

Package ‘meltt’

August 26, 2017

Type Package

Title Matching Event Data by Location, Time and Type

Version 0.3.2

Date 2017-08-26

Author Karsten Donnay and Eric Dunford

Maintainer Karsten Donnay <kdonnay@gmx.net>

Description Framework for merging and disambiguating event data based on spatiotemporal co-occurrence and secondary event characteristics. It can account for intrinsic “fuzziness” in the coding of events, varying event taxonomies and different geo-precision codes.

License LGPL-3

Depends R (>= 2.6)

Imports sp, reticulate, graphics, scales, ggmap, ggplot2, plotGoogleMaps, plyr

SystemRequirements Python (>= 2.7)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-26 08:40:52 UTC

R topics documented:

meltt-package	2
crash_data1	3
crash_data2	4
crash_data3	4
crash_taxonomies	5
is.meltt	6
meltt	7
meltt.data	10
meltt.disambiguate	11
meltt.duplicates	12
meltt.episodal	13

meltt.inspect	15
meltt.match	16
meltt.taxonomy	17
mplot	18
plot.meltt	20
print.meltt	21
summary.meltt	21
tplot	22

Index	24
--------------	-----------

meltt-package	<i>meltt: Matching Event Data by Location, Time, and Type</i>
---------------	---

Description

meltt is a framework for merging and disambiguating event data based on spatiotemporal co-occurrence and secondary event characteristics. It can account for intrinsic "fuzziness" in the coding of events, varying event taxonomies and different geo-precision codes.

Details

The meltt function iteratively matches multiple datasets by isolating proximate events based on a user-specified spatio-temporal window to determine co-occurrence. It then assesses potential matches by leveraging secondary event characteristics formalized as user-specified input taxonomies.

Note

meltt requires the Python package NumPy to run. The package automatically checks whether NumPy is installed at runtime and returns an error if it is not.

Author(s)

Karsten Donnay and Eric Dunford

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#), [meltt.data](#), [meltt.duplicates](#), [meltt.inspect](#), [tplot](#), [mplot](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
summary(output)
plot(output)
tplot(output,time.unit = 'days')

## End(Not run)
```

crash_data1

Dataset to illustrate the functionality of meltt

Description

This artificial dataset illustrates how **meltt** can be used to automatically integrate and disambiguate event data. It contains timing and location information about (simulated) car crashes for one month (Jan. 2012) in the state of Maryland, U.S., with information about the model, color, and type of accident.

Usage

```
data(crashMD)
```

Format

A data.frame containing observations.

Author(s)

Karsten Donnay and Eric Dunford.

Source

Simulated data.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

`crash_data2`*Dataset to illustrate the functionality of meltt*

Description

This artificial dataset illustrates how **meltt** can be used to automatically integrate and disambiguate event data. It contains timing and location information about (simulated) car crashes for one month (Jan. 2012) in the state of Maryland, U.S., with information about the model, color, and type of accident.

Usage

```
data(crashMD)
```

Format

A `data.frame` containing observations.

Author(s)

Karsten Donnay and Eric Dunford.

Source

Simulated data.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

`crash_data3`*Dataset to illustrate the functionality of meltt*

Description

This artificial dataset illustrates how **meltt** can be used to automatically integrate and disambiguate event data. It contains timing and location information about (simulated) car crashes for one month (Jan. 2012) in the state of Maryland, U.S., with information about the model, color and, and type of accident.

Usage

```
data(crashMD)
```

Format

A data.frame containing observations.

Author(s)

Karsten Donnay and Eric Dunford.

Source

Simulated data.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

crash_taxonomies

Taxonomies to illustrate the functionality of meltt

Description

These taxonomies formalize how the information about model, color, and type of accident in our three artificial car crash datasets map onto one another.

Usage

```
data(crashMD)
```

Format

A list of three data.frame containing information about the different categories (specific to general) of models, colors and degree of damages coded in each dataset.

Author(s)

Karsten Donnay and Eric Dunford.

Source

Simulated data.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

`is.meltt`*Tests for objects of type meltt.*

Description

Function returns logical statement whether an object is of class `meltt`.

Usage

```
is.meltt(object)
```

Arguments

`object` object to be tested.

Value

`is.meltt` returns TRUE or FALSE depending on whether its argument is of type `meltt` or not.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
is.meltt(output)

## End(Not run)
```

meltt

Matching Event Data by Location, Time and Type

Description

`meltt` merges and disambiguates event data based on spatiotemporal co-occurrence and secondary event characteristics. It can account for intrinsic "fuzziness" in the coding of events through the incorporation of user-specified taxonomies and adjusts for different degrees of geospatial and temporal precision by allowing for the specification of spatiotemporal "windows".

Usage

```
meltt(..., taxonomies, twindow, spatwindow, smartmatch=TRUE,
       certainty=NA, partial=FALSE, averaging=FALSE, weight=NA)
```

Arguments

<code>...</code>	input datasets. See Details.
<code>taxonomies</code>	list of user-specified taxonomies. Taxonomies map onto a specific variable in the input data that contains the same name as the input taxonomy. See Details.
<code>twindow</code>	specification of temporal window in unit days. See Details.
<code>spatwindow</code>	specification of a spatial window in kilometers. See Details.
<code>smartmatch</code>	implement matching using all available taxonomy levels. When false, matching will occur only on a specified taxonomy level. Default = TRUE. See Details.
<code>certainty</code>	specification of the the exact taxonomy level to match on when <code>smartmatch = FALSE</code> . Default = NULL. See Details.
<code>partial</code>	specifies whether matches along only some of the taxonomy dimensions are permitted. Default = FALSE. See Details.
<code>averaging</code>	implement averaging of all values events are match on when matching across multiple dataframes. Default = FALSE. See Details.
<code>weight</code>	specified weights for each taxonomy level to increase or decrease the importances of each taxonomy's contribution to the matching score. Default = NA. See Details.

Details

`meltt` expects input datasets to be of class `data.frame` and input taxonomies to be of class `list`, which contain one or more taxonomy data frames. Each taxonomy must have a column denoting the "base.category" (i.e. the version of the variable that appears in each data frame) and a "data.source" column that matches the object name of the dataset containing those variables. All subsequent column in each taxonomy denote the user-specified levels of generalization, which capture the degree to which the taxonomy category generalizes out. The most left column must contain the most granular levels while the furthest right the broadest. Error will be issued if taxonomy levels are not in the correct order.

The `twindow` and `spatwindow` inputs specify the temporal and spatial dimensions for which entries are considered to be spatio-temporally proximate, and with that, potential matches (i.e. duplicate entries). For all potential matches, `meltt` then leverages the secondary information about events (formalized through the mapping of categories specified in taxonomies) to identify most likely matches.

`meltt` by default uses `smartmatch`, which leverages all taxonomy levels, i.e., establishes agreement on any taxonomy level while discounting inferior (i.e. more coarse) agreement using a matching score. When `smartmatch` is set to `false`, a `certainty` must be set, specifying which taxonomy level two events must agree on to be considered a match.

`partial` permits matches along less than the full number of taxonomy dimensions specified. In this case, every dimension not matched is assigned the worst matching score in the calculation of the overall fit. `averaging` allows for users to take the average of all input information (date, longitude, latitude, taxonomy, etc.) when merging more than one dataset. When set to `FALSE`, events use the input information of the first or most left dataset in the order the data was received.

`weight` allows to weigh matches for different taxonomies in order to discount one (or several) event dimensions compared to others or vice versa. If `weight=NA` the package assumes homogeneous weights of 1. If weights are manually specified they must sum up to the total number of taxonomy dimensions used, i.e., the normalized overall weight always has to be 1. If not, the package returns an error.

Value

Returns an object of class "meltt".

The functions `summary`, `print`, `plot` overload the standard outputs for objects of type `meltt` providing summary information and visualizations specific to the output object. The generic accessor functions `meltt.data`, `meltt.duplicates`, `tplot`, `mplot` extract various useful features of the integrated data frame: the unique de-duplicated entries, all duplicate entries (or matches), a histogram of the temporal distribution and a map of the integrated output.

An object of class "meltt" is a list containing at least the following components. First, a list named "processed" that contains all outputs of the integration process:

`complete_index` a `data.frame` of initial input data (location information, time stamp, and secondary criteria) converted to a numeric matrix. The matrix is what is processed by the `meltt` algorithm.

`deduplicated_index` a posterior `data.frame` of initial input data converted to a numeric matrix with duplicate entries have been removed. It further contains information about "episodal events" (i.e. events that span more than one time unit with an end and start date) that potentially match to unique events but could not be automatically assigned as matches (or not).

`event_matched` Numeric matrix containing indices for each matching event from each input dataset. The leading data set is the furthest left, every matching event to its right is identified as a duplicate of the initial entry and is removed.

`event_contenders` Numeric matrix containing indices for each "runner up" event from each input dataset that was identified as a potential but less optimal match based on its matching score.

episode_matched	Numeric matrix containing indices for each matching "episodes" (i.e. events that span more than one time unit with an end and start date) from each input dataset. Only contains matches between episodes. Matches between events and episodes must be manually reviewed by users (see meltt.inspect).
episode_contenders	Numeric matrix containing indices for each "runner up" episodes from each input dataset that was identified as a potential but less optimal match based on its matching score.

Second, it contains a comprehensive summary of the input data, parameters and taxonomy specifications. Specifically it returns:

inputData	List containing the original object name and information of the input data prior to integration.
parameters	List containing information on all input parameters on which the data was integrated.
inputDataNames	Vector of the object names of the input datasets. These names are carried through the integration process to differentiate between input datasets. The index keys contained in the numeric matrix representations of the data follow the order the data was entered.
taxonomy	List containing the taxonomy (secondary assumption criteria) datasets used to integrate the input data. The list contains: the names of the taxonomies (which must match the names of the variables they seek to generalize in the input data), an integer of the number of input taxonomies, a vector containing information on the depth (i.e. the number of columns) of each taxonomy, and a list of the original input taxonomies.

Note

meltt requires the Python package NumPy to run. The package automatically checks whether NumPy is installed at runtime and returns an error if it is not.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt.data](#), [meltt.duplicates](#), [meltt.inspect](#), [tplot](#), [mplot](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
summary(output)
plot(output)

# Extract De-duplicated events
dataset <- meltt.data(output)
head(dataset)

## End(Not run)
```

meltt.data

Returns de-duplicated entries from meltt output.

Description

meltt.data returns all unique, de-duplicated entries across all input datasets. Function provides a dataset where all overlapping, duplicate entries are removed, offering a version of the input data with no redundancies.

Usage

```
meltt.data(object,columns=NULL)
```

Arguments

object	object of class <code>meltt</code> .
columns	string vector referencing column names located in the input data. Default is to return all location, time stamp, and taxonomy columns the data was evaluated on.

Details

meltt.data returns all unique entries along with specified columns. Function allows for easy extraction all de-duplicated entries.

Value

Returns an data.frame where the first columns contains the name of the original input data object from which the data entry was drawn, and a unique event ID. The subsequent columns are all columns specified in the columns argument, or location, time stamp, and taxonomy columns the data was evaluated on columns = NULL.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#), [meltt.duplicates](#), [meltt.inspect](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
dataset <- meltt.data(output,column=c("date","longitude","latitude"))
head(dataset)

# Return all original columns
dataset <- meltt.data(output,return_all = T)

## End(Not run)
```

meltt.disambiguate *Tracking and indexing of matching entries [Auxiliary Function]*

Description

Auxiliary function used within `meltt` to index events as multiple datasets are processed.

Usage

```
meltt.disambiguate(data,match_output,indexing,priormatches,averaging)
```

Arguments

<code>data</code>	data to be disambiguated passed as <code>data.frame</code> object from <code>meltt</code> . See details.
<code>match_output</code>	<code>data.frame</code> object of identified matches passed from <code>meltt.match</code> . See details.
<code>indexing</code>	<code>data.frame</code> object passed from <code>meltt</code> that specifies the correct index for every event and dataset. See details.
<code>priormatches</code>	prior matches (if any) passed as <code>data.frame</code> . See details.
<code>averaging</code>	specification if common information among matches should be averaged. Passed from <code>meltt</code> . See details.

Details

Auxiliary function used within `meltt` to index events as multiple datasets are processed. Function keeps track of matching and non-matching events as each subsequent `data.frame` is processed. Using the identified matches from the `meltt.match` output, `meltt.disambiguate` merges the matches in the data and indexes the match. `indexing` is used for correct labeling of matches in case more than two datasets are merged. `averaging` averages the common information between matching events. The parameter is specified within the main function `meltt`.

Value

`meltt.disambiguate` returns a list containing two object: a data frame with all located matches paired and a new index, specifying the data frame as a single frame, and a running index of all matched events.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#)

<code>meltt.duplicates</code>	<i>Return identified duplicate entries removed after integration.</i>
-------------------------------	---

Description

`meltt.duplicates` returns all matching entries that are identified as matches during the integration process.

Usage

```
meltt.duplicates(object, columns=NULL)
```

Arguments

<code>object</code>	object of class meltt .
<code>columns</code>	string vector referencing column names located in the input data. Default is to return all columns contained in the input data.

Details

`meltt.duplicates` returns all duplicated entries along with specified columns to compare which entries matched. Function allows for easy extraction all entries identified as duplicates.

Value

Returns an `data.frame` where the first columns contain an index for the `data.source` and event for each data frame. The subsequent columns are all columns specified in the `columns` argument, or all columns contained in the original input data if `columns = NULL`.

An "event_type" column is added to the output `data.frame` specifying if the match was between events or episode. See [meltt.inspect](#) for handling flagged event-to-episode matches.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#), [meltt.data](#), [meltt.inspect](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
duplicates <- meltt.duplicates(output,column=c("date","longitude","latitude"))
head(duplicates)

## End(Not run)
```

meltt.episodal

Handling events and episodal data [Auxiliary Function]

Description

Auxiliary function that receives the compilation matrix and systematically subsets events and episodes to deal with differences in event duration. The function passes subsets to `meltt.matchto` to be processed. Output includes a full list of matching events and/or episodes.

Usage

```
meltt.episodal(data, indexing, priormatches, twindow, spatwindow,
smartmatch, certainty, k, secondary, partial, averaging, weight)
```

Arguments

data	object of class data.frame.
indexing	list of indices given the entry location of events and episodes in the original input data.
priormatches	prior matches (if any) passed as data.frame.
twindow	specification of temporal window in unit days.
spatwindow	specification of a spatial window in kilometers.
smartmatch	implement matching using all available taxonomy levels. When false, matching will occur only on a specified taxonomy level. Default = TRUE.
certainty	specification of the the exact taxonomy level to match on when smartmatch = F. Default = NULL.
k	number of taxonomies passed from meltt.
secondary	vector of the number of taxonomy levels for each taxonomy passed from meltt.
partial	boolean for wether or not partial matches are permitted. Passed from meltt.
averaging	implement averaging of all values events are match on when matching across multiple dataframes. Default = FALSE.
weight	relative weight of each taxonomy in the calculation of the matching score. Passed from meltt.

Details

Internal function that helps manage integration of event and episodal data by easing the transition between the two logics. meltt algorithm tracks event-to-event matches, episode-to-episode, and event-to-episode matches. meltt.episodal streamlines the transfer between these matching states.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#)

meltt.inspect	Returns flagged event-to-episode matches for review.
---------------	--

Description

`meltt.inspect` returns all episode entries that were flagged to match to an event. Function provides a list containing each flagged event and episode to ease comparison and assessment. All flagged entries should be manually reviewed to determine the validity of the match.

If an flagged event-to-episode is determined to be a match, the duplicate can be removed by providing a Boolean vector to the `confirmed_matches` argument. All TRUE episodes will be removed as duplicates, retaining only the event entry.

Usage

```
meltt.inspect(object, columns=NULL, confirmed_matches=NULL)
```

Arguments

<code>object</code>	object of class <code>meltt</code> .
<code>columns</code>	string vector referencing column names located in the input data. Default is to return all location, time stamp, and taxonomy columns the data was evaluated on.
<code>confirmed_matches</code>	boolean vector specifying entries to be removed from deduplicated set. Function returns a <code>data.frame</code> of unique, deduplicated entries when specified.

Details

`meltt.inspect` returns all episode entries that were flagged to match to an event. Function provides a list containing each flagged event and episode for easy comparison. Matching event-to-episodes can be cleaned by specifying a boolean vector where TRUE identifies that entry as a duplicate, otherwise FALSE

Value

Returns a list object where each entry in the list contains information on the event and the flagged episode for manual assessment of the match. The information by which the entries are evaluated are specified by the `columns` argument. If `columns = NULL`, location, time stamp, and taxonomy information is reported.

Events and episodes confirmed as duplicate entries can be removed by providing a boolean vector to the `confirmed_matches` argument. A `data.frame` of unique entries (similar to the output of `meltt.data`) will be returned.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#), [meltt.data](#), [meltt.duplicates](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
                taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)

flagged <- meltt.inspect(output)
flagged

retain <- c(TRUE,TRUE,TRUE,TRUE,TRUE)
dataset <- meltt.inspect(output,confirmed_matches=retain)
head(dataset)

## End(Not run)
```

meltt.match

Performing iterative comparison and matching [Auxiliary Function]

Description

Auxiliary function that generates an R wrapper around the main python function used to process the numerical matrix generated in `meltt`. Returns a summary of matched entries.

Usage

```
meltt.match(data, twindow, spatwindow, smartmatch, certainty, k,
            secondary, partial, weight, episodal)
```

Arguments

<code>data</code>	numerical matrix passed from <code>meltt.transform</code> .
<code>twindow</code>	specification of temporal window in unit days passed from <code>meltt</code> .
<code>spatwindow</code>	specification of a spatial window in kilometers passed from <code>meltt</code> .
<code>smartmatch</code>	implement matching using all available taxonomy levels. When <code>FALSE</code> , matching will occur only on a specified taxonomy level passed from <code>meltt</code> .
<code>certainty</code>	specification of the the exact taxonomy level to match on when <code>smartmatch = FALSE</code> passed from <code>meltt</code> .

k	number of taxonomies passed from meltt.
secondary	vector of the number of taxonomy levels for each taxonomy passed from meltt.
partial	boolean for wether or not partial matches are permitted. Passed from meltt.
weight	relative weight of each taxonomy in the calculation of the matching score. Passed from meltt.
episodal	boolean for wether normal or episodal matches are performed. Automatically determined and passed from meltt.

Details

Main Auxiliary wrapper function that passes the processed data matrix from meltt to the python code used to manage the matching procedure.

Value

Returns a list object containing output of matching entries and a matrix of optimal selected matches.

Note

meltt.match requires the Python package NumPy to run. The package automatically checks whether NumPy is installed at runtime and returns an error if it is not.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#)

meltt.taxonomy

Handling of taxonomy inputs [Auxiliary Function].

Description

Auxiliary function that maps secondary taxonomies onto the input data and transforms the taxonomies into a numerical matrices.

Usage

```
meltt.taxonomy(data, taxonomies)
```

Arguments

data	object of class <code>data.frame</code> .
taxonomies	object of class <code>list</code> , containing <code>data.frames</code> of input taxonomies for secondary matching criteria.

Details

`meltt.taxonomy` maps the user-created taxonomies onto the input data, and converts the taxonomy to a numerical matrix. The taxonomies are used as secondary criteria in the matching procedure.

Value

Returns a numerical matrix that contains all data indices, date/enddate, lon/lat, and taxonomies.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#)

mplot	<i>Generate a Google Map plotting the spatial distribution of meltt output data.</i>
-------	--

Description

`mplot` provides both a static and interactive Google Map plotting the spatial distribution of duplicate and unique entries in the integrated data.

Usage

```
mplot(object, interactive = FALSE)
```

Arguments

object	object of class meltt .
interactive	Specify if the Google Map should be interactive. Default = FALSE.

Details

mplot generates a static spatial map using the Google Maps API. The map identifies unique and duplicative (i.e. entries with "matches") entries. The function provides a concise summary of the integration output across the spatial bounds of the geo-referenced input data.

mplot also generates an interactive spatial map that renders in the user's browser.

Value

Returns a Google Map, which renders in the user's browser if `interactive=TRUE`, of all unique event locations. Each unique event from the input data is represented by a unique color. Matching events (i.e. events with detected duplicates) are marked as blue diamonds.

For the interactive map, events where duplicates were identified are flagged as a "Match", and the duplicates for the event are marked in the point meta data. All events are provided a unique identifier which corresponds to the input data's name (i.e. the object name of the input data) and its location in the input frame (i.e. "dataA-4" corresponds to the fourth entry in the data.frame "dataA").

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#), [tplot](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)

mplot(output) # Static Map

mplot(output,interactive=T) # Interactive Map (will render in browser)

## End(Not run)
```

`plot.meltt`*Plot function for objects of class meltt.*

Description

Overloads the default `plot()` for objects of class `meltt`.

Usage

```
## S3 method for class 'meltt'  
plot(x, ...)
```

Arguments

`x` object of class `meltt`.
`...` further arguments passed to or from other methods.

Details

Returns a bar plot outlining the proportion of events that are unique and duplicates from an object of class `meltt`.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

`meltt`

Examples

```
## Not run:  
data(crashMD)  
output <- meltt(crash_data1,crash_data2,crash_data3,  
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)  
plot(output)  
  
## End(Not run)
```

print.meltt	<i>Print function for objects of class meltt.</i>
-------------	---

Description

Overloads the default print() for objects of class meltt.

Usage

```
## S3 method for class 'meltt'  
print(x, ...)
```

Arguments

x	object of class meltt .
...	further arguments passed to or from other methods.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#)

summary.meltt	<i>Summary function for objects of class meltt.</i>
---------------	---

Description

Overloads the default summary() for objects of class meltt.

Usage

```
## S3 method for class 'meltt'  
summary(object, ...)
```

Arguments

object object of class `meltt`.
 ... further arguments passed to or from other methods.

Value

Prints a number of summary statistics regarding inputs (datasets, spatial and temporal windows, taxonomies) and observations (unique, matching, duplicate entries removed). It also prints and returns a `data.frame` summarizing the overlap among datasets, i.e., how many entries in any one dataset match up to entries in one or more of the other.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

`meltt`

tplot	<i>Temporal distribution of meltt output data.</i>
-------	--

Description

tplot provides a histogram of integrated data that plots the temporal distribution of duplicate and unique entries

Usage

```
tplot(object, time.unit="month")
```

Arguments

object object of class `meltt`.
 time.unit string specifying the temporal bin: "day", "week", "month", and "year". Default = "month".

Details

tplot generates a temporal histogram that identifies unique entries after duplicates are removed and a reverse temporal histogram charting the distribution of duplicate entries. The function provides a concise summary of the integration output across the input time period presented in a relevant unit.

Value

Returns a histogram plot where the y-axis is a frequency capturing the total number of events for that time period, and the x-axis is time.

Author(s)

Karsten Donnay and Eric Dunford.

References

Karsten Donnay, Eric Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2016). "MELTT: Matching Event Data by Location, Time and Type." Paper presented at *Annual Conference of the Midwest Political Science Association*, April 7-10 2016, Chicago.

See Also

[meltt](#), [mplot](#)

Examples

```
## Not run:
data(crashMD)
output <- meltt(crash_data1,crash_data2,crash_data3,
                taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
tplot(output,time.unit="day")

## End(Not run)
```

Index

crash_data1, [3](#)
crash_data2, [4](#)
crash_data3, [4](#)
crash_taxonomies, [5](#)

is.meltt, [6](#)

meltt, [2](#), [6](#), [7](#), [10–23](#)
meltt-package, [2](#)
meltt.data, [2](#), [8](#), [9](#), [10](#), [13](#), [16](#)
meltt.disambiguate, [11](#)
meltt.duplicates, [2](#), [8](#), [9](#), [11](#), [12](#), [16](#)
meltt.episodal, [13](#)
meltt.inspect, [2](#), [9](#), [11](#), [13](#), [15](#)
meltt.match, [16](#)
meltt.taxonomy, [17](#)
mplot, [2](#), [8](#), [9](#), [18](#), [23](#)

plot.meltt, [20](#)
print.meltt, [21](#)

summary.meltt, [21](#)

tplot, [2](#), [8](#), [9](#), [19](#), [22](#)