# Package 'meltt'

October 27, 2022

**Type** Package

**Title** Matching Event Data by Location, Time and Type

**Version** 0.4.3

**Date** 2022-10-22

**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** utils, ggplot2, plyr, dplyr, reticulate, Rcpp, shiny, shinyjs, tidyr, leaflet, tibble, methods

**SystemRequirements** Python (>= 3.6)

**NeedsCompilation** yes

**LinkingTo** Rcpp, RcppArmadillo

**Repository** CRAN

**Date/Publication** 2022-10-26 22:22:36 UTC

## R topics documented:

---

`meltt-package` *meltt: Matching Event Data by Location, Time, and Type*

---

## 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.

## Author(s)

Karsten Donnay and Eric Dunford

## References

Karsten Donnay, Eric T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

`meltt`, `meltt_data`, `meltt_duplicates`, `meltt_inspect`, `tplot`, `mplot`

## Examples

```
data(crashMD)
output = meltt(crash_data1, crash_data2, crash_data3,
               taxonomies = crash_taxonomies, twindow = 1, spatwindow = 3)
plot(output)
tplot(output, time_unit = 'days')
```

## 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

---

`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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

`meltt`

## Examples

```
data(crashMD)
output = meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
is.meltt(output)
```

---

`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 = 0, averaging = FALSE, weight = NA, silent = FALSE)
```

## Arguments

| | |
|---|---|
| `...` | input datasets. See Details. |
| `taxonomies` | 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. |
| `twindow` | specification of temporal window in unit days. See Details. |
| `spatwindow` | specification of a spatial window in kilometers. See Details. |

| | |
|---|---|
| smartmatch | implement matching using all available taxonomy levels. When false, matching will occur only on a specified taxonomy level. Default = TRUE. See Details. |
| certainty | specification of the the exact taxonomy level to match on when smartmatch = FALSE. Default = NULL. See Details. |
| partial | specifies whether matches along less than the full taxonomy dimensions are permitted. Default = 0. See Details. |
| averaging | implement averaging of all values events are match on when matching across multiple dataframes. Default = FALSE. See Details. |
| weight | 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. |
| silent | Boolean specifying whether or not messages are displayed. Default = FALSE. |

## Details

meltt expects input datasets to be of class data.frame. Minimally each data must have columns "date" (formatted as "YYYY-mm-dd" or "YYYY-mm-dd hh:mm:ss"), "longitude" and "latitude" (both in degree; we assume global coordinates formatted in WGS-84) and the columns representing the dimensions used in the matching taxonomies. Note that meltt requires at least two datasets as input and can otherwise, in principle, handle any number of datasets.

The input taxonomies is expected 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 (i.e., 1 for the base level of the taxonomy, 2 for the next broader level etc.) two events must agree on to be considered a match.

partial specifies the number of dimensions along which no matching information is permitted for events to still be considered a potential match. In this case, every dimension not matched is assigned the worst matching score in the calculation of the overall fit. By default, all dimensions are considered, i.e. partial=0. 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 the 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 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.

## Author(s)

Karsten Donnay and Eric Dunford.

## References

Karsten Donnay, Eric T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

`meltt_data`, `meltt_duplicates`, `meltt_inspect`, `tplot`, `mplot`

## Examples

```
data(crashMD)
output = meltt(crash_data1, crash_data2, crash_data3,
                taxonomies = crash_taxonomies, twindow = 1, spatwindow = 3)
plot(output)

# Extract De-duplicated events
dataset = meltt_data(output)
head(dataset)
```

---

# `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

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

### Details

Auxilary 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

### See Also

meltt

---

## 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 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, silent)
```

## 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 | specifies whether matches along less than the full taxonomy dimensions 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. |
| silent | Boolean specifying whether or not messages are displayed. 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt

`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, silent)
```

## Arguments

| | |
|---|---|
| `data` | numerical matrix passed from `meltt.transform`. |
| `twindow` | specification of temporal window in unit days passed from `meltt`. |
| `spatwindow` | specification of a spatial window in kilometers passed from `meltt`. |
| `smartmatch` | implement matching using all available taxonomy levels. When `FALSE`, matching will occur only on a specified taxonomy level passed from `meltt`. |
| `certainty` | specification of the the exact taxonomy level to match on when `smartmatch = FALSE` passed from `meltt`. |
| `k` | number of taxonomies passed from `meltt`. |
| `secondary` | vector of the number of taxonomy levels for each taxonomy passed from `meltt`. |
| `partial` | specifies whether matches along less than the full taxonomy dimensions 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`. |
| `silent` | Boolean specifying whether or not messages are displayed. 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

### 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 data.frame. |
| taxonomies | object of class list, containing data.frames 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, longitude/latitude, and taxonomies.

### Author(s)

Karsten Donnay and Eric Dunford.

### References

Karsten Donnay, Eric T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt

---

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, return_all = FALSE)
```

## Arguments

| | |
|---|---|
| object | object of class meltt. |
| 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. |
| return_all | Boolean specifying whether all columns in any of the original data should be returned. Default = FALSE. |

## 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt, meltt_duplicates, meltt_inspect

## Examples

```
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 = TRUE)
```

---

# meltt_duplicates *Return identified duplicate entries removed after integration.*

---

## Description

meltt_duplicates returns all matching entries that are identified as matches during the integration process.

## Usage

```
meltt_duplicates(object, columns = NULL)
```

## Arguments

| | |
|---|---|
| object | object of class meltt. |
| columns | 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt, meltt_data, meltt_inspect

## Examples

```
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)
```

---

# 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

| | |
|---|---|
| `object` | object of class meltt. |
| `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. |
| `confirmed_matches` | boolean vector specifying entries to be removed from deduplicated set. Function returns a `data.frame` 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt, meltt_data, meltt_duplicates

## Examples

```
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)
```

meltt_validate *Validation method to assess data integrated by meltt.*

## Description

Function to efficiently sample a subset of integrated data to generate performance statistics.

## Usage

```
meltt_validate(object, description.vars = NULL, sample_prop = .1, within_window = TRUE,
       spatial_window = NULL, temporal_window = NULL, reset = FALSE)
```

## Arguments

| | |
|---|---|
| object | object of class meltt. |
| description.vars | String vector referencing column names located in the input data. These are the variables that will be folded into the description being validated; if none are provided, taxonomy levels are used by default. |
| sample_prop | Argument establishes the proportion of of the total matched pairs that are sampled from. The size of this sample is then used to determine the size of the control group (i.e. all entries not flagged as matches, which are paired with other unique entries and matches. These entries should not be matches). For example, if sample_prop = .1 and this results in a sample of 20 matched pairs, then 20 control pairs will be extracted from the set, leading to a total sample of 40 observations to be reviewed. Input must exist within the range .01 and 1. |
| within_window | Use the same spatio-temporal window used in the initial data integration to calculate what counts as a "proximate event" for all entries in the control group. Default = TRUE. If set to FALSE, user must assign a new spatio-temporal window using spatial_window and temporal_window. |
| spatial_window | If within_window==FALSE, set new spatial window (in km). |
| temporal_window | If within_window==FALSE, set new temporal window (in days). |
| reset | If TRUE, the validation step will be reset and a new validation sample frame will be produced. Default = FALSE. |

## Details

meltt_validate offers an efficient method of assessing the performance of **meltt** for a specific integration, by randomly sampling from a proportion of pairs of matching events flagged by the algorithm as the same event, and then sampling a "control group" of equal size from events that were identified as unique (offering both unique-unique and unique-match pairs). The function compiles the samples and then generates a shiny app to ease assessment. Once all entries in the sample have been assessed, the function then returns accuracy statistics in terms of a confusion matrix. Performance is determined by the difference in the qualitative assessment in comparison to the **meltt** integration.

## Value

Function automatically overwrites input "meltt" object; if validation set has been completely reviewed, then the function prints the performance statistics.

## Author(s)

Karsten Donnay and Eric Dunford.

## References

Karsten Donnay, Eric T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt, meltt_inspect

## Examples

```
data(crashMD)
output <- meltt(crash_data1, crash_data2, crash_data3,
taxonomies = crash_taxonomies, twindow = 1, spatwindow = 3)

## Not run:
# app will activate to validate sample.
meltt_validate(output)

# for smaller sample, must reset to overwrite existing validation sample
meltt_validate(output, sample_prop=.1, reset = TRUE)

# override of the validation to get a sense of the report
output$validation$validation_set$coding = 1

meltt_validate(output)

## End(Not run)
```

---

# mplot *Wrapper to generate an interactive spatial plot of meltt output data via Leaflet.*

---

## Description

mplot provides an interactive javascript map to plot the spatial distribution of duplicate and unique entries in the integrated data.

## Usage

```
mplot(object, matching = FALSE, jitter=.0001)
```

## Arguments

| | |
|---|---|
| object | object of class meltt. |
| matching | if TRUE, plot only matching entries (i.e. duplicates and matches), else plot unique and matching entries. Default = FALSE. |
| jitter | Numeric value to randomly offset longitude and latitude of points for plotting. Useful when points overlap. Default is a small jitter of .0001 degrees. |

## Details

mplot generates a spatial map using javascript via the Leaflet package. 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. Plot renders in the users viewer pane (if using RStudio) or in browser. Images of the map can be saved using the export button.

## Value

Returns a javascript map, which renders in the user's viewer pane, of all unique event locations (or duplicate and matching entries if matching= argument is true). Each unique event are denoted as orange circles, matching entries as blue circles, and duplicate entries as green entries.

## Author(s)

Karsten Donnay and Eric Dunford.

## References

Karsten Donnay, Eric T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt, tplot

## Examples

```
data(crashMD)
output = meltt(crash_data1, crash_data2, crash_data3,
                taxonomies = crash_taxonomies, twindow = 1,spatwindow = 3)

mplot(output)
```

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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

`meltt`

## Examples

```
data(crashMD)
output = meltt(crash_data1,crash_data2,crash_data3,
               taxonomies = crash_taxonomies,twindow = 1,spatwindow = 3)
plot(output)
```

## print.meltt *Print function for objects of class meltt.*

### 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

### See Also

`meltt`

## summary.meltt *Summary function for objects of class meltt.*

### 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

`meltt`

---

`tplot` *Temporal distribution of meltt output data.*

---

## Description

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

## Usage

```
tplot(object, time_unit = "months", free_scale = TRUE)
```

## Arguments

| | |
|---|---|
| `object` | object of class `meltt`. |
| `time_unit` | character specifying the temporal bin: `"days"`, `"weeks"`, `"months"`, and `"years"`. `Default = "months"`. |
| `free_scale` | boolean if duplicates should be presented on a different scale from unique entries. A free scale makes it easier to assess the number of duplicate entries and from which input data they emerge, given that there can be relatively few at times. `Default = TRUE`. |

## 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 T. Dunford, Erin C. McGrath, David Backer, David E. Cunningham. (2018). "Integrating Conflict Event Data." *Journal of Conflict Resolution*.

## See Also

meltt, mplot

## Examples

```
data(crashMD)
output = meltt(crash_data1, crash_data2, crash_data3,
                      taxonomies = crash_taxonomies, twindow = 1, spatwindow = 3)

# Free scale
tplot(output, time_unit = "days")

# Relative scale
tplot(output, time_unit = "days", free_scale = FALSE)
```

# Index