

Package: events (via r-universe)

September 8, 2024

Type Package

Title Store and Manipulate Event Data

Version 0.6.1

Date 2021-12-28

Depends R (>= 2.10)

Description The events package manipulates, aggregates and otherwise messes with event data from 'KEDS' and 'TABARI' software and those with similar output. It also bundles several classic event data sets. Most functions are superseded by those in 'dplyr' and 'tidyr'.

Imports methods

License GPL

Encoding UTF-8

RoxygenNote 7.1.2

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Repository <https://conjugateprior.r-universe.dev>

RemoteUrl <https://github.com/conjugateprior/events>

RemoteRef cran

RemoteSha dce506912286bca31cf915e1dfe5cde0513813c9

Contents

actors	2
add_eventscale	3
as.data.frame.eventscale	4
balkans.weis	4
cameo.scale	5
casia.weis	5

codes	6
events	7
filter_actors	7
filter_codes	8
filter_eventdata	9
filter_time	10
gulf.cameo	11
levant.cameo	11
make_dyads	12
make_fun_from_list	13
make_scale	13
map_actors	14
map_codes	15
one_a_day	16
plot_dyad	17
read_eventdata	17
read_eventdata2	19
read_keds	20
read_keds2	21
scale_codes	23
scale_coverage	23
score	24
scrub_keds	25
sources	25
spotter	26
summary.eventdata	27
summary.eventscale	28
targets	28
turkey.cameo	29
weis.goldstein.scale	30
Index	31

actors *List actor codes*

Description

Lists actor codes

Usage

actors(edo)

Arguments

edo Event data

Details

Lists all the actor codes that occur in the event data in alphabetical order.

Value

Array of actor codes

Author(s)

Will Lowe

See Also

[sources](#), [targets](#), [codes](#)

Examples

```
data(levant.cameo)
acts <- actors(levant.cameo)
head(acts)
tail(acts)
```

add_eventscale	<i>Apply eventscale to event data</i>
----------------	---------------------------------------

Description

Applies an eventscale to event data

Usage

```
add_eventscale(edo, sc)
```

Arguments

edo	Event data
sc	scale

Details

Applies an eventscale to event data. This adds a new field in the event data with the same name as the eventscale. Add as many as you want to keep around.

Value

Event data with a scaling

Author(s)

Will Lowe

as.data.frame.eventscale

Coerce Event Scale to Data Frame

Description

Coerce Event Scale to Data Frame

Usage

```
## S3 method for class 'eventscale'  
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	an event scale
row.names	ignored
optional	ignored
...	ignored

This function converts a list with event codes as names and event scores as values into a data frame with column 'code' containing the event codes and column 'score' as the event's score

Value

a data.frame containing event codes and scores

balkans.weis

Balkans conflict events in WEIS encoding

Description

Event data on the conflict during the collapse of Yugoslavia. Events are coded according to an extended WEIS scheme by the KEDS Project. The event stream contains 72953 events occurring between 2 April 1989 and 31 July 2003 involving 325 actors.

Author(s)

KEDS Project

References

Parus Analytics: <https://www.parusanalytics.com/eventdata/data.dir/balk.html>

`cameo.scale`*CAMEO codes to conflict-cooperation scale*

Description

A mapping of CAMEO event codes to [-10,10] representing a scale of conflict and cooperation, developed by the KEDS project. Taken from the documentation of the KEDS_Count software.

Details

The version of CAMEO used here is 0.9B5 [07.03.2021].

Author(s)

KEDS Project

References

Parus Analytics: <https://eventdata.parusanalytics.com/data.dir/cameo.html>

`casia.weis`*Central Asia events with WEIS event coding*

Description

Event data on Central Asia. Events are coded according to the WEIS scheme by the KEDS Project. The event stream contains 8377 events occurring between 02/05/1989 and 31/07/1999 involving 152 sources and 157 targets.

Details

Original data comes from file CASIA.LEADS.6CODE (six character actor codes and coded from leads) with duplicates removed using the one_a_day filter.

Author(s)

KEDS Project

References

Parus Analytics: <https://www.parusanalytics.com/eventdata/data.dir/casia.html>

codes	<i>List event codes</i>
-------	-------------------------

Description

Lists event codes

Usage

```
codes(edo)
```

Arguments

edo Event data

Details

Lists all the event codes that appear in the event data

Value

Array of event codes

Author(s)

Will Lowe

See Also

[sources](#), [targets](#), [actors](#)

Examples

```
data(levant.cameo)
cod <- codes(levant.cameo)
head(codes)
tail(codes)
```

events	<i>Stores and manipulates event data</i>
--------	--

Description

Stores, manipulates, scales, aggregates and creates directed dyadic time series from event data generated by KEDS, TABARI, or any other extraction tool with similarly structured output.

Details

Events offers simple methods for aggregating and renaming actors and event codes, applying event scales, and constructing regular time series at a choice of temporal scales and measurement levels.

Author(s)

Will Lowe <conjugateprior@gmail.com>

filter_actors	<i>Discard all but relevant actors</i>
---------------	--

Description

Discards all but relevant actors

Usage

```
filter_actors(
  edo,
  fun = function(x) TRUE,
  which = c("both", "target", "source")
)
```

Arguments

edo	Event data
fun	Function that returns TRUE for actor codes that should not be discarded.
which	What actor roles should be filtered

Details

The which parameter specifies whether the filter should be applied only to targets, only to sources, or to all actors in the event data.

Value

Event data containing only actors that pass through fun

Author(s)

Will Lowe

See Also

[filter_codes](#), [filter_time](#)

filter_codes

Discard all but relevant event codes

Description

Discards all but relevant event codes

Usage

```
filter_codes(edo, fun = function(x) TRUE)
```

Arguments

edo	Event data
fun	Function that returns TRUE or event codes that should not be discarded

Details

Applies the filter function to each event code to see whether to keep the observation.

Value

Event data containing only events that pass through fun

Author(s)

Will Lowe

See Also

[filter_actors](#), [filter_time](#)

filter_eventdata	<i>Filter events data</i>
------------------	---------------------------

Description

Applies a generic field filter to event data

Usage

```
filter_eventdata(edo, fun, which)
```

Arguments

edo	Events data object
fun	Function that should be applied
which	Which fields should be filtered

Details

This function applies a filter function to event data. It is the workhorse function behind the `filter_` functions. You should use these in ordinary use.

Value

Event data

Author(s)

Will Lowe

Examples

```
data(levant.cameo)
sp <- spotter("PAL", "ISR")
ev_targ <- filter_eventdata(levant.cameo, sp, "target") # these actors as targets
head(ev_targ, 3)
ev_dyad <- filter_eventdata(levant.cameo, sp, c("source", "target")) # source and target
head(ev_dyad, 3)
```

filter_time	<i>Restrict events to a time period</i>
-------------	---

Description

Restricts events to a time period

Usage

```
filter_time(edo, start = min(edo$date), end = max(edo$date))
```

Arguments

edo	Event data
start	A date or something convertible to a Date using as.Date
end	A date or something convertible to a Date using as.Date

Details

Restricts events on or after start and before or on end.

Value

Event data restricted to a time period

Author(s)

Will Lowe

See Also

[filter_codes](#), [filter_actors](#)

Examples

```
data(levant.cameo)
ev_jan1980 <- filter_time(levant.cameo,
  start = as.Date("1980-01-01"),
  end = as.Date("1980-01-31"))
ev_feb1980 <- filter_time(levant.cameo,
  start = "1980-02-01", end = "1980-01-29")
ev_starttojan1980 <- filter_time(levant.cameo,
  end = "1980-01-29")
head(ev_starttojan1980)
```

gulf.cameo

Gulf States' events in CAMEO encoding

Description

Event data for th Gulf States. Events are coded according to the CAMEO scheme by the KEDS Project. The event stream contains 29029 events occurring between 03/01/1992 and 07/31/2006 involving 411 sources and 397 targets.

Author(s)

KEDS Project

References

Parus Analytics: <https://www.parusanalytics.com/eventdata/data.dir/gulf.html>

levant.cameo

Levant events with CAMEO event coding

Description

Event data on Middle East. Events are coded according to the CAMEO scheme by the KEDS Project. The event stream contains 145709 events occurring between 15/04/1979 and 30/11/2011 involving 741 sources and 688 targets.

Details

Original data comes from file REULE.201111.evt with documentation lines (marked as DOC DOC 999), match information, and duplicates removed.

Author(s)

KEDS Project

References

Parus Analytics: <https://www.parusanalytics.com/eventdata/data.dir/levant.html>

 make_dyads

Aggregate events to a regular time interval

Description

Aggregates events to a regular time interval

Usage

```
make_dyads(
  edo,
  scale = NULL,
  unit = c("week", "day", "month", "quarter", "year"),
  monday = TRUE,
  fun = mean,
  missing.data = NA
)
```

Arguments

edo	Event data
scale	Name of an eventscale or NULL to create counts
unit	Temporal aggregation unit
monday	Whether weeks start on Monday. If FALSE, they start on Sunday
fun	Aggregation function. Should take a vector and return a scalar
missing.data	What weeks with no data are assigned

Details

In an event data set S , assume that $A = \text{length}(\text{actors}(S))$ actors $K = \text{length}(\text{codes}(S))$ event codes occur. This function creates A^2 data streams labelled by the combination of source and target actors. If `scale` is NULL these are K -dimensional time series of event counts. If `scale` names a scale that has been added to the event data `fun` is used to aggregate the events falling into each temporal interval. This creates a univariate interval valued time series for each directed dyad.

See the vignette for more detail and a worked example.

Value

A list of named dyadic aggregated time series

Author(s)

Will Lowe

make_fun_from_list	<i>Create a mapping function from list</i>
--------------------	--

Description

Creates a mapping function from list

Usage

```
make_fun_from_list(lst)
```

Arguments

lst	A list
-----	--------

Details

Turns a list of the form `list(a=c(1,2), b=3)` into a function that returns 'a' when given 1 or 2 as argument, 'b' when given 3 and otherwise gives back its argument unchanged.

This is a convenience function to make it possible to specify onto mappings using lists. The `map_*` functions use it internally, but you might find a use for it.

Value

A function that inverts the mapping specified by `lst`

Author(s)

Will Lowe

make_scale	<i>Make an event scale</i>
------------	----------------------------

Description

Makes an event scale

Usage

```
make_scale(  
  name,  
  types = NULL,  
  values = NULL,  
  file = NULL,  
  desc = "",  
  default = NA,  
  sep = ", "  
)
```

Arguments

name	Name of scale
types	Array of event codes
values	Array of event code values
file	Input file defining event codes and their values
desc	Optional description of the scale
default	What to assign event codes that have no mapping in the scale. Defaults to NA.
sep	Separator in file

Details

Makes an event scale from a specification found in a file or using the `types` and `variables` parameters. If a file is specified it is assumed to be headerless and to contain event codes in the first column and numerical values in the second column.

Scales must be assigned a name and may also be assigned a description. If you wish to assign codes without a specified value to some particular value, set `default` to something other than NA.

Value

An event scale object

Author(s)

Will Lowe

map_actors	<i>Aggregate actor codes</i>
------------	------------------------------

Description

Aggregates actor codes

Usage

```
map_actors(edo, fun = function(x) x)
```

Arguments

edo	Event data
fun	Function or list specifying the aggregation mapping

Details

The function relabels actor codes according to the filter. The filter may either be a function that returns the new name of an event when handed the old one, or a list structured like `list(fruit=c('tomato', 'orange'), veg=c('red pepper', 'carrot'))`.

This function can also be used as a renaming function, but it is most useful when multiple codes should be treated as equivalent.

For a detailed example of event code and actor aggregation functions, see the 'Actor Filtering' and 'Count Aggregation' section of the vignette.

Value

Event data with new actor codes

Author(s)

Will Lowe

See Also

[map_codes](#)

map_codes

Aggregate event codes

Description

Aggregates event codes

Usage

```
map_codes(edo, fun = function(x) x)
```

Arguments

edo	Event data
fun	Function or list specifying the aggregation mapping

Details

This function relabels event codes according to fun, which may either be a function that returns the new name of an event when handed the old one, or a list with entries of the form: `list[[newname]] = c(olddname1, olddname2)`.

It can also be used as a renaming function, but it is most useful when multiple codes should be treated as equivalent.

For a detailed example of event code and actor aggregation functions, see the 'Actor Filtering' and 'Count Aggregation' section of the vignette.

Value

Event data with new event codes

Author(s)

Will Lowe

See Also

[map_actors](#)

one_a_day

Apply the one-a-day filter

Description

Tries to remove duplicate events

Usage

```
one_a_day(edo)
```

Arguments

edo Event data object

Details

This function removes duplicates of any event that occurs to the same source and target on the same date with the same event code, on the assumption that these are in fact the same event reported twice.

This function can also be applied as part of [read_keds](#)

Value

New event data object with duplicate events removed

Author(s)

Will Lowe

See Also

[read_keds](#)

plot_dyad	<i>Plot scaled directed dyad</i>
-----------	----------------------------------

Description

Plots scaled directed dyad

Usage

```
plot_dyad(dyad, ...)
```

Arguments

dyad	One directed dyadic time series from the make_dyads function
...	Extra arguments to plot

Details

A convenience function to plot the named scale within a directed dyad against time.

Value

Nothing, used for side effect

Author(s)

Will Lowe

read_eventdata	<i>Read event data files</i>
----------------	------------------------------

Description

Reads event data output files in free format

Usage

```
read_eventdata(  
  d,  
  col.format = "D.STC",  
  one.a.day = TRUE,  
  scrub.keds = TRUE,  
  date.format = "%y%m%d",  
  sep = "\t",  
  head = FALSE  
)
```

Arguments

d	Names of event data files
col.format	Format for columns in d (see details)
one.a.day	Whether to apply the duplicate event remover
scrub.keds	Whether to apply the data cleaner
date.format	How dates are represented in the original file
sep	File separator
head	Whether there is a header row in d

Details

Reads event data output and optionally applies the [scrub_keds](#) cleaning function and the [one_a_day](#) duplicate removal filter.

This function assumes that d is a vector of output files. These are assumed to be sep-separated text files. The column ordering is given by the col.format parameter:

- D the date field
- S the source actor field
- T the target actor field
- C the event code field
- L the event code label field (optional)
- Q the quote field (optional)
- . (or anything not shown above) an ignorable column

e.g. the default "D.STC" format means that column 1 is the date, column 2 should be ignored, column 3 is the source, column 4 is the target, and column 5 is the event code. In this specification no quote or label columns are extracted.

The specification need only use the period to generate correct spacing, e.g. if there are 10 fields in each line, the first five of which are: data, something ignorable, source, target, event code, and the remaining five fields are ignorable then ""D.STC" is sufficient to extract date, source, target, and event code

The code plucks out just these columns, formats them appropriately and ignores everything else in the file. Only D, S, T, and C are required.

The format of the date field is given by format.date

Value

An event data set

Author(s)

Will Lowe

Examples

```
# the first 1000 lines of raw TABARI output for Levant data,
# (see data set "levant.cameo" for complete unlabeled data set)
lev1000 <- system.file("extdata", "levant.cameo.top1000.txt",
  package = "events")
evs1000 <- read_eventdata(lev1000, col.format = "DSTCL")
head(evs1000, 3)
```

read_eventdata2	<i>Read event data files</i>
-----------------	------------------------------

Description

Reads event data output files more robustly than `read_eventdata`

Usage

```
read_eventdata2(
  d,
  col.format = "D.STC",
  one.a.day = TRUE,
  scrub.keds = TRUE,
  date.format = "%y%m%d",
  sep = "\t",
  head = FALSE,
  verbose = TRUE
)
```

Arguments

<code>d</code>	Names of event data files
<code>col.format</code>	Format for columns in <code>d</code> (see details)
<code>one.a.day</code>	Whether to apply the duplicate event remover
<code>scrub.keds</code>	Whether to apply the data cleaner
<code>date.format</code>	How dates are represented in the original file
<code>sep</code>	File separator
<code>head</code>	Whether there is a header row in <code>d</code>
<code>verbose</code>	Whether to update read progress and report unreadable lines

Details

Reads event data output and optionally applies the `scrub_keds` cleaning function and the `one_a_day` duplicate removal filter. This function is slower but more robust to line noise than `read_eventdata`. Use this when that one fails.

This function assumes that `d` is a vector of output files. These are assumed to be sep-separated text files. The column ordering is given by the `col.format` parameter:

- D the date field
- S the source actor field
- T the target actor field
- C the event code field
- L the event code label field (optional)
- Q the quote field (optional)
- . (or anything not shown above) an ignorable column

e.g. the default "D.STC" format means that column 1 is the date, column 2 should be ignored, column 3 is the source, column 4 is the target, and column 5 is the event code. The optional quote and label column are not searched for.

The code plucks out just these columns, formats them appropriately and ignores everything else in the file. Only D, S, T, and C are required.

The format of the date field is given by `format.date`

Value

An event data set

Author(s)

Will Lowe

Examples

```
# the first 1000 lines of raw TABARI output for Levant data,
# (see data set "levant.cameo" for complete unlabeled data set)
lev1000 <- system.file("extdata", "levant.cameo.top1000.txt",
  package = "events")
evs1000 <- read_eventdata2(lev1000, col.format = "DSTCL")
head(evs1000, 3)
```

read_keds

Read KEDS (or TABARI) events files

Description

Reads KEDS event data output files

Usage

```
read_keds(
  d,
  keep.quote = FALSE,
  keep.label = TRUE,
  one.a.day = TRUE,
  scrub.keds = TRUE,
  date.format = "%y%m%d"
)
```

Arguments

d	Names of files of KEDS/TABARI output
keep.quote	Whether the exact noun phrase be retained
keep.label	Whether the label for the event code should be retained
one.a.day	Whether to apply the duplicate event remover
scrub.keds	Whether to apply the data cleaner
date.format	How dates are represented in the first column

Details

Reads KEDS output and optionally applies the [scrub_keds](#) cleaning function and the [one_a_day](#) duplicate removal filter. This function is thin wrapper around `read_eventdata` which is a thin wrapper around `read.csv`.

For noisy datasets [read_keds2](#) is slower but more robust. Use that if this one fails.

This function assumes that `d` are a vector of KEDS/TABARI output files. These are assumed to be tab separated text files wherein the first field is a date in `yymmdd` format or as specified by `date.format`, the second and third fields are actor codes, the fourth field is an event code, and the fifth field is a text label for the event type, and the sixth field is a quote - some kind of text from which the event code was inferred. Label and quote are optional and can be discarded when reading in.

Value

An event data set

Author(s)

Will Lowe

read_keds2	<i>Read KEDS (or TABARI) events files</i>
------------	---

Description

Reads KEDS/TABARI event data output files more robustly than `read_keds`

Usage

```
read_keds2(
  d,
  keep.quote = FALSE,
  keep.label = TRUE,
  one.a.day = TRUE,
  scrub.keds = TRUE,
  date.format = "%y%m%d",
  verbose = TRUE
)
```

Arguments

<code>d</code>	Names of files of KEDS/TABARI output
<code>keep.quote</code>	Whether the exact noun phrase be retained
<code>keep.label</code>	Whether the label for the event code should be retained
<code>one.a.day</code>	Whether to apply the duplicate event remover
<code>scrub.keds</code>	Whether to apply the data cleaner
<code>date.format</code>	How dates are represented in the date column
<code>verbose</code>	Whether to show progress and report unreadable lines

Details

Reads KEDS/TABARI output and optionally applies the [scrub_keds](#) cleaning function and the [one_a_day](#) duplicate removal filter. This function is slower but more robust to line noise than [read_keds](#). Use this when that one fails.

This function assumes that `d` are a vector of KEDS/TABARI output files. These are assumed to be tab separated text files wherein the first field is a date in `yymmdd` format or as specified by `date.format`, the second and third fields are actor codes, the fourth field is an event code, and the fifth field is a text label for the event type, and the sixth field is a quote - some kind of text from which the event code was inferred. Label and quote are optional and can be discarded when reading in.

Value

An event data set

Author(s)

Will Lowe

Examples

```
# the first 1000 lines of raw TABARI output for Levant data,
# (see data set "levant.cameo" for complete unlabeled data set)
lev1000 <- system.file("extdata", "levant.cameo.top1000.txt",
  package = "events")
evs1000 <- read_keds2(lev1000)
head(evs1000, 3)
```

scale_codes	<i>Show which events are scaleable</i>
-------------	--

Description

Shows which events codes are covered by a scale

Usage

```
scale_codes(es)
```

Arguments

es	Eventscale
----	------------

Details

Returns an array of event codes to which an eventscale assigns a value.

Value

Array of scaleable event codes

Author(s)

Will Lowe

scale_coverage	<i>Check coverage of scale for event data</i>
----------------	---

Description

Checks coverage of scale for event data

Usage

```
scale_coverage(sc, edo)
```

Arguments

sc	An eventscale
edo	Event data

Details

Returns an array of event codes that occur in an event data set but are not assigned values by the scale. These are the codes that will, in subsequent processing, be assigned the scale's default value.

Value

Array of unscalable event codes

Author(s)

Will Lowe

score

Score event codes with an event scale

Description

Gets scale scores for event codes

Usage

```
score(eventscale, codes)
```

Arguments

eventscale	An event scale
codes	Event codes

Details

Returns an array of scores corresponding to the the second argument's scale values or the scale's default value if not recognized.

You should use this function to avoid relying on the internal structure of event scales. They are currently lists, but this may change.

Value

Numerical values for each event codes from the scale

Author(s)

Will Lowe

scrub_keds	<i>Remove well-known noise from KEDS event data file</i>
------------	--

Description

Removes well-known noise from KEDS output files

Usage

```
scrub_keds(edo)
```

Arguments

edo	An event data object
-----	----------------------

Details

This function applies the regular expression based cleaning routine from the KEDS website. This is a direct translation from the original PERL which replaces capital 'O's and small 'l's with 0 and 1 respectively and removes the event code '—]', on the assumption that these are all output noise.

Value

Event data

Author(s)

Will Lowe

See Also

[read_keds](#)

sources	<i>List source actor codes</i>
---------	--------------------------------

Description

Lists source actor codes

Usage

```
sources(edo)
```

Arguments

edo	Event data
-----	------------

Details

Lists all the actor codes that appear as a source in the event data in alphabetical order.

Value

Array of actor codes

Author(s)

Will Lowe

See Also

[actors](#), [targets](#), [codes](#)

Examples

```
data(levant.cameo)
src <- sources(levant.cameo)
head(src)
tail(src)
```

spotter

Make a spotting function

Description

Hands back a function to spot the items it was given in (. . .)

Usage

```
spotter(...)
```

Arguments

... The actor names for which the new function should return TRUE

Details

This is a convenience function for creates a function that returns true for exact matches to its arguments.

Value

A function

Author(s)

Will Lowe

Examples

```
data("balkans.weis")
head(balkans.weis, 3)
sp <- spotter("SER", "SERMIL")
events <- filter_actors(balkans.weis, sp)
head(events, 3)
```

summary.eventdata	<i>Summarise event data</i>
-------------------	-----------------------------

Description

Summarises a set of event data

Usage

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

Arguments

object	Event data object
...	Not used

Details

This is a compact summary of an event data object. For more detail consult the object itself. Currently it is simply a data.frame with conventionally named column names, but that almost certainly will change to deal with larger datasets in later package versions. If your code uses the package's accessor functions then you won't feel a thing when this happens.

Value

A short description of the event data

Author(s)

Will Lowe

summary.eventscale *Summarise an eventscale*

Description

Summarise an eventscale

Usage

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

Arguments

object	Scale
...	Not used

Details

Print summary statistics for an eventscale.

Value

Nothing, used for side effect

Author(s)

Will Lowe

targets *Lists target actor codes*

Description

Lists target actor codes

Usage

```
targets(edo)
```

Arguments

edo	Event data
-----	------------

Details

Lists all the actor codes that appear as a target in the event data in alphabetical order.

Value

Array of actor codes

Author(s)

Will Lowe

See Also

[sources](#), [actors](#), [codes](#)

Examples

```
data(levant.cameo)
targs <- sources(levant.cameo)
head(targs)
tail(targs)
```

turkey.cameo

Turkey events in CAMEO encoding

Description

Event data for Turkey. Events are coded according to the CAMEO scheme by the KEDS Project. The event stream contains 54466 events involving 164 sources and 166 targets between 15/04/1979 and 31/03/1999.

Details

Note: This is the data set with only leads coded (GULF99.zip)

Author(s)

KEDS Project

References

Parus Analytics: <https://www.parusanalytics.com/eventdata/data.dir/turkey.html>

weis.goldstein.scale *WEIS codes to Goldstein conflict-cooperation scale*

Description

A mapping of WEIS event codes to [-10,10] representing a scale of conflict and cooperation, developed by Joshua Goldstein and slightly extended for the KEDS project. Note: This mapping does not cover all the event codes in balkans.weis. Taken from the KEDS Project's documentation.

Author(s)

KEDS Project

References

Parus Analytics: <https://www.parusanalytics.com/eventdata/data.dir/weis.html>

Index

* data

- balkans.weis, 4
 - cameo.scale, 5
 - casia.weis, 5
 - gulf.cameo, 11
 - levant.cameo, 11
 - turkey.cameo, 29
 - weis.goldstein.scale, 30
- actors, 2, 6, 26, 29
- add_eventscale, 3
- as.data.frame.eventscale, 4
- balkans.weis, 4
- cameo.scale, 5
- casia.weis, 5
- codes, 3, 6, 26, 29
- events, 7
- filter_actors, 7, 8, 10
- filter_codes, 8, 8, 10
- filter_eventdata, 9
- filter_time, 8, 10
- gulf.cameo, 11
- levant.cameo, 11
- make_dyads, 12
- make_fun_from_list, 13
- make_scale, 13
- map_actors, 14, 16
- map_codes, 15, 15
- one_a_day, 16, 18, 19, 21, 22
- package-events (events), 7
- plot_dyad, 17
- read_eventdata, 17, 19
- read_eventdata2, 19
- read_keds, 16, 20, 22, 25
- read_keds2, 21, 21
- scale_codes, 23
- scale_coverage, 23
- score, 24
- scrub_keds, 18, 19, 21, 22, 25
- sources, 3, 6, 25, 29
- spotter, 26
- summary.eventdata, 27
- summary.eventscale, 28
- targets, 3, 6, 26, 28
- turkey.cameo, 29
- weis.goldstein.scale, 30