A `data.frame` containing the transport costs matrix

`tc.mode`
: A `list` containing information about the transportation costs matrix

Author(s)

Thomas Wieland

See Also

`huff`, `mcimat.create`, `tcmat.create`

Examples

```
data(Haslach_tcmatDrvtime)
# Travel times from the Haslach example
Haslach_tcmat <- to.tcmat (Haslach_tcmatDrvtime, "from", "to", "tc")
# Creating new tcmat object
Haslach_tcmat
```

Index