

# Package ‘CoxAalenCR’

August 21, 2026

**Type** Package

**Title** Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model  
for Competing Risks

**Version** 0.1.0

**Description** Implements the flexible additive-multiplicative Cox-Aalen subdistribution hazard regression model for competing risks data as proposed by Li and Long (2019) <[doi:10.1007/s11424-019-7281-6](https://doi.org/10.1007/s11424-019-7281-6)>. The framework accommodates both time-varying non-parametric additive covariate effects through an Aalen (1980) additive model and constant multiplicative effects via a Cox proportional hazards structure, generalizing Scheike and Zhang (2002) <[doi:10.1111/1467-9469.00065](https://doi.org/10.1111/1467-9469.00065)> and Martinussen and Scheike (2002) <[doi:10.1093/biomet/89.2.283](https://doi.org/10.1093/biomet/89.2.283)>. Includes inverse probability of censoring weighting (IPCW) with both Kaplan-Meier weights (Fine and Gray, 1999 <[doi:10.1080/01621459.1999.10474144](https://doi.org/10.1080/01621459.1999.10474144)>) and covariate-dependent Cox censoring weights (He et al., 2016 <[doi:10.1111/sjos.12172](https://doi.org/10.1111/sjos.12172)>; Li and Long, 2019 <[doi:10.1007/s11424-019-7281-6](https://doi.org/10.1007/s11424-019-7281-6)>). Provides simultaneous estimating equations based on Huffer and McKeague (1991) <[doi:10.1080/01621459.1991.10475010](https://doi.org/10.1080/01621459.1991.10475010)>, asymptotic sandwich variance estimation with censoring-weight martingale corrections, cumulative incidence function (CIF) prediction with pointwise confidence intervals, supremum-type goodness-of-fit tests for time-varying covariate effects, and Monte Carlo competing risks data simulation.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**Depends** R (>= 4.0.0)

**Imports** stats, graphics, grDevices, utils, survival

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Shikhar Tyagi [aut, cre] (ORCID:  
<https://orcid.org/0000-0003-1606-0844>),  
 Arvind Pandey [aut],  
 Bhupendra Singh [aut],  
 Vrijesh Tripathi [aut]

**Maintainer** Shikhar Tyagi <shikhar1093tyagi@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-08-21 15:20:02 UTC

## Contents

|                                       |    |
|---------------------------------------|----|
| AIC.cox_aalen_fit . . . . .           | 3  |
| cif_predict . . . . .                 | 3  |
| coef.cox_aalen_fit . . . . .          | 5  |
| confint.cox_aalen_fit . . . . .       | 5  |
| cox_aalen . . . . .                   | 6  |
| cox_aalen_fit . . . . .               | 7  |
| cox_weights . . . . .                 | 9  |
| km_weights . . . . .                  | 10 |
| logLik.cox_aalen_fit . . . . .        | 11 |
| nobs.cox_aalen_fit . . . . .          | 12 |
| plot.cox_aalen . . . . .              | 12 |
| plot.cox_aalen_fit . . . . .          | 13 |
| plot_cif . . . . .                    | 13 |
| plot_cumulative_coef . . . . .        | 14 |
| predict.cox_aalen . . . . .           | 15 |
| print.cox_aalen . . . . .             | 15 |
| print.cox_aalen_fit . . . . .         | 16 |
| print.simulation_study . . . . .      | 16 |
| print.summary.cox_aalen_fit . . . . . | 17 |
| print.time_varying_test . . . . .     | 17 |
| simulate_coxaalen . . . . .           | 18 |
| simulation_study . . . . .            | 19 |
| summary.cox_aalen . . . . .           | 20 |
| summary.cox_aalen_fit . . . . .       | 21 |
| tamoxifen . . . . .                   | 21 |
| time_varying_test . . . . .           | 22 |
| vcov.cox_aalen_fit . . . . .          | 23 |

## Index

24

---

|                   |                                                         |
|-------------------|---------------------------------------------------------|
| AIC.cox_aalen_fit | <i>Akaike Information Criterion for Cox-Aalen Model</i> |
|-------------------|---------------------------------------------------------|

---

**Description**

AIC placeholder for semiparametric estimating equations.

**Usage**

```
## S3 method for class 'cox_aalen_fit'
AIC(object, ..., k = 2)
```

**Arguments**

|        |                                            |
|--------|--------------------------------------------|
| object | an object of class "cox_aalen_fit".        |
| ...    | further arguments.                         |
| k      | numeric penalty per parameter (default 2). |

**Value**

Numeric NA.

---

|             |                                                                   |
|-------------|-------------------------------------------------------------------|
| cif_predict | <i>Predict Cumulative Incidence Functions for Cox-Aalen Model</i> |
|-------------|-------------------------------------------------------------------|

---

**Description**

Computes predicted cumulative incidence functions (CIF) for specified covariate profiles under the additive-multiplicative Cox-Aalen model with analytical pointwise standard errors and confidence intervals as derived in Theorem 4.3 of Li and Long (2019) <doi:10.1007/s11424-019-7281-6>.

**Usage**

```
cif_predict(
  object,
  newdata = NULL,
  newZ = NULL,
  newtime = NULL,
  se.fit = TRUE,
  conf.level = 0.95,
  ...
)
```

## Arguments

|                         |                                                                                                                                                                         |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code>     | an object of class "cox_aalen_fit" or "cox_aalen".                                                                                                                      |
| <code>newdata</code>    | matrix or data frame of new additive covariates $X$ . If NULL, uses the design matrix from the fitted object.                                                           |
| <code>newZ</code>       | matrix or data frame of new multiplicative covariates $Z$ . If NULL and <code>object\$n_mult &gt; 0</code> , uses the multiplicative covariates from the fitted object. |
| <code>newtime</code>    | numeric vector of time points for prediction. If NULL, predictions are evaluated at all event times <code>object\$A_times</code> .                                      |
| <code>se.fit</code>     | logical indicating whether standard errors and confidence intervals should be computed (TRUE or FALSE). Default is TRUE.                                                |
| <code>conf.level</code> | numeric confidence level for pointwise confidence intervals (default 0.95).                                                                                             |
| <code>...</code>        | additional arguments (currently unused).                                                                                                                                |

## Value

A list of class "cif\_predict" containing:

**cif** Numeric matrix of predicted CIF values (dimension  $n_{new} \times n_{times}$ ).

**se** Numeric matrix of pointwise standard errors (or NULL if `se.fit = FALSE`).

**lower** Numeric matrix of lower confidence bounds clamped to  $[0, 1]$ .

**upper** Numeric matrix of upper confidence bounds clamped to  $[0, 1]$ .

**time** Numeric vector of time points where predictions are evaluated.

**conf.level** Confidence level used.

## References

Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:[10.1007/s1142401972816](https://doi.org/10.1007/s1142401972816)

## Examples

```
set.seed(123)
dat <- simulate_coxaalen(n = 60, p = 0.3, censoring = "cox", cens_rate = 0.20)
fit <- cox_aalen_fit(
  time = dat$time,
  status = dat$status,
  X = cbind(Intercept = 1, X = dat$X),
  Z = matrix(dat$Z, ncol = 1, dimnames = list(NULL, "Z")),
  W = dat$W
)
pred <- cif_predict(fit, newdata = cbind(1, c(0.2, 0.8)), newZ = matrix(c(0.5, 0.5), ncol = 1))
print(pred$time[1:5])
print(pred$cif[, 1:5])
```

---

|                    |                                                  |
|--------------------|--------------------------------------------------|
| coef.cox_aalen_fit | <i>Extract Coefficients from Cox-Aalen Model</i> |
|--------------------|--------------------------------------------------|

---

**Description**

Extracts multiplicative regression coefficients  $\beta$ .

**Usage**

```
## S3 method for class 'cox_aalen_fit'  
coef(object, ...)
```

**Arguments**

|        |                                     |
|--------|-------------------------------------|
| object | an object of class "cox_aalen_fit". |
| ...    | further arguments.                  |

**Value**

Named numeric vector of multiplicative regression coefficients.

---

|                       |                                                                     |
|-----------------------|---------------------------------------------------------------------|
| confint.cox_aalen_fit | <i>Confidence Intervals for Cox-Aalen Multiplicative Parameters</i> |
|-----------------------|---------------------------------------------------------------------|

---

**Description**

Computes Wald confidence intervals for multiplicative parameters  $\beta$ .

**Usage**

```
## S3 method for class 'cox_aalen_fit'  
confint(object, parm = NULL, level = 0.95, ...)
```

**Arguments**

|        |                                                                           |
|--------|---------------------------------------------------------------------------|
| object | an object of class "cox_aalen_fit".                                       |
| parm   | a specification of which parameters are to be given confidence intervals. |
| level  | confidence level (default 0.95).                                          |
| ...    | further arguments.                                                        |

**Value**

A matrix with columns for the lower and upper confidence limits.

cox\_aalen

*Formula Interface for Additive-Multiplicative Cox-Aalen Model***Description**

Fits the additive-multiplicative Cox-Aalen subdistribution hazard model for competing risks data using a formula specification.

**Usage**

```
cox_aalen(
  formula,
  data,
  W = NULL,
  weight_type = c("cox", "km"),
  tau = NULL,
  control = list()
)
```

**Arguments**

|             |                                                                                                                                                                                                                                                                                                                                       |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| formula     | a two-part formula of the form <code>Surv(time, status) ~ X1 + X2   Z1 + Z2</code> where terms before <code> </code> represent additive (time-varying) covariates $X$ , and terms after <code> </code> represent multiplicative (constant) covariates $Z$ . If <code> </code> is omitted, all RHS covariates are treated as additive. |
| data        | a data frame containing the variables named in formula and W.                                                                                                                                                                                                                                                                         |
| W           | optional formula (e.g. <code>~ W1 + W2</code> ) or numeric matrix specifying covariates for the censoring distribution model.                                                                                                                                                                                                         |
| weight_type | character string specifying the IPCW weighting scheme: "cox" (default) or "km".                                                                                                                                                                                                                                                       |
| tau         | optional numeric value specifying the maximum follow-up time.                                                                                                                                                                                                                                                                         |
| control     | list of control parameters forwarded to <a href="#">cox_aalen_fit</a> .                                                                                                                                                                                                                                                               |

**Value**

An object of class `c("cox_aalen", "cox_aalen_fit")` containing the fitted model components, formula, and matched call. See [cox\\_aalen\\_fit](#) for full details.

**References**

Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:[10.1007/s1142401972816](https://doi.org/10.1007/s1142401972816)

## Examples

```
set.seed(123)
dat <- simulate_coxaalen(n = 60, p = 0.3, censoring = "cox", cens_rate = 0.20)
fit <- cox_aalen(
  formula = survival::Surv(time, status) ~ X | Z,
  data = dat,
  W = ~ W,
  weight_type = "cox"
)
print(fit)
summary(fit)
```

---

cox\_aalen\_fit

*Fit Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model*


---

## Description

Fits the flexible additive-multiplicative Cox-Aalen subdistribution hazard regression model for competing risks data via the estimating equations approach of Li and Long (2019) <doi:10.1007/s11424-019-7281-6>.

## Usage

```
cox_aalen_fit(
  time,
  status,
  X,
  Z = NULL,
  W = NULL,
  weight_type = c("cox", "km"),
  tau = NULL,
  control = list()
)
```

## Arguments

|        |                                                                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| time   | numeric vector of observed follow-up times $T_i = \min(\tilde{T}_i, C_i)$ . Missing values are not allowed.                                         |
| status | integer vector of event status indicators: 1 = event of interest (cause 1), 0 = censored, 2 = competing risk event. Missing values are not allowed. |
| X      | matrix or data frame of covariates for the additive (time-varying) component. Must have $n$ rows.                                                   |
| Z      | matrix or data frame of covariates for the multiplicative (constant) component. If NULL, a purely additive Aalen subdistribution model is fitted.   |
| W      | matrix or data frame of covariates for the censoring distribution model. Used when <code>weight_type = "cox"</code> .                               |

|             |                                                                                                                                                                                                                                  |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| weight_type | character string specifying the IPCW weighting scheme: "cox" for covariate-dependent Cox censoring weights (default), or "km" for univariate Kaplan-Meier weights.                                                               |
| tau         | numeric value specifying the maximum follow-up time. Default is the maximum observed time for cause 1 events.                                                                                                                    |
| control     | a list of control parameters for the Newton-Raphson solver:<br><b>max_iter</b> integer maximum number of iterations (default 50).<br><b>tol</b> numeric convergence tolerance on parameter change and score norm (default 1e-6). |

## Value

An object of class "cox\_aalen\_fit" containing:

**beta** Estimated regression coefficient vector for multiplicative covariates  $Z$ .  
**beta\_se** Standard error vector for beta.  
**beta\_names** Character vector of variable names for beta.  
**A** Matrix of cumulative additive coefficient estimates  $\hat{A}(t)$  across event times.  
**A\_times** Numeric vector of unique event times corresponding to rows of A.  
**A\_se** Matrix of pointwise standard errors for A.  
**vcov** Estimated asymptotic variance-covariance matrix for beta.  
**var\_A** 3D array of variance-covariance matrices for  $\hat{A}(t)$  across event times.  
**gamma** Estimated coefficient vector for the censoring model (if Cox weights used).  
**gamma\_se** Standard error vector for gamma.  
**weight\_type** Character string of weight type used.  
**psi1** Matrix of influence function values  $\hat{\psi}_{1i}$  for beta.  
**psi2** 3D array of influence function values  $\hat{\psi}_{2i}(t)$  for A.  
**n** Integer sample size.  
**n\_add** Integer number of additive covariates.  
**n\_mult** Integer number of multiplicative covariates.  
**converged** Logical indicating whether the optimization converged (TRUE or FALSE).  
**iterations** Integer number of iterations performed.  
**tau** Maximum follow-up time used.  
**call** Matched call.

## References

Fine, J. P. and Gray, R. J. (1999). A proportional hazards model for the subdistribution of a competing risk. *Journal of the American Statistical Association*, 94(446), 496-509. doi:10.1080/01621459.1999.10474144

He, P., Ewell, M. and Scheike, T. H. (2016). A proportional hazards regression model for the subdistribution with covariates-adjusted censoring weight for competing risks data. *Scandinavian Journal of Statistics*, 43(1), 103-122. doi:10.1111/sjos.12172

Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:[10.1007/s11424-019-7281-6](https://doi.org/10.1007/s11424-019-7281-6)

## Examples

```
set.seed(123)
dat <- simulate_coxaalen(n = 60, p = 0.3, censoring = "cox", cens_rate = 0.20)
fit <- cox_aalen_fit(
  time = dat$time,
  status = dat$status,
  X = cbind(Intercept = 1, X = dat$X),
  Z = matrix(dat$Z, ncol = 1, dimnames = list(NULL, "Z")),
  W = dat$W,
  weight_type = "cox"
)
print(fit)
```

---

cox\_weights

Covariate-Dependent Cox IPCW Weights for Competing Risks

---

## Description

Fits a Cox proportional hazards model for the censoring distribution conditional on censoring covariates  $W$  and computes covariate-adjusted IPCW weights as proposed by He et al. (2016) <doi:10.1111/sjos.12172> and Li and Long (2019) <doi:10.1007/s11424-019-7281-6>.

## Usage

```
cox_weights(time, status, W = NULL)
```

## Arguments

|        |                                                                                                                                                                        |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| time   | numeric vector of observed follow-up times $T_i = \min(\tilde{T}_i, C_i)$ . Missing values are not permitted.                                                          |
| status | integer vector of event status indicators: 1 = event of interest, 0 = censored, 2 = competing risk event. Missing values are not permitted.                            |
| W      | matrix or data frame of covariates associated with the censoring distribution. Must have $n$ rows matching time. If NULL or empty, falls back to Kaplan-Meier weights. |

## Value

A list of class "cox\_weights" containing:

**gamma** Estimated regression coefficient vector for the censoring Cox model.

**gamma\_se** Standard error vector for gamma.

**cox\_fit** The fitted `coxph` model object for censoring.

**baseline\_haz** Data frame containing the Breslow cumulative baseline hazard estimate with columns `time` and `hazard`.

**weights\_obs** Numeric vector of censoring survival probabilities  $\hat{G}(T_i; W_i)$  evaluated at each subject's observed time.

**n** Integer sample size.

**n\_covariates** Integer number of censoring covariates.

## References

He, P., Ewell, M. and Scheike, T. H. (2016). A proportional hazards regression model for the subdistribution with covariates-adjusted censoring weight for competing risks data. *Scandinavian Journal of Statistics*, 43(1), 103-122. doi:[10.1111/sjos.12172](https://doi.org/10.1111/sjos.12172)

Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:[10.1007/s1142401972816](https://doi.org/10.1007/s1142401972816)

## Examples

```
set.seed(123)
dat <- simulate_coxaalen(n = 60, p = 0.3, censoring = "cox", cens_rate = 0.20)
cw_res <- cox_weights(time = dat$time, status = dat$status, W = dat$W)
print(cw_res$gamma)
```

---

km\_weights

Kaplan-Meier IPCW Weights for Competing Risks

---

## Description

Computes inverse probability of censoring weighting (IPCW) using the univariate Kaplan-Meier estimator for the censoring distribution as proposed by Fine and Gray (1999) <doi:10.1080/01621459.1999.10474144>.

## Usage

```
km_weights(time, status)
```

## Arguments

|                     |                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>time</code>   | numeric vector of observed follow-up times $T_i = \min(\tilde{T}_i, C_i)$ . Missing values (NA, NaN) are not permitted.                               |
| <code>status</code> | integer vector of event status indicators: 1 = event of interest (cause 1), 0 = censored, 2 = competing risk event. Missing values are not permitted. |

**Value**

A list of class "km\_weights" containing:

**surv\_fit** The fitted `survfit` object for censoring.

**weights\_obs** Numeric vector of censoring survival probabilities  $\hat{G}(T_i)$  evaluated at each subject's observed time.

**n** Integer sample size.

**n\_events** Integer count of events of interest.

**n\_censored** Integer count of censored observations.

**n\_competing** Integer count of competing risk events.

**References**

Fine, J. P. and Gray, R. J. (1999). A proportional hazards model for the subdistribution of a competing risk. *Journal of the American Statistical Association*, 94(446), 496-509. doi:10.1080/01621459.1999.10474144

**Examples**

```
set.seed(123)
dat <- simulate_coxaalen(n = 50, p = 0.3, censoring = "independent", cens_rate = 0.20)
km_res <- km_weights(time = dat$time, status = dat$status)
print(head(km_res$weights_obs))
```

---

logLik.cox\_aalen\_fit    *Log-Likelihood Method for Cox-Aalen Model*

---

**Description**

Pseudo log-likelihood for estimating equation models.

**Usage**

```
## S3 method for class 'cox_aalen_fit'
logLik(object, ...)
```

**Arguments**

**object**                    an object of class "cox\_aalen\_fit".  
**...**                      further arguments.

**Value**

An object of class "logLik" (evaluates to NA with df and nobs attributes).

---

|                    |                                                  |
|--------------------|--------------------------------------------------|
| nobs.cox_aalen_fit | <i>Number of Observations in Cox-Aalen Model</i> |
|--------------------|--------------------------------------------------|

---

**Description**

Returns the number of observations used in model fitting.

**Usage**

```
## S3 method for class 'cox_aalen_fit'
nobs(object, ...)
```

**Arguments**

|        |                                     |
|--------|-------------------------------------|
| object | an object of class "cox_aalen_fit". |
| ...    | further arguments.                  |

**Value**

An integer representing the number of observations.

---

|                |                                          |
|----------------|------------------------------------------|
| plot.cox_aalen | <i>Plot Method for Cox-Aalen Objects</i> |
|----------------|------------------------------------------|

---

**Description**

Plots predicted cumulative incidence functions or cumulative additive coefficient paths.

**Usage**

```
## S3 method for class 'cox_aalen'
plot(x, which = c("cif", "cumulative_coef"), ...)
```

**Arguments**

|       |                                                                                                |
|-------|------------------------------------------------------------------------------------------------|
| x     | an object of class "cox_aalen" or "cox_aalen_fit".                                             |
| which | character string specifying which plot to produce: "cif" or "cumulative_coef".                 |
| ...   | further arguments passed to <a href="#">plot_cif</a> or <a href="#">plot_cumulative_coef</a> . |

**Value**

No return value, called for side effects.

---

|                    |                                              |
|--------------------|----------------------------------------------|
| plot.cox_aalen_fit | <i>Plot Method for Cox-Aalen Fit Objects</i> |
|--------------------|----------------------------------------------|

---

**Description**

S3 plot method for objects of class "cox\_aalen\_fit".

**Usage**

```
## S3 method for class 'cox_aalen_fit'
plot(x, which = c("cif", "cumulative_coef"), ...)
```

**Arguments**

|       |                                                                                |
|-------|--------------------------------------------------------------------------------|
| x     | an object of class "cox_aalen_fit".                                            |
| which | character string specifying which plot to produce: "cif" or "cumulative_coef". |
| ...   | further arguments passed to plotting routines.                                 |

**Value**

No return value, called for side effects.

---

|          |                                                      |
|----------|------------------------------------------------------|
| plot_cif | <i>Plot Predicted Cumulative Incidence Functions</i> |
|----------|------------------------------------------------------|

---

**Description**

Displays predicted cumulative incidence functions (CIF) with optional confidence bands.

**Usage**

```
plot_cif(
  fit,
  newdata = NULL,
  newZ = NULL,
  conf_level = 0.95,
  xlab = "Time",
  ylab = "Cumulative Incidence",
  main = "Predicted Cumulative Incidence Function",
  col = NULL,
  ...
)
```

**Arguments**

|            |                                                                                   |
|------------|-----------------------------------------------------------------------------------|
| fit        | a fitted object of class "cox_aalen_fit" or "cox_aalen".                          |
| newdata    | optional matrix or data frame of new additive covariates $X$ .                    |
| newZ       | optional matrix or data frame of new multiplicative covariates $Z$ .              |
| conf_level | numeric confidence level for confidence bands (default 0.95).                     |
| xlab       | character label for the x-axis (default "Time").                                  |
| ylab       | character label for the y-axis (default "Cumulative Incidence").                  |
| main       | character title for the plot (default "Predicted Cumulative Incidence Function"). |
| col        | optional vector of colors for the predicted curves.                               |
| ...        | additional graphical parameters passed to <a href="#">plot</a> .                  |

**Value**

No return value, called for side effects.

---

plot\_cumulative\_coef    *Plot Cumulative Additive Regression Functions*

---

**Description**

Plots the estimated non-parametric cumulative additive coefficient curves  $\hat{A}_j(t)$  over time along with 95% pointwise confidence bands.

**Usage**

```
plot_cumulative_coef(
  fit,
  conf_level = 0.95,
  xlab = "Time",
  ylab = "Cumulative Coefficient",
  main = NULL,
  ...
)
```

**Arguments**

|            |                                                                    |
|------------|--------------------------------------------------------------------|
| fit        | a fitted object of class "cox_aalen_fit" or "cox_aalen".           |
| conf_level | numeric confidence level for confidence bands (default 0.95).      |
| xlab       | character label for the x-axis (default "Time").                   |
| ylab       | character label for the y-axis (default "Cumulative Coefficient"). |
| main       | optional main title prefix.                                        |
| ...        | additional graphical parameters.                                   |

**Value**

No return value, called for side effects.

---

|                   |                                             |
|-------------------|---------------------------------------------|
| predict.cox_aalen | <i>Predict Method for Cox-Aalen Objects</i> |
|-------------------|---------------------------------------------|

---

**Description**

Predicts cumulative incidence functions using fitted model.

**Usage**

```
## S3 method for class 'cox_aalen'
predict(
  object,
  newdata = NULL,
  newtime = NULL,
  se.fit = TRUE,
  conf.level = 0.95,
  ...
)
```

**Arguments**

|            |                                                                |
|------------|----------------------------------------------------------------|
| object     | an object of class "cox_aalen".                                |
| newdata    | optional data frame of new observation profiles.               |
| newtime    | optional numeric vector of prediction times.                   |
| se.fit     | logical indicating whether standard errors should be computed. |
| conf.level | numeric confidence level (default 0.95).                       |
| ...        | further arguments.                                             |

**Value**

An object of class "cif\_predict".

---

|                 |                                                  |
|-----------------|--------------------------------------------------|
| print.cox_aalen | <i>Print Method for Cox-Aalen Formula Object</i> |
|-----------------|--------------------------------------------------|

---

**Description**

Prints the matched call and model summary.

**Usage**

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

**Arguments**

x                    an object of class "cox\_aalen".  
...                  further arguments passed to or from other methods.

**Value**

No return value, called for side effects.

---

print.cox\_aalen\_fit    *Print Method for Cox-Aalen Fitted Models*

---

**Description**

Prints a concise summary of the fitted additive-multiplicative Cox-Aalen model.

**Usage**

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

**Arguments**

x                    an object of class "cox\_aalen\_fit" or "cox\_aalen".  
...                  further arguments passed to or from other methods.

**Value**

No return value, called for side effects.

---

print.simulation\_study                    *Print Method for Simulation Study Results*

---

**Description**

Prints the formatted simulation study performance metrics.

**Usage**

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

**Arguments**

x                    an object of class "simulation\_study".  
...                  further arguments.

**Value**

No return value, called for side effects.

---

```
print.summary.cox_aalen_fit
```

*Print Method for Summary of Cox-Aalen Model*

---

**Description**

Prints the formatted summary table.

**Usage**

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

**Arguments**

|     |                                                    |
|-----|----------------------------------------------------|
| x   | an object of class "summary.cox_aalen_fit".        |
| ... | further arguments passed to or from other methods. |

**Value**

No return value, called for side effects.

---

```
print.time_varying_test
```

*Print Method for Goodness-of-Fit Test*

---

**Description**

Prints the summary of time-varying effect goodness-of-fit tests.

**Usage**

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

**Arguments**

|     |                                         |
|-----|-----------------------------------------|
| x   | an object of class "time_varying_test". |
| ... | further arguments.                      |

**Value**

No return value, called for side effects.

---

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| simulate_coxaalen | <i>Simulate Competing Risks Data under Cox-Aalen Model</i> |
|-------------------|------------------------------------------------------------|

---

## Description

Generates competing risks survival data exactly following the simulation design in Section 5 of Li and Long (2019) <doi:10.1007/s11424-019-7281-6>.

## Usage

```
simulate_coxaalen(
  n,
  p = 0.3,
  alpha = 1,
  beta1 = 1,
  beta2 = 1,
  censoring = c("independent", "cox", "threshold_uniform"),
  cens_rate = c(0.2, 0.4)
)
```

## Arguments

|           |                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| n         | integer sample size.                                                                                                                                                                                        |
| p         | numeric baseline subdistribution probability $p = F_1(\infty; 0, 0) \in (0, 1)$ . Default is 0.3.                                                                                                           |
| alpha     | numeric parameter $\alpha$ for time-varying effect $\alpha_2(t) = \alpha e^{-t}$ . Default is 1.0.                                                                                                          |
| beta1     | numeric true coefficient $\beta_1$ for the event of interest. Default is 1.0.                                                                                                                               |
| beta2     | numeric true coefficient $\beta_2$ for the competing risk event. Default is 1.0.                                                                                                                            |
| censoring | character string specifying the censoring scenario: "independent" ( $C \sim U(0, a)$ ), "cox" ( $\lambda_C(t; W) = \lambda_C e^{\gamma W}$ ), or "threshold_uniform" (piecewise uniform depending on $W$ ). |
| cens_rate | numeric target censoring rate: 0.20 (20%) or 0.40 (40%).                                                                                                                                                    |

## Value

A data frame with  $n$  rows and columns:

**time** Observed follow-up time  $T_i = \min(\tilde{T}_i, C_i)$ .

**status** Event status indicator: 1 = cause 1 (event of interest), 2 = cause 2 (competing risk), 0 = censored.

**X** Additive covariate generated from  $U(0, 1)$ .

**Z** Multiplicative covariate generated from  $U(0, 1)$ .

**W** Censoring-dependent covariate (equals  $Z$ ).

## References

Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:10.1007/s1142401972816

## Examples

```
set.seed(123)
dat <- simulate_coxaalen(n = 100, p = 0.3, censoring = "cox", cens_rate = 0.20)
head(dat)
table(dat$status)
```

---

simulation\_study

---

Monte Carlo Simulation Study for Cox-Aalen Model

---

## Description

Conducts a Monte Carlo simulation study to evaluate the empirical bias, standard errors, and 95% coverage probability of the Cox-Aalen model estimators under various censoring scenarios.

## Usage

```
simulation_study(
  n_sim = 50L,
  n = 100L,
  p = 0.3,
  alpha = 1,
  beta1 = 1,
  beta2 = 1,
  censoring = c("independent", "cox", "threshold_uniform"),
  cens_rate = 0.2,
  weight_type = c("cox", "km")
)
```

## Arguments

|             |                                                                 |
|-------------|-----------------------------------------------------------------|
| n_sim       | integer number of simulation replications (default 50).         |
| n           | integer sample size per replication (default 100).              |
| p           | numeric baseline subdistribution parameter (default 0.3).       |
| alpha       | numeric time-varying parameter (default 1.0).                   |
| beta1       | numeric true multiplicative coefficient (default 1.0).          |
| beta2       | numeric true competing risk coefficient (default 1.0).          |
| censoring   | character string: "independent", "cox", or "threshold_uniform". |
| cens_rate   | numeric target censoring rate (0.20 or 0.40).                   |
| weight_type | character string: "cox" or "km".                                |

**Value**

An object of class "simulation\_study" containing:

**results** Data frame with columns Parameter, True, Bias, Empirical\_SE, Average\_SE, and Coverage\_95.

**settings** List of simulation configuration parameters.

**References**

Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:[10.1007/s1142401972816](https://doi.org/10.1007/s1142401972816)

**Examples**

```
set.seed(123)
sim_res <- simulation_study(n_sim = 5, n = 40, p = 0.3, censoring = "independent")
print(sim_res)
```

---

|                   |                                                    |
|-------------------|----------------------------------------------------|
| summary.cox_aalen | <i>Summary Method for Cox-Aalen Formula Object</i> |
|-------------------|----------------------------------------------------|

---

**Description**

Summary method for formula-fitted Cox-Aalen model.

**Usage**

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

**Arguments**

|        |                                                    |
|--------|----------------------------------------------------|
| object | an object of class "cox_aalen".                    |
| ...    | further arguments passed to or from other methods. |

**Value**

An object of class "summary.cox\_aalen".

---

summary.cox\_aalen\_fit *Summary Method for Cox-Aalen Fitted Models*


---

**Description**

Produces a detailed summary of the fitted model parameters, standard errors, z-values, p-values, and cumulative additive functions.

**Usage**

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

**Arguments**

**object** an object of class "cox\_aalen\_fit" or "cox\_aalen".  
**...** further arguments passed to or from other methods.

**Value**

An object of class "summary.cox\_aalen\_fit" containing summary tables and fit statistics.

---

tamoxifen *Tamoxifen Breast Cancer Competing Risks Clinical Trial Dataset*


---

**Description**

Clinical trial dataset of 641 women aged 50 years or older with early breast cancer treated at Princess Margaret Hospital between 1992 and 2000, analyzing the effect of breast irradiation plus tamoxifen vs. tamoxifen alone under competing risks as analyzed in Li and Long (2019) <doi:10.1007/s11424-019-7281-6>, Fyles et al. (2004) <doi:10.1056/NEJMoa040595>, and Melania Pintilie (2006, ISBN:978-0-470-87068-6).

**Usage**

```
tamoxifen
```

**Format**

A data frame with 641 rows and 5 variables:

**time** Follow-up time in years.

**status** Event status: 1 = second malignancy (event of interest), 2 = death without second malignancy (competing risk), 0 = censored.

**age** Patient age in years at study entry (mean 66.45, standard deviation 8.91).

**pathsize** Pathological tumor size in centimeters (mean 1.55, standard deviation 0.77).

**treatment** Treatment indicator: 1 = radiation plus tamoxifen, 0 = tamoxifen alone.

## Source

Princess Margaret Hospital breast cancer study, published in Fyles et al. (2004) and analyzed in Li and Long (2019).

## References

- Fyles, A. W., McCready, D. R., Manchul, L. A., et al. (2004). Tamoxifen with or without breast irradiation in women 50 years of age or older with early breast cancer. *New England Journal of Medicine*, 351(10), 963-970. doi:10.1056/NEJMoa040595
- Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:10.1007/s1142401972816
- Pintilie, M. (2006). *Competing Risks: A Practical Perspective*. John Wiley & Sons, Chichester. ISBN:978-0-470-87068-6.

## Examples

```
data(tamoxifen)
head(tamoxifen)
table(tamoxifen$status)
```

---

|                   |                                                                         |
|-------------------|-------------------------------------------------------------------------|
| time_varying_test | <i>Goodness-of-Fit Test for Time-Varying Effects in Cox-Aalen Model</i> |
|-------------------|-------------------------------------------------------------------------|

---

## Description

Performs supremum-type Kolmogorov-Smirnov and Cramer-von Mises goodness-of-fit tests to evaluate whether additive covariates have time-varying or constant effects, implementing Remark 4.4 of Li and Long (2019) <doi:10.1007/s11424-019-7281-6> using Gaussian multiplier resampling (Lin et al., 1993 <doi:10.1093/biomet/80.3.557>; Scheike and Zhang, 2002 <doi:10.1111/1467-9469.00065>).

## Usage

```
time_varying_test(fit, B = 500L)
```

## Arguments

|     |                                                                    |
|-----|--------------------------------------------------------------------|
| fit | a fitted model object of class "cox_aalen_fit" or "cox_aalen".     |
| B   | integer number of Monte Carlo resampling iterations (default 500). |

## Value

A list of class "time\_varying\_test" containing:

**test\_results** A data frame with columns Covariate, Sup\_Stat (supremum statistic), p\_value\_sup (resampling p-value for supremum test), CVM\_Stat (Cramer-von Mises statistic), and p\_value\_cvm.

**B** Integer number of resampling iterations used.

**n\_add** Number of additive covariates tested.

## References

- Li, W. and Long, Y. (2019). An Additive-Multiplicative Cox-Aalen Subdistribution Hazard Model for Competing Risks Data. *Journal of Systems Science and Complexity*, 32(6), 1727-1746. doi:[10.1007/s1142401972816](https://doi.org/10.1007/s1142401972816)
- Lin, D. Y., Wei, L. J., Yang, I. and Ying, Z. (1993). Checking the Cox model with cumulative sums of martingale-based residuals. *Biometrika*, 80(3), 557-572. doi:[10.1093/biomet/80.3.557](https://doi.org/10.1093/biomet/80.3.557)
- Scheike, T. H. and Zhang, M. J. (2002). An additive-multiplicative Cox-Aalen regression model. *Scandinavian Journal of Statistics*, 29(1), 75-88. doi:[10.1111/14679469.00065](https://doi.org/10.1111/14679469.00065)

## Examples

```
set.seed(123)
dat <- simulate_coxaalen(n = 50, p = 0.3, censoring = "independent", cens_rate = 0.20)
fit <- cox_aalen_fit(
  time = dat$time,
  status = dat$status,
  X = cbind(Intercept = 1, X = dat$X),
  Z = matrix(dat$Z, ncol = 1, dimnames = list(NULL, "Z"))
)
test_res <- time_varying_test(fit, B = 100)
print(test_res)
```

---

|                    |                                                |
|--------------------|------------------------------------------------|
| vcov.cox_aalen_fit | Variance-Covariance Matrix for Cox-Aalen Model |
|--------------------|------------------------------------------------|

---

## Description

Extracts the asymptotic variance-covariance matrix of  $\beta$ .

## Usage

```
## S3 method for class 'cox_aalen_fit'
vcov(object, ...)
```

## Arguments

|        |                                     |
|--------|-------------------------------------|
| object | an object of class "cox_aalen_fit". |
| ...    | further arguments.                  |

## Value

Matrix of variance-covariance estimates for multiplicative parameters.

# Index

## \* datasets

tamoxifen, [21](#)

AIC.cox\_aalen\_fit, [3](#)

cif\_predict, [3](#)

coef.cox\_aalen\_fit, [5](#)

confint.cox\_aalen\_fit, [5](#)

cox\_aalen, [6](#)

cox\_aalen\_fit, [6](#), [7](#)

cox\_weights, [9](#)

coxph, [10](#)

km\_weights, [10](#)

logLik.cox\_aalen\_fit, [11](#)

nobs.cox\_aalen\_fit, [12](#)

plot, [14](#)

plot.cox\_aalen, [12](#)

plot.cox\_aalen\_fit, [13](#)

plot\_cif, [12](#), [13](#)

plot\_cumulative\_coef, [12](#), [14](#)

predict.cox\_aalen, [15](#)

print.cox\_aalen, [15](#)

print.cox\_aalen\_fit, [16](#)

print.simulation\_study, [16](#)

print.summary.cox\_aalen\_fit, [17](#)

print.time\_varying\_test, [17](#)

simulate\_coxaalen, [18](#)

simulation\_study, [19](#)

summary.cox\_aalen, [20](#)

summary.cox\_aalen\_fit, [21](#)

survfit, [11](#)

tamoxifen, [21](#)

time\_varying\_test, [22](#)

vcov.cox\_aalen\_fit, [23](#)