Package: eventreport (via r-universe)

June 9, 2026

Type Package

Title Diagnose, Visualize, and Aggregate Event Report Level Data

Version 0.1.2

Maintainer Sebastian van Baalen <sebastian.van-baalen@pcr.uu.se>

Description Diagnose, visualize, and aggregate event report level data to the event level. Users provide an event report level dataset, specify their aggregation rules, and the package produces a dataset aggregated at the event level. Also includes the Modes and Agents of Election-Related Violence in Côte d'Ivoire and Kenya (MAVERICK) dataset, an event report level dataset that records all documented instances of electoral violence from the first multiparty election to 2022 in Côte d'Ivoire (1995-2022) and Kenya (1992-2022). For more details see van Baalen and Höglund (2026) <doi:10.1093/isq/sqag014>. Users of the enclosed MAVERICK dataset should also cite van Baalen and Höglund (2026) <doi:10.1093/jopres/xjaf012>.

License CC BY 4.0

Encoding UTF-8

LazyData true

Imports dplyr, lubridate, purrr, tidyr, ggplot2, magrittr, rlang, scales, tidyselect, tibble

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0), knitr, rmarkdown, devtools, tidyverse, tinytable

Depends R (>= 3.5.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/sebastianvanbaalen/eventreport

BugReports https://github.com/sebastianvanbaalen/eventreport/issues

Config/pak/sysreqs libicu-dev

Repository https://sebastianvanbaalen.r-universe.dev

Date/Publication 2026-03-11 06:25:24 UTC

RemoteUrl https://github.com/sebastianvanbaalen/eventreport

RemoteRef HEAD

RemoteSha 3527543e6b24fc380de023dd23f65c8b172bb9f9

Contents

`aggregate_maverick_con` *Load the most-conservative aggregation of MAVERICK*

Description

This convenience function aggregates the MAVERICK event report data to the event level using the most-conservative aggregation model.

Usage

```
aggregate_maverick_con(data)
```

Arguments

data	The MAVERICK event report level dataset. Already pre-loaded.

Value

Returns a dataframe of the most-conservative aggregation of the MAVERICK dataset.

Examples

```
maverick_conservative <- aggregate_maverick_con()
```

---

aggregate_maverick_inf

Load the most-informative aggregation of MAVERICK

Description

This convenience function aggregates the MAVERICK event report data to the event level using the most-informative aggregation model.

Usage

```
aggregate_maverick_inf(data)
```

Arguments

data	The MAVERICK event report level dataset. Already pre-loaded.

Value

Returns a dataframe of the most-informative aggregation of the MAVERICK dataset.

Examples

```
maverick_informative <- aggregate_maverick_inf()
```

aggregate_maverick_rep *Load the most-representative aggregation of MAVERICK*

Description

This convenience function aggregates the MAVERICK event report data to the event level using the most-representative aggregation model.

Usage

```
aggregate_maverick_rep(data)
```

Arguments

data	The MAVERICK event report level dataset. Already pre-loaded.

Value

Returns a dataframe of the most-representative aggregation of the MAVERICK dataset

Examples

```
maverick_representative <- aggregate_maverick_rep()
```

aggregate_strings *Combine strings from a character variable*

Description

This function combines strings from a character variable.

Usage

```
aggregate_strings(str_var)
```

Arguments

str_var	A character vector.

Value

Returns a single character string with unique strings concatenated by semicolons.

Examples

```
aggregate_strings(c("apple", "banana", "apple", "Unknown", "orange", " "))
```

aggregateData *Aggregate event report data*

Description

This function aggregates event report data based on a specified grouping variable and various aggregation criteria.

Usage

```
aggregateData(
  data,
  group_var = "event_id",
  find_mode = NULL,
  find_mode_na_ignore = NULL,
  find_mode_bin = NULL,
  find_mode_date = NULL,
  find_mode_numeric = NULL,
  find_least_precise = NULL,
  find_most_precise = NULL,
  combine_strings = NULL,
  find_max = NULL,
  find_min = NULL,
  summarize_vars = NULL,
  aggregation_name = NULL,
  tie_break = "default_tie_break",
  second_tie_break = "default_tie_break"
)
```

Arguments

data	A data frame containing the data to be aggregated.
group_var	A string specifying the variable to group by. Default is "event_id".
find_mode	A vector of variable names for which to find the mode.
find_mode_na_ignore	A vector of variable names for which to find the mode, ignoring NAs.
find_mode_bin	A vector of variable names for which to find the binary mode.
find_mode_date	A vector of variable names for which to find the mode for dates.
find_mode_numeric	A vector of variable names for which to find the mode for numeric values.
find_least_precise	A list of lists, each containing a variable name and its corresponding precision variable, to find the least precise value.
find_most_precise	A list of lists, each containing a variable name and its corresponding precision variable, to find the most precise value.

`combine_strings`
: A vector of variable names for which to combine strings.

`find_max`
: A vector of variable names for which to find the maximum value.

`find_min`
: A vector of variable names for which to find the minimum value.

`summarize_vars`
: A vector of variable names for which to sum all values.

`aggregation_name`
: A string specifying the name of the aggregation.

`tie_break`
: A string specifying the tie break column name. Default is "default_tie_break".

`second_tie_break`
: A string specifying the second tie break column name. Default is "default_tie_break".

Value

A data frame with the aggregated results.

Examples

```
small_maverick_event_report %>%
  aggregateData(group_var = "event_id", find_mode = "city") %>%
  utils::head(10)
```

---

aggregation_diagnostics

Compute multiple aggregation diagnostics for a set of variables

---

Description

This convenience function runs all six diagnostic functions in the package, mean divergence, normalized divergence, mean standard deviation, mean range, share of events with disagreement, and modal confidence, and returns a combined tibble with one row per variable.

Usage

```
aggregation_diagnostics(data, group_var, variables)
```

Arguments

`data`
: A data frame containing event report level data.

`group_var`
: A character string naming the column that uniquely identifies events (e.g., "event_id").

`variables`
: A character vector of column names to include in the diagnostics.

Details

The function handles mixed-type input: each diagnostic is only run on the subset of variables for which it is valid. Variables that do not apply to a particular diagnostic will have ‘NA‘ in that column.

Value

A tibble with one row per variable and columns:

variable The name of each variable.

dscore Mean divergence score.

dscore_normalized Normalized divergence score.

mean_sd Mean within-event standard deviation (numeric variables only).

mean_range Mean within-event range (numeric variables only).

share_disagreement Share of events with any disagreement.

modal_confidence Average modal confidence per variable.

#' @importFrom dplyr full_join

Examples

```
small_maverick_event_report %>%
  aggregation_diagnostics(
    group_var = "event_id",
    variables = c("city", "deaths_best", "actor1")
   )
```

`calc_max_precision` *Calculate the mode value at maximum precision*

Description

This function determines the mode of a variable 'x', filtered to entries with the maximum value of a specified precision vector 'precision_var'. It optionally resolves ties using one or two additional vectors for tie-breaking.

Usage

```
calc_max_precision(x, precision_var, tie_break = NULL, second_tie_break = NULL)
```

Arguments

`x`	A vector of values for which to find the mode.
`precision_var`	A vector of precision values corresponding to 'x', used to filter to maximum values.
`tie_break`	Optional; a vector used as the first tie-break criterion.
`second_tie_break`	Optional; a vector used as the second tie-break criterion.

Value

Returns the mode of ‘x‘ for entries with maximum ‘precision_var‘ value. If no valid entries exist, returns an empty string.

Examples

```
x = c("apple", "apple", "banana", "banana")
precision_var = c(1, 2, 1, 2)
tie_break = c(1, 2, 1, 2)
second_tie_break = c(1, 1, 2, 1)
calc_max_precision(x, precision_var, tie_break, second_tie_break)
```

---

calc_min_precision *Calculate the mode value at minimum precision*

Description

This function determines the mode of a variable ‘x‘, filtered to entries with the minimum value of a specified precision vector ‘precision_var‘. It optionally resolves ties using one or two additional vectors for tie-breaking.

Usage

```
calc_min_precision(x, precision_var, tie_break = NULL, second_tie_break = NULL)
```

Arguments

x	A vector of values for which to find the mode.
precision_var	A vector of precision values corresponding to ‘x‘, used to filter to minimum values.
tie_break	Optional; a vector used as the first tie-break criterion.
second_tie_break	Optional; a vector used as the second tie-break criterion.

Value

Returns the mode of ‘x‘ for entries with minimum ‘precision_var‘ value. If no valid entries exist, returns an empty string.

Examples

```
x = c("apple", "apple", "banana", "banana")
precision_var = c(1, 2, 1, 2)
tie_break = c(1, 2, 1, 2)
second_tie_break = c(1, 1, 2, 1)
calc_min_precision(x, precision_var, tie_break, second_tie_break)
```

`calc_mode` *Calculate mode with optional tie-breaks*

Description

This function calculates the mode of a given vector and optionally resolves ties using one or two levels of tie-breaks.

Usage

```
calc_mode(x, tie_break = NULL, second_tie_break = NULL)
```

Arguments

`x`	A character vector for which to find the mode.
`tie_break`	An optional numeric vector used as the first tie-break criterion.
`second_tie_break`	An optional numeric vector used as the second tie-break criterion when the first is insufficient.

Value

Returns the mode of ‘x‘. If there are multiple modes and no tie-breaks are specified or they do not resolve the ties, returns "Indeterminate".

Examples

```
data <- c("apple", "apple", "banana", "banana")
tie_break <- c(1, 2, 1, 2)
second_tie_break <- c(1, 1, 2, 1)
calc_mode(data)  # Expect: "Indeterminate"
calc_mode(data, tie_break)  # Expect: "Indeterminate"
calc_mode(data, tie_break, second_tie_break)  # Expect: "banana"
```

`calc_mode_binary` *Calculate mode of a binary numeric vector*

Description

Calculate mode of a binary numeric vector

Usage

```
calc_mode_binary(x)
```

Arguments

x	A numeric vector consisting only of binary values (0 and 1).

Value

Returns a numeric vector representing the mode value. Returns 1 if there is a tie. Returns 'NA' if the vector is empty.

Examples

```
calc_mode_binary(c(0, 1, 1, 0, 1))
```

---

calc_mode_date *Calculate mode of date vector*

Description

Calculate mode of date vector

Usage

```
calc_mode_date(x)
```

Arguments

x	A character vector where each element is a date in "YYYY-MM-DD" format.

Value

Returns a date vector representing the modal date, or the mean of the modal dates if there is a tie.

Examples

```
calc_mode_date(c("2021-01-01", "2021-01-02", "2021-01-01"))
```

calc_mode_na_ignore *Calculate mode with optional tie-breaks ignoring NA and empty strings*

Description

This function calculates the mode of a given vector, ignoring 'NA' and empty strings, and optionally resolves ties using one or two levels of tie-breaks. If all values are 'NA' or empty, the function returns 'NA'.

Usage

```
calc_mode_na_ignore(x, tie_break = NULL, second_tie_break = NULL)
```

Arguments

x	A character vector for which to find the mode.
tie_break	An optional numeric vector used as the first tie-break criterion.
second_tie_break	An optional numeric vector used as the second tie-break criterion when the first is insufficient.

Value

Returns the mode of 'x' ignoring 'NA' and empty strings. If the filtered vector is empty or all elements are 'NA' or empty, returns 'NA'.

Examples

```
data <- c("apple", "", "banana", NA)
tie_break <- c(1, NA, 1, NA)
second_tie_break <- c(1, NA, 2, NA)
calc_mode_na_ignore(data)  # Expect: "apple"
calc_mode_na_ignore(data, tie_break)  # Expect: "banana"
calc_mode_na_ignore(data, tie_break, second_tie_break)  # Expect: "banana"
```

calc_mode_numeric *Calculate mode of numeric vector*

Description

This function calculates the mode of a given numeric vector, and returns the smallest mode value if multiple modes exist.

Usage

```
calc_mode_numeric(x)
```

Arguments

x	A numeric vector.

Value

Returns a numeric vector representing the mode value. Returns the smallest mode value if multiple modes exist, and NA if the vector is empty or contains non-numeric elements.

Examples

```
calc_mode_numeric(c(1, 2, 2, 3, 4, 4))
```

---

dscore *Calculate discrepancy score*

Description

This function computes the mean number of unique values minus one for each specified variable within each group specified by the group_var. It is designed to provide insights into the variability of each variable while adjusting for the minimum possible unique count.

Usage

```
dscore(data, group_var, variables)
```

Arguments

data	A dataframe containing the data to be analyzed.
group_var	A character string specifying the column name used for grouping the data.
variables	A character vector of column names in ‘data‘ for which the mean number of unique values minus one is calculated.

Value

A tibble with each specified variable showing the mean of (unique values - 1) for each group. The data is grouped by the ‘group_var‘ and returns the results in a wide format, where each variable is prefixed with "dscore_" to indicate the calculation.

Examples

```
df <- data.frame(
  group = c("A", "A", "B", "B", "B"),
  age = c(25, 25, 30, 35, 30),
  gender = c("Male", "Male", "Female", "Female", "Female"),
  income = c(50000, 50000, 60000, 65000, 60000)
)
result <- dscore(df, "group", c("age", "gender", "income"))
print(result)
```

`event_level_disagreement` *Calculate event-level disagreement scores by variable (wide format)*

Description

This function calculates the level of disagreement across event reports for each event and variable. For a given event and variable, it computes 1 minus the proportion of reports that agree with the modal value. A score of 0 indicates full agreement, while higher scores indicate greater disagreement.

Usage

```
event_level_disagreement(data, group_var, variables)
```

Arguments

`data`	A data frame containing event report level data.
`group_var`	A character string naming the column that uniquely identifies events (e.g., "event_id").
`variables`	A character vector of column names to check for disagreement.

Details

The result is a wide-format tibble with one row per event and one column per variable.

Value

A wide-format tibble where each row is an event and each column is a disagreement score for a variable.

Examples

```
df <- data.frame(
  event_id = c(1, 1, 2, 2, 3),
  actor1 = c("Actor A", "Actor B", "Actor B", "Actor B", "Actor C"),
  deaths_best = c(10, 10, 5, 15, 10)
)
event_level_disagreement(
```

```
  df,
  group_var = "event_id",
  variables = c("actor1", "deaths_best")
)
```

`maverick_event_report` *The MAVERICK event report level dataset*

Description

The Modes and Agents of Election-Related Violence in Côte d'Ivoire and Kenya (MAVERICK) is an event report level dataset of electoral violence incidents.

Usage

```
maverick_event_report
```

Format

A data frame with 3287 rows and 108 columns.

id A unique event report identifier.

event_id A unique event identifier assigned by the coders. Needed to aggregate event reports into events.

country A character class variable that contains the name of the country in which the event took place.

election A character class variable that contains the name of the election to which the event was most closely associated.

certain A numeric class variable that denotes the number of inclusion criteria that the event report fulfilled.

certain1 A integer class variable that denotes whether the reported event was inferred to be election-related because the event report or another event report explicitly identified the event as election-related.

certain2 A integer class variable that denotes whether the reported event was inferred to be election-related because at least one of the actors involved had explicit ties to a political party or was referred to by their party affiliation.

certain3 A integer class variable that denotes whether the reported event was inferred to be election-related because at least one of the targets was election-related, such as voters at a polling station, political candidates, election observers, security forces deployed to overlook the election, electoral material, or electoral infrastructure.

certain4 A integer class variable that denotes whether the reported event was inferred to be election-related because the reported purpose of the event was to influence an electoral process or outcome.

certain5 A integer class variable that denotes whether the reported event was inferred to be election-related because the event was part of an episode of electoral violence or occurred as a reaction to an earlier electoral violence event.

certain6 A integer class variable that denotes whether the the reported event was inferred to be election-related because it occurred at most 6 months prior to or after an election.

date_start A character class variable that contains the earliest possible event date expressed in YYYY-MM-DD format.

date_end A character class variable that contains the latest possible event date expressed in YYYY-MM-DD format.

city A character class variable that contains the name of the city or village in which the event took place.

location A character class variable that contains a text description of the most precise event location described in the report.

latitude A numeric class variable that contains the latitude for the location indicated in *location*.

longitude A numeric class variable that contains the longitude for the location indicated in *location*.

geo_precision A numeric class variable that denotes how precisely the geo-coordinates are coded, ranging from the country level (1) to the exact street or building (6).

actor1 A character class variable that contains the name of the actor involved in the event.

actor1_id A unique actor identifier assigned by the coders.

actor1_type A character class variable that records the type of actor.

actor1_subtype A character class variable that records the subtype of actor.

actor1_party A character class variable that records the party affiliation of actor.

actor1_violence A character class variable that records all forms of violence used by the actor.

actor1_precision A numeric class variable that denotes how precisely the actor information is coded.

actor1_initiator An integer class variable that denotes whether the actor was the initiator of the violence.

actor1_perpetrator An integer class variable that denotes whether the actor was a perpetrator of the violence.

actor1_intervener An integer class variable that denotes whether the actor was an intervener in the violence.

actor1_bystander An integer class variable that denotes whether the actor was a passive bystander to the violence.

actor1_victim An integer class variable that denotes whether the actor was also a victim of the violence.

actor2 A character class variable that contains the name of the actor involved in the event.

actor2_id A unique actor identifier assigned by the coders.

actor2_type A character class variable that records the type of actor.

actor2_subtype A character class variable that records the subtype of actor.

actor2_party A character class variable that records the party affiliation of actor.

actor2_violence A character class variable that records all forms of violence used by the actor.

actor2_precision A numeric class variable that denotes how precisely the actor information is coded.

actor2_initiator An integer class variable that denotes whether the actor was the initiator of the violence.

actor2_perpetrator An integer class variable that denotes whether the actor was a perpetrator of the violence.

actor2_intervener An integer class variable that denotes whether the actor was an intervener in the violence.

actor2_bystander An integer class variable that denotes whether the actor was a passive bystander to the violence.

actor2_victim An integer class variable that denotes whether the actor was also a victim of the violence.

actor3 A character class variable that contains the name of the actor involved in the event.

actor3_id A unique actor identifier assigned by the coders.

actor3_type A character class variable that records the type of actor.

actor3_subtype A character class variable that records the subtype of actor.

actor3_party A character class variable that records the party affiliation of actor.

actor3_violence A character class variable that records all forms of violence used by the actor.

actor3_precision A numeric class variable that denotes how precisely the actor information is coded.

actor3_initiator An integer class variable that denotes whether the actor was the initiator of the violence.

actor3_perpetrator An integer class variable that denotes whether the actor was a perpetrator of the violence.

actor3_intervener An integer class variable that denotes whether the actor was an intervener in the violence.

actor3_bystander An integer class variable that denotes whether the actor was a passive bystander to the violence.

actor3_victim An integer class variable that denotes whether the actor was also a victim of the violence.

actor4 A character class variable that contains the name of the actor involved in the event.

actor4_id A unique actor identifier assigned by the coders.

actor4_type A character class variable that records the type of actor.

actor4_subtype A character class variable that records the subtype of actor.

actor4_party A character class variable that records the party affiliation of actor.

actor4_violence A character class variable that records all forms of violence used by the actor.

actor4_precision A numeric class variable that denotes how precisely the actor information is coded.

actor4_initiator An integer class variable that denotes whether the actor was the initiator of the violence.

actor4_perpetrator An integer class variable that denotes whether the actor was a perpetrator of the violence.

actor4_intervener An integer class variable that denotes whether the actor was an intervener in the violence.

actor4_bystander An integer class variable that denotes whether the actor was a passive bystander to the violence.

actor4_victim An integer class variable that denotes whether the actor was also a victim of the violence.

actor5 A character class variable that contains the name of the actor involved in the event.

actor5_id A unique actor identifier assigned by the coders.

actor5_type A character class variable that records the type of actor.

actor5_subtype A character class variable that records the subtype of actor.

actor5_party A character class variable that records the party affiliation of actor.

actor5_violence A character class variable that records all forms of violence used by the actor.

actor5_precision A numeric class variable that denotes how precisely the actor information is coded.

actor5_initiator An integer class variable that denotes whether the actor was the initiator of the violence.

actor5_perpetrator An integer class variable that denotes whether the actor was a perpetrator of the violence.

actor5_intervener An integer class variable that denotes whether the actor was an intervener in the violence.

actor5_bystander An integer class variable that denotes whether the actor was a passive bystander to the violence.

actor5_victim An integer class variable that denotes whether the actor was also a victim of the violence.

actor6 A character class variable that contains the name of the actor involved in the event.

actor6_id A unique actor identifier assigned by the coders.

actor6_type A character class variable that records the type of actor.

actor6_subtype A character class variable that records the subtype of actor.

actor6_party A character class variable that records the party affiliation of actor.

actor6_violence A character class variable that records all forms of violence used by the actor.

actor6_precision A numeric class variable that denotes how precisely the actor information is coded.

actor6_initiator An integer class variable that denotes whether the actor was the initiator of the violence.

actor6_perpetrator An integer class variable that denotes whether the actor was a perpetrator of the violence.

actor6_intervener An integer class variable that denotes whether the actor was an intervener in the violence.

actor6_bystander An integer class variable that denotes whether the actor was a passive bystander to the violence.

actor6_victim An integer class variable that denotes whether the actor was also a victim of the violence.

event_context A character class variable that records the context in which the violence took place.

target A character class variable that records the primary target of the violence

deaths_best An integer class variable that records the best estimated number of deaths.

deaths_low An integer class variable that records the lowest estimated number of deaths.

deaths_high An integer class variable that records the highest estimated number of deaths.

injuries_best An integer class variable that records the best estimated number of injured people.

injuries_low An integer class variable that records the lowest estimated number of injured people.

injuries_high An integer class variable that records the highest estimated number of injured people.

displacement An integer class variable that denotes whether the event resulted in displacement.

damage An integer class variable that denotes whether the event resulted in material destruction.

source A character class variable that records the source.

number_of_sources An integer class variable that records the number of sources the event is based on. Only relevant once the dataset is aggregated to the event level.

source_author A character class variable that records the author of the source.

source_type A character class variable that records the type of source.

source_classification An integer class variable that denotes how reputable the source is considered.

sampling An integer class variable that denotes whether the report was sampled from Factiva or another secondary source.

unit_of_analysis A character class variable that records the unit of analysis.

aggregation A character class variable that records the chosen aggregation model. Only relevant once the data is aggregated to the event level.

Source

The data set is based on newspaper articles identified through the Factiva news repository, as well as a range of human rights reports, election monitoring reports, and special commission reports.

`mean_dscore` *Calculate the mean divergence scores across event reports*

Description

This function calculates the mean divergence score for one or more variables grouped by an event identifier. The divergence score captures how often values for a given variable differ across event reports describing the same event.

Usage

```
mean_dscore(data, group_var, variables, normalize = FALSE, plot = FALSE)
```

Arguments

data	A data frame containing event report level data.
group_var	A character string naming the column that uniquely identifies events (e.g., "event_id").
variables	A character vector of column names to compute divergence scores for.
normalize	Logical, indicating whether to normalize the scores by the total number of unique values for each variable.
plot	Logical, indicating whether to return a ggplot object visualizing the scores.

Details

For each variable and event, the function computes the number of unique values reported, subtracts one, and averages these values across all events. This reflects how much inconsistency exists across sources. Optionally, the scores can be normalized by the total number of unique values observed for each variable across the dataset. The result is a long-format dataframe showing which variables are most sensitive to aggregation. A plotting option is also available.

Value

Either a tibble or a ggplot object, depending on the value of `plot`. If `plot = FALSE`, returns a tibble with two columns:

variable The name of each variable.

dscore The mean divergence score or normalized score.

If `plot = TRUE`, returns a lollipop-style plot showing divergence scores by variable.

Examples

```
df <- data.frame(
  event_id = c(1, 1, 2, 2, 3),
  country = c("US", "US", "UK", "UK", "CA"),
  actor1 = c("Actor A", "Actor B", "Actor B", "Actor C", "Actor D"),
  deaths_best = c(10, 20, 5, 15, 10)
)
mean_dscore(df, "event_id", c("country", "actor1", "deaths_best"), normalize = TRUE, plot = TRUE)
```

---

mean_range *Calculate the mean within-event range across event reports for numeric variables*

Description

This function calculates the mean range for one or more numeric variables grouped by an event identifier. It is useful for diagnosing aggregation sensitivity by assessing how much spread exists in numeric values reported across event reports concerning the same event.

Usage

```
mean_range(data, group_var, variables)
```

Arguments

data	A data frame containing event report level data.
group_var	A character string naming the column that uniquely identifies events (e.g., "event_id").
variables	A character vector of column names to compute ranges for. All specified variables must be numeric.

Details

For each variable and event, the function computes the range (i.e., the difference between the maximum and minimum) of values reported across event reports. These values are then averaged across all events to produce a single score per variable. The result is a long-format dataframe that shows which numeric variables exhibit the widest event report level disagreement.

Value

A tibble with two columns:

variable The name of each variable.

mean_range The mean range across events for that variable.

Examples

```
df <- data.frame(
  event_id = c(1, 1, 2, 2, 3),
  deaths_best = c(10, 20, 5, 15, 10)
)
mean_range(
  df,
  group_var = "event_id",
  variables = c("deaths_best")
)
```

---

mean_sd *Calculate the mean within-event standard deviation across event reports for numeric variables*

Description

This function calculates the mean standard deviation for one or more numeric variables grouped by an event identifier. It is useful for diagnosing aggregation sensitivity by assessing how much variation exists in numeric values reported across event reports concerning the same event.

Usage

```
mean_sd(data, group_var, variables)
```

Arguments

data	A data frame containing event report level data.
group_var	A character string naming the column that uniquely identifies events (e.g., "event_id").
variables	A character vector of column names to compute standard deviations for. All specified variables must be numeric.

Details

For each variable and event, the function computes the standard deviation of values reported across event reports These values are then averaged across all events to produce a single score per variable. The result is a long-format dataframe that shows which numeric variables exhibit the most event report level disagreement

Value

A tibble with two columns:

variable The name of each variable.

mean_sd The mean standard deviation across events for that variable.

Examples

```
df <- data.frame(
  event_id = c(1, 1, 2, 2, 3),
  country = c("US", "US", "UK", "UK", "CA"),
  actor1 = c("Actor A", "Actor B", "Actor B", "Actor C", "Actor D"),
  deaths_best = c(10, 20, 5, 15, 10)
)
mean_sd(
  df,
  group_var = "event_id",
  variables = c("deaths_best")
)
```

---

modal_confidence *Calculate the modal confidence across event reports*

Description

This function calculates the modal confidence score for one or more variables grouped by an event identifier. The modal confidence score captures how dominant the most common value is within each event — that is, the proportion of event reports that agree with the modal (most frequent) value for each variable.

Usage

```
modal_confidence(data, group_var, variables)
```

Arguments

data	A data frame containing event report level data.
group_var	A character string naming the column that uniquely identifies events (e.g., "event_id").
variables	A character vector of column names to assess modal confidence for.

Details

For each variable and event, the function computes the share of event reports that match the modal value. These proportions are then averaged across all events to produce a single score per variable. The result is a long-format dataframe that shows which variables tend to exhibit the greatest agreement in reporting.

Value

A tibble with two columns:

variable The name of each variable.

modal_confidence The average share of reports per event that match the modal value.

Examples

```
df <- data.frame(
  event_id = c(1, 1, 2, 2, 3),
  actor1 = c("A", "A", "B", "C", "D"),
  deaths_best = c(10, 10, 5, 15, 10)
)
modal_confidence(
  df,
  group_var = "event_id",
  variables = c("actor1", "deaths_best")
)
```

---

share_disagreement *Calculate the share of events with any disagreement across event reports*

Description

This function calculates the proportion of events for which two or more distinct values are reported for each specified variable. It is useful for identifying which variables are most commonly inconsistent across event reports describing the same event.

Usage

```
share_disagreement(data, group_var, variables)
```

Arguments

data	A data frame containing event report level data.
group_var	A character string naming the column that uniquely identifies events (e.g., "event_id").
variables	A character vector of column names to check for disagreement.

Details

For each event and variable, the function checks whether all values reported across event reports are identical. It then calculates the share of events for which at least two different values are reported. The result is a long-format dataframe that highlights which variables most frequently exhibit inter-source disagreement.

Value

A tibble with two columns:

variable The name of each variable.

share_disagreement The proportion of events with disagreement for that variable.

Examples

```
df <- data.frame(
  event_id = c(1, 1, 2, 2, 3),
  actor1 = c("Actor A", "Actor B", "Actor B", "Actor B", "Actor C"),
  deaths_best = c(10, 10, 5, 15, 10)
)
share_disagreement(
  df,
  group_var = "event_id",
  variables = c("actor1", "deaths_best")
)
```

---

small_maverick_event_report

A Subset of the MAVERICK Event Report Dataset

Description

This dataset contains 100 event reports from the MAVERICK event report dataset, arranged by ‘event_id‘. It is used for examples and vignettes in the ‘eventreport‘ package.

Usage

```
small_maverick_event_report
```

Format

A subset of the MAVERICK data frame with 100 rows and 10 columns:

id A unique event report identifier.

event_id A unique event identifier assigned by the coders. Needed to aggregate event reports into events.

country A character class variable that contains the name of the country in which the event took place.

date_start A character class variable that contains the earliest possible event date expressed in YYYY-MM-DD format.

city A character class variable that contains the name of the city or village in which the event took place.

location A character class variable that contains a text description of the most precise event location described in the report.

actor1 A character class variable that contains the name of the actor involved in the event.

deaths_best An integer class variable that records the best estimated number of deaths.

injuries_best An integer class variable that records the best estimated number of injured people.

source A character class variable that records the source. ...

Source

MAVERICK dataset

Index