

Package: AURMC (via r-universe)

October 29, 2026

Title Area Under the Repeated Measures Curve

Version 0.5.0

Description Estimates and compares areas under curves formed by repeatedly measured outcomes observed subject to censoring and a terminal event. Implements Kaplan-Meier adjustment for censoring, influence-function standard errors, Wald inference, and multiplier perturbation inference, with utilities for interpolation, simulation, tabulation, and plotting. For additional details, see: McCaw ZR, Armbruster S, Tian L, Claggett BL, Fitzmaurice G, Foà A, Pfeffer MA, Wei LJ (2026) <[doi:10.1016/j.jchf.2026.103013](https://doi.org/10.1016/j.jchf.2026.103013)>.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests knitr, markdown, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

LinkingTo RcppArmadillo, Rcpp

Imports cowplot, dplyr, ggplot2, methods, Rcpp, tibble

VignetteBuilder knitr

URL <https://github.com/zrmacc/AURMC>

BugReports <https://github.com/zrmacc/AURMC/issues>

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

Date/Publication 2026-10-16 17:03:10 UTC

RemoteUrl <https://github.com/zrmacc/aurmc>

RemoteRef HEAD

RemoteSha 9d4b465e187862d3b2817c9392906d4a076ad4fa

Contents

AtRiskMatrixR	2
AURMC	3
AURMC-class	4
BootstrapSamplesR	4
CalcMartingaleR	5
CalcMuR	6
CensorAfterLast	6
CompareAURMCs	7
Curve	8
DrawBootstrapR	9
EstimatorR	9
GenData	10
InfluenceR	11
Interpolate	12
InterpolateR	13
KaplanMeierR	14
PerturbationR	14
PlotTwoSample	15
print.AURMC	17
show,AURMC-method	17
TabRMC	18
ValueMatrixR	19
Index	20

AtRiskMatrixR	<i>Tabulate At-Risk Matrix R</i>
---------------	----------------------------------

Description

Construct a subject (row) by evaluation time (col) matrix.

Usage

```
AtRiskMatrixR(eval_times, idx, time)
```

Arguments

- | | |
|------------|-----------------------|
| eval_times | Evaluation times. |
| idx | Unique subject index. |
| time | Observation time. |

Value

Numeric matrix.

AURMC

*Area Under the Repeated Measures Curve***Description**

Area Under the Repeated Measures Curve

Usage

```
AURMC(
  data,
  alpha = 0.05,
  censor_after_last = TRUE,
  idx_name = "idx",
  int_method = "trapezoid",
  perturbations = NULL,
  random_state = 0,
  status_name = "status",
  tau = NULL,
  time_name = "time",
  value_name = "value"
)
```

Arguments

<code>data</code>	Data.frame.
<code>alpha</code>	Type I error.
<code>censor_after_last</code>	Introduce censoring after the last event <i>if</i> no observation-terminating event is present.
<code>idx_name</code>	Name of column containing a unique subject index.
<code>int_method</code>	Integration method, selected from "left", "right", "trapezoid".
<code>perturbations</code>	Number of perturbations to use for bootstrap inference. If NULL, only analytical inference is performed.
<code>random_state</code>	Seed to ensure perturbations are reproducible.
<code>status_name</code>	Name of column containing the status. Must be coded as 0 for censoring, 1 for a measurement, 2 for death. Each subject should have an observation-terminating event, either censoring or death.
<code>tau</code>	Truncation time.
<code>time_name</code>	Name of column containing the observation time.
<code>value_name</code>	Name of the column containing the measurement.

Value

A data.frame containing the method, truncation time, area estimate, standard error, confidence limits, and p-value.

Examples

```
example_data <- data.frame(
  idx = rep(1:3, each = 2),
  time = rep(c(0, 1), 3),
  status = rep(c(1, 0), 3),
  value = c(1, 1, 2, 2, 3, 3)
)
AURMC(example_data, tau = 1)
```

AURMC-class

AURMC Object

Description

AURMC Object

Slots

Arm0 Results for the reference arm.

Arm1 Results for the treatment arm.

Contrast Contrasts.

BootstrapSamplesR

Bootstrap Samples R

Description

Returns a bootstrap (row) by statistic (col) matrix. If return_auc = TRUE, the output has a single column, the AUC. If return_auc = FALSE, the output has a single column for each evaluation time.

Usage

```
BootstrapSamplesR(
  boot,
  eval_times,
  idx,
  status,
  time,
  value,
  int_method = "trapezoid",
```

```

    replace_na = FALSE,
    return_auc = FALSE,
    trunc_time = NULL
  )

```

Arguments

<code>boot</code>	Bootstrap replicates.
<code>eval_times</code>	Evaluation times.
<code>idx</code>	Unique subject index.
<code>status</code>	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
<code>time</code>	Observation time.
<code>value</code>	Observation value.
<code>int_method</code>	Integration method, selected from "left", "right", "trapezoid".
<code>replace_na</code>	Replace NaN with zero?
<code>return_auc</code>	Return the AUC?
<code>trunc_time</code>	Truncation time? Optional. If omitted, defaults to the maximum evaluation time.

Value

Numeric matrix.

CalcMartingaleR	<i>Calculate Martingale</i>
-----------------	-----------------------------

Description

Calculate $dM_i(t) = dN_i(t) - Y_i(t)d\Lambda(t)$.

Usage

```
CalcMartingaleR(haz, idx, status, time, unique_times)
```

Arguments

<code>haz</code>	Value of the hazard at each unique time.
<code>idx</code>	Subject index.
<code>status</code>	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
<code>time</code>	Subject observation times.
<code>unique_times</code>	Unique times at which to obtain the martingale.

Value

Matrix with subjects as rows and unique times as columns.

CalcMuR	<i>Calculate Mu R</i>
---------	-----------------------

Description

Evaluate the quadrature tail associated with $\mu(t; \tau) = \int_t^\tau S(u-)d(u)/y(u)du$. The current grid point is excluded because the hazard jump at t affects $\hat{S}(u-)$ only for $u > t$.

Usage

CalcMuR(d, surv, unique_times, y, int_method = "trapezoid")

Arguments

- d Value of d(t) at each time point.
- surv Value of $S(t-)$ at each time point.
- unique_times Unique values of time t.
- y Value of y(t) at each time point.
- int_method Integration method, selected from "left", "right", "trapezoid".

Value

Numeric vector of $\mu(t; \tau)$.

CensorAfterLast	<i>Censor After Last</i>
-----------------	--------------------------

Description

Introduce a censoring after the last event if no observation terminating event is present.

Usage

CensorAfterLast(data)

Arguments

- data Data.frame.

Value

None.

CompareAURMCs

*Compare Areas Under Repeated Measures Curves***Description**

Compare Areas Under Repeated Measures Curves

Usage

```
CompareAURMCs(
  data,
  alpha = 0.05,
  arm_name = "arm",
  censor_after_last = TRUE,
  idx_name = "idx",
  int_method = "trapezoid",
  perturbations = NULL,
  random_state = 0,
  status_name = "status",
  tau = NULL,
  time_name = "time",
  value_name = "value"
)
```

Arguments

<code>data</code>	Data.frame.
<code>alpha</code>	Type I error.
<code>arm_name</code>	Name of the column containing treatment arm. Coded as 0 for reference, 1 for treatment.
<code>censor_after_last</code>	Introduce censoring after the last event <i>if</i> no observation-terminating event is present.
<code>idx_name</code>	Name of column containing a unique subject index.
<code>int_method</code>	Integration method, selected from "left", "right", "trapezoid".
<code>perturbations</code>	Number of perturbations to use for bootstrap inference. If NULL, only analytical inference is performed.
<code>random_state</code>	Seed to ensure perturbations are reproducible.
<code>status_name</code>	Name of column containing the status. Must be coded as 0 for censoring, 1 for a measurement, 2 for death. Each subject should have an observation-terminating event, either censoring or death.
<code>tau</code>	Truncation time.
<code>time_name</code>	Name of column containing the observation time.
<code>value_name</code>	Name of the column containing the measurement.

Value

An object of class AURMC containing arm-specific estimates and difference and ratio contrasts. Ratio inference is returned as NA when either arm-specific area is non-positive.

Examples

```
example_data <- data.frame(
  idx = rep(1:4, each = 2),
  arm = rep(c(0, 0, 1, 1), each = 2),
  time = rep(c(0, 1), 4),
  status = rep(c(1, 0), 4),
  value = rep(c(1, 1, 2, 2), each = 2)
)
CompareAURMCs(example_data, tau = 1)
```

Curve	<i>Estimate Curve</i>
-------	-----------------------

Description

Estimate Curve

Usage

```
Curve(
  data,
  censor_after_last = TRUE,
  idx_name = "idx",
  status_name = "status",
  time_name = "time",
  value_name = "value"
)
```

Arguments

data	Data.frame.
censor_after_last	Introduce censoring after the last event <i>if</i> no observation-terminating event is present.
idx_name	Name of column containing a unique subject index.
status_name	Name of column containing the status. Must be coded as 0 for censoring, 1 for a measurement, 2 for death. Each subject should have an observation-terminating event, either censoring or death.
time_name	Name of column containing the observation time.
value_name	Name of the column containing the measurement.

Value

Function.

DrawBootstrapR	<i>Draw Bootstrap R</i>
----------------	-------------------------

Description

Draw Bootstrap R

Usage

```
DrawBootstrapR(idx, status, time, value)
```

Arguments

- | | |
|--------|--|
| idx | Unique subject index. |
| status | Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event. |
| time | Observation time. |
| value | Observation value. |

Value

Numeric matrix.

EstimatorR	<i>Tabulate Estimator R</i>
------------	-----------------------------

Description

Tabulate Estimator R

Usage

```
EstimatorR(  
  idx,  
  status,  
  time,  
  value,  
  eval_times = NULL,  
  int_method = "trapezoid",  
  replace_na = FALSE,  
  return_auc = FALSE,  
  trunc_time = NULL  
)
```

Arguments

<code>idx</code>	Unique subject index.
<code>status</code>	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
<code>time</code>	Observation time.
<code>value</code>	Observation value.
<code>eval_times</code>	Evaluation times. If omitted, defaults to the unique values of time.
<code>int_method</code>	Integration method, selected from "left", "right", "trapezoid".
<code>replace_na</code>	Replace NaN with zero? Default: FALSE.
<code>return_auc</code>	Return the AUC? Default: FALSE.
<code>trunc_time</code>	Truncation time? Optional. If omitted, defaults to the maximum evaluation time.

Value

Data.frame.

GenData

Simulate Repeated Measures Data

Description

Simulates repeated measures data at regular time points subject to exponential censoring and death.

Usage

```
GenData(
  censoring_rate,
  death_rate,
  n,
  tau,
  last_missing = FALSE,
  value_mean = 0,
  value_sd = 1
)
```

Arguments

<code>censoring_rate</code>	Rate for the time to censoring.
<code>death_rate</code>	Rate for the time to death.
<code>n</code>	Number of subjects.
<code>tau</code>	Truncation time.
<code>last_missing</code>	Should the last value be missing? Default: FALSE.
<code>value_mean</code>	Mean of measurement.
<code>value_sd</code>	Standard deviation of measurement.

Value

A data.frame with subject index, observation time, status, and repeated measurement value.

Examples

```
set.seed(1)
GenData(censoring_rate = 0.5, death_rate = 0.25, n = 5, tau = 2)
```

InfluenceR	<i>Influence Function R</i>
------------	-----------------------------

Description

Influence function contributions for the AUC. Includes the three component integrals (i1, i2, i3) and the overall influence psi.

Usage

```
InfluenceR(idx, status, time, trunc_time, value, int_method = "trapezoid")
```

Arguments

idx	Unique subject index.
status	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
time	Observation time.
trunc_time	Truncation time? Optional. If omitted, defaults to the maximum evaluation time.
value	Observation value.
int_method	Integration method, selected from "left", "right", "trapezoid".

Value

Data.frame.

Interpolate

*Interpolate***Description**

Linearly interpolates between each subject's measurements. The input data should contain no missing values.

Usage

```
Interpolate(
  data,
  grid,
  idx_name = "idx",
  status_name = "status",
  rm_na = TRUE,
  time_name = "time",
  value_name = "value"
)
```

Arguments

<code>data</code>	Data.frame.
<code>grid</code>	Grid of unique points at which to interpolate.
<code>idx_name</code>	Name of column containing a unique subject index.
<code>status_name</code>	Name of column containing the status. Must be coded as 0 for censoring, 1 for a measurement, 2 for death. Each subject should have an observation-terminating event, either censoring or death.
<code>rm_na</code>	Remove records interpolated to NA?
<code>time_name</code>	Name of column containing the observation time.
<code>value_name</code>	Name of the column containing the measurement.

Value

A data.frame containing subject index, status, time, and the interpolated value.

Notes

The grid of points will be augmented to include the time of each subjects observation terminating event.

Examples

```
example_data <- data.frame(
  idx = c(1, 1, 2, 2),
  status = c(1, 0, 1, 2),
  time = c(0, 1, 0, 1),
  value = c(0, 1, 0, -1)
)
Interpolate(example_data, grid = c(0, 0.5, 1))
```

InterpolateR	<i>Interpolation R Linearly interpolates between each subject's measurements. The input data should contain no missing values.</i>
--------------	--

Description

Interpolation R

Linearly interpolates between each subject's measurements. The input data should contain no missing values.

Usage

```
InterpolateR(grid, idx, status, time, value)
```

Arguments

grid	Grid of unique points at which to interpolate.
idx	Unique subject index.
status	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
time	Observation time.
value	Observation value.

Value

Data.frame.

KaplanMeierR	<i>Tabulate Kaplan Meier R</i>
--------------	--------------------------------

Description

Constructs a matrix with evaluation times as rows, and 4 columns:

- time Evaluation times.
- nar Number at risk.
- surv Left-limit survival probability, $\hat{S}(t-)$.
- haz Hazard.

Usage

KaplanMeierR(eval_times, idx, status, time)

Arguments

eval_times	Evaluation times.
idx	Unique subject index.
status	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
time	Observation time.

Value

Data.frame.

PerturbationR	<i>Perturbation R Generates realizations of $\frac{1}{n} \sum_{i=1}^n \psi_i w_i$, where ψ_i is the influence function for the ith subject and the w_i are IID random weights.</i>
---------------	---

Description

Perturbation R

Generates realizations of $\frac{1}{n} \sum_{i=1}^n \psi_i w_i$, where ψ_i is the influence function for the i th subject and the w_i are IID random weights.

Usage

```
PerturbationR(
  idx,
  perturbations,
  status,
  time,
  trunc_time,
  value,
  int_method = "trapezoid"
)
```

Arguments

<code>idx</code>	Unique subject index.
<code>perturbations</code>	Number of perturbations
<code>status</code>	Status, coded as 0 for censoring, 1 for a measurement, 2 for a terminal event.
<code>time</code>	Observation time.
<code>trunc_time</code>	Truncation time? Optional. If omitted, defaults to the maximum evaluation time.
<code>value</code>	Observation value.
<code>int_method</code>	Integration method, selected from "left", "right", "trapezoid".

Value

Numeric vector.

Notes

The random seed should be set in R prior to calling this function.

PlotTwoSample

Plot Two Sample

Description

Plot Two Sample

Usage

```
PlotTwoSample(
  data,
  arm_name = "arm",
  censor_after_last = TRUE,
  color_labs = c("Ctrl", "Trt"),
  colors = c("#EFC000FF", "#6385B8"),
```

```

eval_pts = 200,
idx_name = "idx",
show_auc = FALSE,
show_nar = TRUE,
status_name = "status",
tau = NULL,
time_name = "time",
title = NULL,
value_name = "value",
x_breaks = NULL,
x_labs = NULL,
x_name = "Time",
x_max = NULL,
y_name = "Expected Value",
y_lim = NULL
)

```

Arguments

<code>data</code>	Data.frame.
<code>arm_name</code>	Name of the column containing treatment arm. Coded as 0 for reference, 1 for treatment.
<code>censor_after_last</code>	Introduce censoring after the last event <i>if</i> no observation-terminating event is present.
<code>color_labs</code>	Color labels.
<code>colors</code>	Colors for control and treatment respectively.
<code>eval_pts</code>	Evaluation points between 0 and tau.
<code>idx_name</code>	Name of column containing a unique subject index.
<code>show_auc</code>	Show area under the curve?
<code>show_nar</code>	Show number at risk?
<code>status_name</code>	Name of column containing the status. Must be coded as 0 for censoring, 1 for a measurement, 2 for death. Each subject should have an observation-terminating event, either censoring or death.
<code>tau</code>	Truncation time.
<code>time_name</code>	Name of column containing the observation time.
<code>title</code>	Plot title.
<code>value_name</code>	Name of the column containing the measurement.
<code>x_breaks</code>	X-axis breaks.
<code>x_labs</code>	X-axis labels.
<code>x_name</code>	X-axis name.
<code>x_max</code>	X-axis upper limit.
<code>y_name</code>	Y-axis name.
<code>y_lim</code>	Y-axis limits.

Value

ggplot

print.AURMC	<i>Print Method for AURMC Object</i>
-------------	--------------------------------------

Description

Print method for objects of class AURMC.

Usage

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

Arguments

- x An object of class AURMC.
- ... Unused.

Value

Invisibly returns NULL. This method is called for its side effect of displaying the reference-arm, treatment-arm, and contrast tables stored in x, with noninteger numeric values rounded to three significant digits.

show,AURMC-method	<i>Show Method for AURMC Object</i>
-------------------	-------------------------------------

Description

Show Method for AURMC Object

Usage

```
## S4 method for signature 'AURMC'
show(object)
```

Arguments

- object An object of class AURMC.

TabRMC

Tabulate the Repeated Measures Curve

Description

Tabulate the Repeated Measures Curve

Usage

```
TabRMC(
  data,
  censor_after_last = TRUE,
  idx_name = "idx",
  status_name = "status",
  tau = NULL,
  time_name = "time",
  value_name = "value"
)
```

Arguments

<code>data</code>	Data.frame.
<code>censor_after_last</code>	Introduce censoring after the last event <i>if</i> no observation-terminating event is present.
<code>idx_name</code>	Name of column containing a unique subject index.
<code>status_name</code>	Name of column containing the status. Must be coded as 0 for censoring, 1 for a measurement, 2 for death. Each subject should have an observation-terminating event, either censoring or death.
<code>tau</code>	Truncation time.
<code>time_name</code>	Name of column containing the observation time.
<code>value_name</code>	Name of the column containing the measurement.

Value

A data.frame tabulating time, number and proportion at risk, terminal-event hazard, left-limit survival $\hat{S}(t-)$, observed-value mean, and the estimated repeated measures curve.

ValueMatrixR*Tabulate Value Matrix R*

Description

Tabulate $D_i(t)$ with subjects as rows and evaluation times as columns.

Usage

```
ValueMatrixR(eval_times, idx, time, value)
```

Arguments

eval_times	Evaluation times.
idx	Unique subject index.
time	Observation time.
value	Observation value.

Value

Numeric matrix.

Index

AtRiskMatrixR, [2](#)
AURMC, [3](#)
AURMC-class, [4](#)

BootstrapSamplesR, [4](#)

CalcMartingaleR, [5](#)
CalcMuR, [6](#)
CensorAfterLast, [6](#)
CompareAURMCs, [7](#)
Curve, [8](#)

DrawBootstrapR, [9](#)

EstimatorR, [9](#)

GenData, [10](#)

InfluenceR, [11](#)
Interpolate, [12](#)
InterpolateR, [13](#)

KaplanMeierR, [14](#)

PerturbationR, [14](#)
PlotTwoSample, [15](#)
print.AURMC, [17](#)

show, AURMC-method, [17](#)

TabRMC, [18](#)

ValueMatrixR, [19](#)