Package: mccount (via r-universe)

May 13, 2026

Title Estimate Recurrent Event Burden with Competing Risks

Version 0.1.1.9000

Description Calculates mean cumulative count (MCC) to estimate the expected cumulative number of recurrent events per person over time in the presence of competing risks and censoring. Implements both the Dong-Yasui equation method and sum of cumulative incidence method described in Dong, et al. (2015) <doi:10.1093/aje/kwu289>. Supports inverse probability weighting for causal inference as outlined in Gaber, et al. (2023) <doi:10.1093/aje/kwad031>. Provides S3 methods for printing, summarizing, plotting, and extracting results. Handles grouped analyses and integrates with 'ggplot2' <`https://ggplot2.tidyverse.org/`> for visualization.

License MIT + file LICENSE

URL `https://github.com/KennethATaylor/mccount`, `https://kennethataylor.github.io/mccount/`

BugReports `https://github.com/KennethATaylor/mccount/issues`

Depends R (>= 4.1.0)

Imports cards, cli, cmprsk, data.table, dplyr, glue, lifecycle, rlang, stringr, tibble, tidyr

Suggests boot, ggplot2 (>= 3.5.0), knitr, MatchIt, mstate, optmatch, patchwork, rmarkdown, spelling, survival, testthat (>= 3.0.0), WeightIt, withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/Needs/website rmarkdown

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev

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

Date/Publication 2025-11-14 14:01:35 UTC

RemoteUrl https://github.com/kennethataylor/mccount

RemoteRef HEAD

RemoteSha e2222cbf5c2942fa42709a53d3083a36d6520226

Contents

`as.data.frame.mcc` *Convert* `mcc` *object to data.frame*

Description

Extracts the MCC estimates from an `mcc` object and returns them as a standard data.frame. This is useful for further analysis or when working with packages that expect standard data.frame objects.

Usage

```
## S3 method for class 'mcc'
as.data.frame(x, ...)
```

Arguments

`x`	An `mcc` object
`...`	Additional arguments (currently unused)

Value

A data.frame with MCC estimates

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
) |>
  arrange(id, time)

# Calculate MCC
mcc_result <- mcc(df, "id", "time", "cause")

# Convert to data.frame
mcc_df <- as.data.frame(mcc_result)
print(mcc_df)
class(mcc_df)  # "data.frame"

# This is equivalent to extracting mcc_final
identical(mcc_df, as.data.frame(mcc_result$mcc_final))

# Useful for further analysis with base R functions
summary(mcc_df)
plot(mcc_df$time, mcc_df$mcc, type = "s")

# Clean up
rm(df, mcc_result, mcc_df)
```

`as_mcc` *Convert objects to* `mcc` *class*

Description

Converts objects to MCC class. This is useful when you have calculation results from other sources that you want to treat as MCC objects.

Usage

```
as_mcc(x, method, weighted = FALSE, by_group = NULL, call = NULL, ...)
```

Arguments

`x`	Object to convert to `mcc`
`method`	Method used for calculation ("equation" or "sci")
`weighted`	Logical indicating if weighted estimation was used
`by_group`	Optional name of grouping variable
`call`	Optional function call to store
`...`	Additional arguments (currently unused)

Value

An `mcc` S3 object

Examples

```
# Convert a data.frame to MCC object
library(dplyr)

# Create a simple data.frame with MCC results
mcc_data <- data.frame(
  time = c(1, 2, 3, 4, 5),
  mcc = c(0.1, 0.3, 0.5, 0.7, 1.0)
)

# Convert to MCC object (equation method)
mcc_obj <- as_mcc(mcc_data, method = "equation")
print(mcc_obj)
is_mcc(mcc_obj)  # TRUE

# Convert for SCI method (requires SumCIs column)
sci_data <- data.frame(
  time = c(1, 2, 3, 4, 5),
  SumCIs = c(0.1, 0.3, 0.5, 0.7, 1.0)
)

mcc_sci_obj <- as_mcc(sci_data, method = "sci")
print(mcc_sci_obj)

# Convert a list to MCC object
mcc_list <- list(
  mcc_final = data.frame(
    time = c(1, 2, 3),
    mcc = c(0.2, 0.5, 0.8)
  )
)

mcc_from_list <- as_mcc(mcc_list, method = "equation")
print(mcc_from_list)

# Clean up
rm(mcc_data, sci_data, mcc_list, mcc_obj, mcc_sci_obj, mcc_from_list)
```

`autoplot.mcc` *Auto-plot method for* `mcc` *objects*

Description

Convenience function that automatically creates an appropriate plot for `mcc` objects. This is called when using the base R `plot()` function.

Usage

```
## S3 method for class 'mcc'
autoplot(x, ...)
```

Arguments

`x`	An `mcc` object
`...`	Additional arguments passed to plot.mcc

Value

A ggplot2 object

Examples

```
# Create sample data
library(dplyr)
library(ggplot2)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2),
  treatment = c("Control", "Control", "Treatment", "Treatment",
                "Treatment", "Treatment", "Treatment", "Control", "Control")
) |>
  arrange(id, time)

# Calculate MCC
mcc_result <- mcc(df, "id", "time", "cause", by = "treatment")

# Use autoplot (ggplot2 style)
p <- autoplot(mcc_result)
print(p)

# Customize with ggplot2 functions
p_custom <- autoplot(mcc_result) +
  theme_classic() +
  labs(caption = "Data from hypothetical study") +
```

```
  geom_hline(yintercept = 1, linetype = "dashed", alpha = 0.5)

print(p_custom)

# Clean up
rm(df, mcc_result, p, p_custom)
```

`compare_mcc` *Compare* `mcc` *objects*

Description

Compares two `mcc` objects and returns a summary of differences. Useful for comparing results from different methods or parameter settings.

Usage

```
compare_mcc(x, y, tolerance = 1e-06)
```

Arguments

`x`	First `mcc` object
`y`	Second `mcc` object
`tolerance`	Numeric tolerance for comparing MCC values (default: 1e-6)

Value

A list summarizing the comparison

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
) |>
  arrange(id, time)

# Calculate MCC using different methods
mcc_eq <- mcc(df, "id", "time", "cause", method = "equation")
mcc_sci <- mcc(df, "id", "time", "cause", method = "sci")

# Compare the results
comparison <- compare_mcc(mcc_eq, mcc_sci)
print(comparison)
```

```
# Clean up
rm(df, mcc_eq, mcc_sci, comparison)
```

`filter_mcc` *Filter* `mcc` *object by groups*

Description

For grouped `mcc` objects, extracts results for specified groups only. This is useful for focusing on specific groups of interest or creating custom visualizations.

Usage

```
filter_mcc(x, groups)
```

Arguments

`x`	A grouped `mcc` object
`groups`	Character vector of group names to include

Value

An `mcc` object containing only the specified groups

Examples

```
# Create sample data with groups
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5, 6, 6, 7, 8),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3, 4, 5, 9, 2),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2, 1, 0, 0, 2),
  treatment = c("Control", "Control", "Treatment", "Treatment",
                "Treatment", "Treatment", "Treatment", "Control", "Control",
                "Placebo", "Placebo", "Placebo", "Placebo")
) |>
  arrange(id, time)

# Grouped analysis
mcc_full <- mcc(df, "id", "time", "cause", by = "treatment")

# Show all groups
mcc_groups(mcc_full)

# Filter to specific groups
mcc_filtered <- filter_mcc(mcc_full, c("Control", "Treatment"))
mcc_groups(mcc_filtered)  # Only "Control" and "Treatment"
```

```
# Plot the filtered mcc object
plot(mcc_filtered)

# Clean up
rm(df, mcc_full, mcc_filtered)
```

`geom_line_mcc` *Add Reference Lines at an MCC Threshold to ggplot2 Objects*

Description

Adds horizontal and vertical reference lines to mark when the Mean Cumulative Count (MCC) reaches the `threshold`. This function returns a list of `ggplot2` geoms that can be added to existing plots using the `+` operator. For grouped analyses, it creates separate reference lines for each group.

Usage

```
geom_line_mcc(
  mcc_object,
  threshold = 1,
  linetype = 2,
  color = NULL,
  alpha = 0.7,
  linewidth = 0.5,
  show_labels = FALSE,
  label_size = 3,
  label_nudge_x = 0,
  label_nudge_y = 0.05
)
```

Arguments

`mcc_object`	An object of class `mcc` containing MCC estimates.
`threshold`	numeric;determines MCC value threshold to use (default = `1.0`)
`linetype`	Line type for the reference lines. Default is `2` (dashed). Can be numeric (1-6) or character ("solid", "dashed", "dotted", etc.).
`color`	Color for the reference lines. If `NULL` (default), uses gray.
`alpha`	Transparency level for the reference lines. Default is `0.7`.
`linewidth`	Width of the reference lines. Default is `0.5`.
`show_labels`	Logical indicating whether to add text labels at the intersection points. Default is `FALSE`.
`label_size`	Size of the text labels if `show_labels` = `TRUE`. Default is `3`.
`label_nudge_x`	Horizontal offset for labels. Default is `0`.
`label_nudge_y`	Vertical offset for labels. Default is `0.05`.

Details

This function identifies the time when MCC first reaches or exceeds the specified MCC `threshold`. It then creates:

- A horizontal line from x = 0 to the time when MCC = `threshold`
- A vertical line from y = 0 to MCC = `threshold` at that time point

For grouped analyses, separate reference lines are created for each group that reaches MCC = `threshold`. Groups that never reach MCC = `threshold` will not have reference lines added.

The function is designed to work seamlessly with the existing `plot.mcc()` method and can be chained using ggplot2's + syntax.

Value

A `ggplot2` layer object that can be added to a ggplot using the + operator.

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2),
  group = c("A", "A", "B", "B", "B", "B", "B", "A", "A")
) |>
  arrange(id, time)

# Ungrouped analysis
mcc_overall <- mcc(df, "id", "time", "cause")

# Basic plot with reference lines
plot(mcc_overall) +
  geom_line_mcc(mcc_overall) +
  labs(title = "MCC with Reference Lines at 1.0")

# Grouped analysis
mcc_grouped <- mcc(df, "id", "time", "cause", by = "group")

# Plot with group-specific reference lines
plot(mcc_grouped) +
  geom_line_mcc(mcc_grouped, linetype = "dotted", alpha = 0.8) +
  labs(title = "Grouped MCC with Reference Lines")

# With labels
plot(mcc_overall) +
  geom_line_mcc(mcc_overall, show_labels = TRUE, color = "red") +
  labs(title = "MCC with Labeled Reference Lines")

# Clean up
rm(df, mcc_overall, mcc_grouped)
```

`is_grouped` *Check if* `mcc` *object is from grouped analysis*

Description

Check if `mcc` object is from grouped analysis

Usage

```
is_grouped(x)
```

Arguments

`x`	An `mcc` object

Value

Logical indicating whether the analysis was grouped

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2),
  group = c("A", "A", "B", "B", "B", "B", "B", "A", "A")
) |>
  arrange(id, time)

# Ungrouped analysis
mcc_ungrouped <- mcc(df, "id", "time", "cause")
is_grouped(mcc_ungrouped)  # FALSE

# Grouped analysis
mcc_grouped <- mcc(df, "id", "time", "cause", by = "group")
is_grouped(mcc_grouped)  # TRUE

# Clean up
rm(df, mcc_ungrouped, mcc_grouped)
```

`is_mcc` *Check if object is an* `mcc` *result*

Description

Check if object is an `mcc` result

Usage

```
is_mcc(x)
```

Arguments

`x`	An object to test

Value

TRUE if x is an `mcc` object, FALSE otherwise

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)

# Calculate MCC
mcc_result <- mcc(df, "id", "time", "cause")

# Test if it's an MCC object
is_mcc(mcc_result)  # TRUE

# Clean up
rm(df, mcc_result)
```

`is_weighted` *Check if* `mcc` *object uses weighted estimation*

Description

Check if `mcc` object uses weighted estimation

Usage

```
is_weighted(x)
```

Arguments

`x`	An `mcc` object

Value

Logical indicating whether weighted estimation was used

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
) |>
  arrange(id, time)

# Calculate unweighted MCC
mcc_unweighted <- mcc(df, "id", "time", "cause")
is_weighted(mcc_unweighted)  # FALSE

# Create weighted data
df_weighted <- df |>
  group_by(id) |>
  slice(1) |>
  ungroup() |>
  mutate(weights = runif(n(), 0.5, 2.0)) |>
  select(id, weights) |>
  right_join(df, by = "id") |>
  arrange(id, time)

# Calculate weighted MCC
mcc_weighted <- mcc(df_weighted, "id", "time", "cause", weights = "weights")
is_weighted(mcc_weighted)  # TRUE

# Clean up
rm(df, df_weighted, mcc_unweighted, mcc_weighted)
```

`mcc` *Calculate Mean Cumulative Count (MCC)*

Description

Calculates the mean cumulative count (MCC), which estimates the expected cumulative number of events per person over time, while accounting for potential competing risks and censoring. This function provides a unified interface to two different estimation approaches: the Dong-Yasui (`"equation"`) method and the sum of cumulative incidence (`"sci"`) method.

The `"equation"` method calculates MCC directly through probability calculations, while the `"sci"` method derives MCC by summing the cumulative incidence functions for each recurrent event. The two approaches yield equivalent results in certain circumstances. When they do not, the choice between methods depends on the specific outcome, analysis needs, and data structure. See `vignette("choosing-between-methods")` for more details.

Usage

```
mcc(
  data,
  id_var,
  time_var,
  cause_var,
  by = NULL,
  method = c("equation", "sci"),
  tstart_var = NULL,
  weights = NULL,
  adjust_times = TRUE,
  time_precision = 1e-06,
  include_details = TRUE
)
```

Arguments

`data`	(`data.frame` or `tbl_df`) A `data.frame` or tibble containing the required variables
`id_var`	(`string`) Name of the column containing participant identifiers
`time_var`	(`string`) Name of the column containing follow-up times
`cause_var`	(`string`) Name of the column containing event indicator values (1 = event of interest, 2 = competing risk, 0 = censoring)
`by`	(`string`, optional) Name of the column to group by for calculating MCC within subgroups. If provided, MCC will be calculated separately for each level of this variable

`method`	(`string`) Method to use for MCC calculation. Either `"equation"` (default) or `"sci"` (sum of cumulative incidence)
`tstart_var`	(`string`) Name of the column containing start times of follow-up for incorporating left truncation. Only allowed to be specified when `method = "sci"`. If `NULL` (default), a constant value of `0` is used in calculation (i.e., right truncation only)
`weights`	(`string`, optional) Name of the column containing weights for weighted MCC estimation. Currently only supported with `method = "equation"`. If provided, all weights must be non-negative and non-missing
`adjust_times`	(`logical`) If `TRUE` (default), automatically adjusts times to account for outcome events and competing risk events occurring at the same time
`time_precision`	(`numeric`) Precision used for adjusting simultaneous events (default: 1e-6). Must be a positive numeric value
`include_details`	(`logical`) Whether to include detailed calculation tables and intermediate objects in the output. Default is `TRUE`, which returns all calculation details. Setting to `FALSE` returns only the final MCC estimates, making the function more efficient for bootstrapping

Value

An S3 object of class `"mcc"` with method-specific subclasses. The object contains:

When `include_details = TRUE` (default):

For `method = "equation"`:

- `mcc_final`: A tibble with columns for `time` and `mcc`
- `mcc_table`: A tibble with detailed calculation steps
- `original_data`: The input `data` with standardized column names
- `adjusted_data`: Present only if time adjustments were applied

For `method = "sci"`:

- `mcc_final`: A tibble with columns for `time` and MCC (expressed as `SumCIs`)
- `sci_table`: A tibble with cumulative incidence for each event number and their sum
- `all_cis`: A list of cumulative incidence data for each event number
- `mcc_base`: A tibble with calculation details for the MCC
- `original_data`: The input `data` with standardized column names
- `adjusted_data`: Present only if time adjustments were applied

When `include_details = FALSE`:

- `mcc_final`: A tibble with columns for `time` and `mcc` (or `SumCIs` for `method` = `"sci"`)

All objects include metadata:

- `method`: The method used for calculation
- `weighted`: Logical indicating whether weighted estimation was used
- `by_group`: Name of grouping variable (for grouped analyses)
- `call`: The original function call

When by is specified, all tibbles contain an additional column with the grouping variable values, and the object has the additional class `"mcc_grouped"`.

References

Dong H, Robison LL, Leisenring WM, Martin LJ, Armstrong GT, Yasui Y. Estimating the burden of recurrent events in the presence of competing risks: the method of mean cumulative count. *Am J Epidemiol*. 2015 Apr 1;181(7):532-40. doi: 10.1093/aje/kwu289

Gaber CE, Edwards JK, Lund JL, Peery AF, Richardson DB, Kinlaw AC. Inverse Probability Weighting to Estimate Exposure Effects on the Burden of Recurrent Outcomes in the Presence of Competing Events. *Am J Epidemiol*. 2023;192(5):830-839. doi: 10.1093/aje/kwad031

Examples

```
# Attach dplyr
library(dplyr)
# Create sample data with recurrent events
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3), # Times will be adjusted for id = 5
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)  # Sort the data by id and time

# Print the dataset
print("Hypothetical dataset from Dong et al. (2015):")
print(df)

# Calculate MCC using the equation method
mcc_eq <- mcc(df, id_var = "id", time_var = "time", cause_var = "cause")

# Print the S3 object
mcc_eq

# Get summary
summary(mcc_eq)

# Extract MCC estimates
mcc_estimates(mcc_eq)

# Extract calculation details
mcc_details(mcc_eq)
```

```
# Calculate MCC using the sum of cumulative incidence method
mcc_sci <- mcc(
  df,
  id_var = "id",
  time_var = "time",
  cause_var = "cause",
  method = "sci"
)

mcc_sci

# Clean up
rm(df, mcc_eq, mcc_sci)
```

`mcc_details` *Extract calculation details from* `mcc` *objects*

Description

Extract calculation details from `mcc` objects

Usage

```
mcc_details(x, ...)
```

Arguments

`x`	An `mcc` object
`...`	Additional arguments (currently unused)

Value

A tibble with calculation details, or NULL if not available

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)

# Calculate MCC with details
mcc_eq <- mcc(df, "id", "time", "cause", method = "equation")
```

```
mcc_sci <- mcc(df, "id", "time", "cause", method = "sci")

# Extract calculation details
details_eq <- mcc_details(mcc_eq)   # Returns mcc_table
details_sci <- mcc_details(mcc_sci) # Returns sci_table

print(details_eq)
print(details_sci)

# Clean up
rm(df, mcc_eq, mcc_sci, details_eq, details_sci)
```

`mcc_estimates` *Extract MCC estimates from* `mcc` *objects*

Description

Extract MCC estimates from `mcc` objects

Usage

```
mcc_estimates(x, ...)
```

Arguments

`x`	An `mcc` object
`...`	Additional arguments (currently unused)

Value

A tibble with MCC estimates

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)

# Calculate MCC
mcc_result <- mcc(df, "id", "time", "cause")

# Extract MCC estimates
estimates <- mcc_estimates(mcc_result)
```

```
print(estimates)

# For grouped analysis
df_grouped <- df |>
  mutate(group = c("A", "A", "B", "B", "B", "B", "B", "A", "A"))

mcc_grouped <- mcc(df_grouped, "id", "time", "cause", by = "group")
estimates_grouped <- mcc_estimates(mcc_grouped)
print(estimates_grouped)

# Clean up
rm(df, df_grouped, mcc_result, mcc_grouped, estimates, estimates_grouped)
```

`mcc_final_values` *Get final MCC value for each group*

Description

Extracts the final (maximum time) MCC value for each group in a grouped analysis, or the overall final MCC value for ungrouped analyses.

Usage

```
mcc_final_values(x)
```

Arguments

`x`	An `mcc` object

Value

A named numeric vector with final MCC values

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 1, 2),
  group = c("A", "A", "B", "B", "B", "B", "A", "A")
) |>
  arrange(id, time)

# Ungrouped analysis
mcc_ungrouped <- mcc(df, "id", "time", "cause")
mcc_final_values(mcc_ungrouped)
```

```
# Grouped analysis
mcc_grouped <- mcc(df, "id", "time", "cause", by = "group")
mcc_final_values(mcc_grouped)

# Clean up
rm(df, mcc_ungrouped, mcc_grouped)
```

`mcc_grouping_var` *Get grouping variable name from grouped* `mcc` *object*

Description

Get grouping variable name from grouped `mcc` object

Usage

```
mcc_grouping_var(x)
```

Arguments

`x`	An `mcc` object

Value

Character string with grouping variable name, or NULL if not grouped

Examples

```
# Create sample data with groups
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2),
  treatment = c("Control", "Control", "Treatment", "Treatment",
                "Treatment", "Treatment", "Treatment", "Control",
                "Control")
) |>
  arrange(id, time)

# Grouped analysis
mcc_grouped <- mcc(df, "id", "time", "cause", by = "treatment")

# Get grouping variable name
mcc_grouping_var(mcc_grouped)  # "treatment"

# Clean up
rm(df, mcc_grouped)
```

`mcc_groups` *Extract unique groups from grouped `mcc` object*

Description

Extract unique groups from grouped `mcc` object

Usage

```
mcc_groups(x)
```

Arguments

`x`	An `mcc` object

Value

Character vector of unique group values, or NULL if not grouped

Examples

```
# Create sample data with groups
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 5, 5, 6, 7, 8),
  time = c(8, 1, 5, 2, 6, 7, 3, 3, 4, 9, 2),
  cause = c(0, 0, 2, 1, 1, 1, 1, 2, 1, 0, 2),
  treatment = c("Control", "Control", "Treatment", "Treatment",
                "Treatment", "Treatment", "Control", "Control",
                "Placebo", "Placebo", "Placebo")
) |>
  arrange(id, time)

# Grouped analysis
mcc_grouped <- mcc(df, "id", "time", "cause", by = "treatment")

# Get all unique groups
mcc_groups(mcc_grouped)  # "Control", "Placebo", "Treatment"

# Clean up
rm(df, mcc_grouped)
```

`mcc_method` *Get the method used for MCC calculation*

Description

Get the method used for MCC calculation

Usage

```
mcc_method(x)
```

Arguments

`x`	An `mcc` object

Value

Character string indicating the method ("equation" or "sci")

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2),
  group = c("A", "A", "B", "B", "B", "B", "B", "A", "A")
) |>
  arrange(id, time)

# Calculate MCC
mcc_result <- mcc(df, "id", "time", "cause")

# Get the method used
mcc_method(mcc_result)

# Clean up
rm(df, mcc_result)
```

`plot.mcc` *Plot MCC results*

Description

Creates plots for Mean Cumulative Count (MCC) results. The plotting method automatically adapts based on the `mcc` object class and whether the analysis was grouped.

Usage

```
## S3 method for class 'mcc'
plot(
  x,
  type = c("mcc", "components"),
  groups = NULL,
  conf_int = FALSE,
  colors = NULL,
  title = NULL,
  subtitle = NULL,
  ...
)
```

Arguments

`x`	An `mcc` object
`type`	Character string specifying plot type: • "mcc" (default): Plot MCC estimates over time • "components": Show individual cumulative incidence components (SCI method only)
`groups`	Character vector specifying which groups to include in grouped analyses. If NULL (default), all groups are included
`conf_int`	Logical indicating whether to include confidence intervals if available
`colors`	Character vector of colors to use for groups. If NULL, uses default colors
`title`	Character string for plot title. If NULL, generates automatic title
`subtitle`	Character string for plot subtitle. If NULL, generates automatic subtitle
`...`	Additional arguments passed to ggplot2 functions

Value

A ggplot2 object

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2),
  group = c("A", "A", "B", "B", "B", "B", "B", "A", "A")
) |>
  arrange(id, time)

# Basic MCC plot (ungrouped)
mcc_result <- mcc(df, "id", "time", "cause")
plot(mcc_result)

# Grouped analysis with custom colors
mcc_grouped <- mcc(df, "id", "time", "cause", by = "group")
plot(mcc_grouped)

# Customize the grouped plot
plot(mcc_grouped,
     colors = c("red", "blue"),
     title = "MCC by Treatment Group",
     subtitle = "Comparison of Event Burden")

# Plot only specific groups
plot(mcc_grouped, groups = c("A"))

# Compare different methods - equation method only shows MCC
mcc_eq <- mcc(df, "id", "time", "cause", method = "equation")
plot(mcc_eq)

# SCI method can show components of cumulative incidence components
mcc_sci <- mcc(df, "id", "time", "cause", method = "sci")
plot(mcc_sci)  # Shows main MCC plot
plot(mcc_sci, type = "components")  # Shows CI components

# Clean up
rm(df, mcc_result, mcc_grouped, mcc_eq, mcc_sci)
```

`print.mcc` *Print method for* `mcc` *objects*

Description

Print method for `mcc` objects

Usage

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

Arguments

`x`	An `mcc` object
`...`	Additional arguments (currently unused)

Value

x invisibly

Examples

```
# Attach dplyr
library(dplyr)
# Create sample data with recurrent events
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)  # Sort the data by id and time

# Calculate MCC using the equation method (default)
mcc_eq <- mcc(df, id_var = "id", time_var = "time", cause_var = "cause")

# Print the S3 object (uses print.mcc method)
mcc_eq

# Calculate MCC using the sum of cumulative incidence method
mcc_sci <- mcc(
  df,
  id_var = "id",
  time_var = "time",
  cause_var = "cause",
  method = "sci"
)

# Print the S3 object
mcc_sci

# Clean up
rm(df, mcc_eq, mcc_sci)
```

`print.mcc_comparison` *Print method for MCC comparison objects*

Description

Print method for MCC comparison objects

Usage

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

Arguments

`x`	An mcc_comparison object
`...`	Additional arguments (currently unused)

Value

x invisibly

Examples

```
# Create sample data
library(dplyr)
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
) |>
  arrange(id, time)

# Calculate MCC using different methods
mcc_eq <- mcc(df, "id", "time", "cause", method = "equation")
mcc_sci <- mcc(df, "id", "time", "cause", method = "sci")

# Compare the results
comparison <- compare_mcc(mcc_eq, mcc_sci)
print(comparison)

# Clean up
rm(df, mcc_eq, mcc_sci, comparison)
```

`print.summary.mcc` *Print method for* `mcc` *summary objects*

Description

Print method for `mcc` summary objects

Usage

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

Arguments

`x`	A `summary.mcc` object
`...`	Additional arguments (currently unused)

Value

Invisibly returns `x`

Examples

```
# Attach dplyr
library(dplyr)
# Create sample data with recurrent events
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)  # Sort the data by id and time

# Calculate MCC using the equation method (default)
mcc_eq <- mcc(df, id_var = "id", time_var = "time", cause_var = "cause")

summary(mcc_eq)

# Calculate MCC using the sum of cumulative incidence method
mcc_sci <- mcc(
  df,
  id_var = "id",
  time_var = "time",
  cause_var = "cause",
  method = "sci"
)

print(summary(mcc_sci))
```

```
# Clean up
rm(df, mcc_eq, mcc_sci)
```

`summary.mcc` *Summary method for* `mcc` *objects*

Description

Summary method for `mcc` objects

Usage

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

Arguments

`object`	An `mcc` object
`...`	Additional arguments (currently unused)

Value

A summary object with class `summary.mcc`

Examples

```
# Attach dplyr
library(dplyr)
# Create sample data with recurrent events
df <- data.frame(
  id = c(1, 2, 3, 4, 4, 4, 4, 5, 5),
  time = c(8, 1, 5, 2, 6, 7, 8, 3, 3),
  cause = c(0, 0, 2, 1, 1, 1, 0, 1, 2)
 ) |>
  arrange(id, time)  # Sort the data by id and time

# Calculate MCC using the equation method (default)
mcc_eq <- mcc(df, id_var = "id", time_var = "time", cause_var = "cause")

summary(mcc_eq)

# Calculate MCC using the sum of cumulative incidence method
mcc_sci <- mcc(
  df,
  id_var = "id",
  time_var = "time",
  cause_var = "cause",
  method = "sci"
```

```
)

summary(mcc_sci)

# Clean up
rm(df, mcc_eq, mcc_sci)
```

Index