

# Package ‘ICcalib’

July 21, 2025

**Type** Package

**Title** Cox Model with Interval-Censored Starting Time of a Covariate

**Version** 1.0.8

**Date** 2018-07-23

**Author** Daniel Nevo

**Maintainer** Daniel Nevo <danielnevo@gmail.com>

**Description** Calibration and risk-set calibration methods for fitting Cox proportional hazard model when a binary covariate is measured intermittently. Methods include functions to fit calibration models from interval-censored data and modified partial likelihood for the proportional hazard model, Nevo et al. (2018+) <[doi:10.48550/arXiv.1801.01529](https://doi.org/10.48550/arXiv.1801.01529)>.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.12.5), survival, fitdistrplus, icenReg, numDeriv, ICsurv, msm, MASS

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-08-01 04:30:03 UTC

## Contents

CalcCoxCalibP . . . . .	2
CalcCoxCalibRSIntsP . . . . .	3
CalcNpmleCalibP . . . . .	4
CalcNpmleRSP . . . . .	5
CalcVarParam . . . . .	6
CalcWeibullCalibP . . . . .	8
CalcWeibullRSP . . . . .	9
FitCalibCox . . . . .	10
FitCalibCoxRS . . . . .	11
FitCalibNpmle . . . . .	13
FitCalibWeibull . . . . .	14
FitCalibWeibullRS . . . . .	14

**Index****16**


---

CalcCoxCalibP	<i>Calculating the probabilities of positive binary exposure status at a given time point using a proportional hazards calibration model</i>
---------------	--

---

**Description**

For a given time point, calculate the probability of positive exposure value for multiple observations (participants). The function uses the results of a proportional hazards calibration model fit, and given covariates and collected data on the history of the binary exposure for each participant.

**Usage**

```
CalcCoxCalibP(w, w.res, point, fit.cox, hz.times, Q)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in w
point	The time point at which the probabilities are estimated
fit.cox	The result of <code>icenReg::ic_sp</code> on the interval-censored data
hz.times	Times used for calculating the baseline hazard function from PH calibration model
Q	Matrix of covariates for the PH calibration model

**Value**

A vector of estimated probabilities of positive exposure status at time point.

**Examples**

```
sim.data <- ICcalib::SimCoxIntervalCensCox(n.sample = 200, lambda = 0.1,
                                           alpha = 0.25, beta0 = 0,
                                           gamma.q = c(log(0.75), log(2.5)),
                                           gamma.z = log(1.5), mu = 0.2,
                                           n.points = 2)

# The baseline hazard for the calibration model is calculated in observation times
cox.hz.times <- sort(unique(sim.data$obs.tm))
# Fit proportional hazards calibration model
fit.cox <- FitCalibCox(w = sim.data$w, w.res = sim.data$w.res, Q = sim.data$Q,
                      hz.times = cox.hz.times, n.int = 5, order = 2)
# Calculate the conditional probabilities of binary covariate=1 at time one
probs <- CalcCoxCalibP(w = sim.data$w, w.res = sim.data$w.res, point = 1,
                      Q = sim.data$Q, fit.cox = fit.cox, hz.times = cox.hz.times)

summary(probs)
```

---

CalcCoxCalibRSIntsP	<i>Calculating the probabilities of positive binary exposure status at a given time point using proportional hazards grouped risk-set calibration models</i>
---------------------	--

---

## Description

For a given time point, calculate the probability of positive exposure value for multiple observations (participants). The function uses the results of proportional hazards grouped risk-set calibration model fit, and given covariates and collected data on the history of the binary exposure for each participant.

## Usage

```
CalcCoxCalibRSIntsP(w, w.res, point, fit.cox.rs.ints, hz.times, Q, pts.for.ints)
```

## Arguments

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in w
point	The time point at which the probabilities are estimated
fit.cox.rs.ints	The result of FitCalibCoxRSInts on the interval-censored data
hz.times	Times used for calculating the baseline hazard function from PH calibration model
Q	Matrix of covariates for the PH calibration model
pts.for.ints	Points defining the intervals for grouping risk-sets (first one has to be zero). Should be sorted from zero up

## Value

A vector of estimated probabilities of positive exposure status at time point.

## Examples

```
set.seed(17)
sim.data <- ICcalib::SimCoxIntervalCensCox(n.sample = 100, lambda = 0.1,
                                           alpha = 0.25, beta0 = 0,
                                           gamma.q = c(log(0.75), log(2.5)),
                                           gamma.z = log(1.5), mu = 0.2,
                                           n.points = 2)
# The baseline hazard for the calibration model is calculated in observation times
cox.hz.times <- sort(unique(sim.data$obs.tm))
# Fit proportional hazards calibration model
```

```

fit.cox.rs.ints <- FitCalibCoxRSInts(w = sim.data$w, w.res = sim.data$w.res,
                                   Q = sim.data$Q, hz.times = cox.hz.times,
                                   n.int = 5, order = 2, pts.for.ints = seq(0,4,1),
                                   tm = sim.data$obs.tm, event = sim.data$delta)
# Calculate the conditional probabilities of binary covariate=1 at time one
probs <- CalcCoxCalibRSIntsP(w = sim.data$w, w.res = sim.data$w.res, point = 1,
                             fit.cox.rs.ints = fit.cox.rs.ints,
                             pts.for.ints = seq(0,4,1), Q = sim.data$Q,
                             hz.times = cox.hz.times)

summary(probs)

```

---

CalcNpmleCalibP

*Calculating the probabilities of positive binary exposure status at a given time point using a nonparametric calibration model*


---

### Description

For a given time point, calculate the probability of positive exposure value for multiple observations (participants). The function uses the results of a nonparametric calibration model fit, and given collected data on the history of the binary exposure for each participant.

### Usage

```
CalcNpmleCalibP(w, w.res, point, fit.npmle)
```

### Arguments

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in w
point	The time point at which the probabilities are estimated.
fit.npmle	The result of <code>icenReg::ic_np</code> on the interval-censored data

### Value

A vector of estimated probabilities of positive exposure status at time point.

### Examples

```

sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                             alpha = 0.25, beta0 = log(0.5),
                                             mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)

# Fit nonparametric calibration model
fit.npmle <- FitCalibNpmle(w = sim.data$w, w.res = sim.data$w.res)
# Calculate the conditional probabilities of binary covariate=1 at time one
probs <- CalcNpmleCalibP(w = sim.data$w, w.res = sim.data$w.res,
                         point = 1, fit.npmle = fit.npmle)

summary(probs)

```

---

CalcNpmleRSP	<i>Calculating the probabilities of positive binary exposure status at a given time point using a nonparametric risk-set calibration models</i>
--------------	---

---

## Description

For a given time point, calculate the probability of positive exposure value for multiple observations (participants). The function first fits the nonparametric risk-set calibration models at each main event time point and then calculates the probabilities of positive binary exposure status.

## Usage

```
CalcNpmleRSP(w, w.res, point, obs.tm)
```

## Arguments

<code>w</code>	A matrix of time points when measurements on the binary covariate were obtained.
<code>w.res</code>	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in <code>w</code>
<code>point</code>	The time point at which the probabilities are estimated
<code>obs.tm</code>	Vector of observed main event time or censoring time

## Details

This function calculates the NPMLE at each main event time point and then provides the estimated probabilities for positive exposure status at time point.

## Value

A vector of estimated probabilities of positive exposure status at time point.

## See Also

[ic\\_np](#)

## Examples

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                             alpha = 0.25, beta0 = log(0.5),
                                             mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)

# Calculate the conditional probabilities of binary covariate=1 at time one
# Unlike CalcNpmle, CalcNpmleRSP includes the calibration model fitting
probs <- CalcNpmleRSP(w = sim.data$w, w.res = sim.data$w.res, point = 1,
                     obs.tm = sim.data$obs.tm)

summary(probs)
```

CalcVarParam

*Variance estimation for the main proportional hazards model***Description**

Estimation of the covariance matrix for the parameters of the main proportional hazards model. This includes the variance of the binary exposure estimate and the other covariates, if included in the model. Each function correspond to a different calibration (or risk-set calibration model).

For nonparametric calibration, bootstrap calculations of the variance and confidence interval for the for the log hazard-ratio of the binary exposure.

**Usage**

```
CalcVarParam(theta, tm, event, Z, Q, ps, ps.deriv, w, w.res, fit.cox)
```

```
CalcVarParamRSInts(theta, tm, event, Z, Q, ps, ps.deriv, w, w.res,  
  fit.cox.rs.ints, pts.for.ints, n.etas.per.fit)
```

```
CalcVarThetaWeib(beta, etas, tm, event, ps, ps.deriv.shape, ps.deriv.scale, w,  
  w.res)
```

```
CalcVarThetaWeibRS(beta, etas.matrix, tm, event, ps.rs, ps.deriv.shape.rs,  
  ps.deriv.scale.rs, w, w.res)
```

```
CalcVarNpmle(tm, event, w, w.res, BS = 100, CI = T)
```

```
CalcVarNpmleRS(tm, event, w, w.res, BS = 100, CI = T)
```

**Arguments**

theta	Coefficient vector from main PH model. First coefficient corresponds to X, the rest to Z
tm	Vector of observed main event time or censoring time
event	Vector of censoring indicators. 1 for event 0 for censored
Z	Additional variables for the main model other than the binary covariate
Q	For PH calibration models: additional covariates
ps	A matrix. Rows are observations, columns are time points of the events. The entry at the i-th row and j-column is the conditional probability of positive exposure status for observation i at the j-th event time.
ps.deriv	A matrix. Rows are observations, columns are time points of the events. The derivative of ps with respect to the calibration model parameters
w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. The measurement corresponding to the time points in w.

<code>fit.cox</code>	For PH calibration models: The result of <code>icenReg::ic_sp</code> on the interval-censored data
<code>fit.cox.rs.ints</code>	For grouped risk-set PH calibration: The result of <code>FitCalibCoxRSInts</code> on the interval-censored data
<code>pts.for.ints</code>	For grouped-risk set PH calibration: Points defining the intervals for grouping risk-sets (first one has to be zero). Should be sorted from zero up
<code>n.etas.per.fit</code>	For grouped-risk set PH calibration: A vector. Total number of parameters for each PH calibration fit.
<code>beta</code>	Coefficient of the binary covariate. The analogue of $\theta$ for non-PH calibration models
<code>etas</code>	For Weibull calibration: Shape and scale parameters of the Weibull calibration model.
<code>ps.deriv.shape</code>	The derivative of <code>ps</code> with respect to the shape parameter of the Weibull calibration model.
<code>ps.deriv.scale</code>	The derivative of <code>ps</code> with respect to the scale parameter of the Weibull calibration model.
<code>etas.matrix</code>	For Weibull risk-set calibration: Two-columns matrix. Each row contains shape and scale parameters from a Weibull risk-set calibration model
<code>ps.rs</code>	A matrix. Rows are observations, columns are time points of the events. The entry at the $i$ -th row and $j$ -column is the conditional probability of positive exposure status for observation $i$ at the $j$ -th event time.
<code>ps.deriv.shape.rs</code>	For Weibull risk-set calibration: The derivative of <code>ps</code> with respect to the shape parameters of the Weibull risk-set calibration models.
<code>ps.deriv.scale.rs</code>	For Weibull risk-set calibration: The derivative of <code>ps</code> with respect to the scale parameters of the Weibull risk-set calibration models.
<code>BS</code>	For nonparametric calibration: Number of bootstrap iterations, Default: 100
<code>CI</code>	For nonparametric calibration: Should the function return confidence intervals?, Default: T

## Value

The covariance matrix. The first row and column are for the binary exposure.

For nonparametric calibration: Variance estimate and possibly confidence interval for the log hazard-ratio of the binary exposure under a nonparametric calibration model.

## Examples

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1, alpha = 0.25,
                                             beta0 = log(0.5), mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)
case.times <- sim.data$obs.tm[sim.data$delta==1]
```

```

# Fit a Weibull calibration model for the covariate starting time distribution
calib.weib.params <- FitCalibWeibull(w = sim.data$w, w.res = sim.data$w.res)
px <- t(sapply(case.times, CalcWeibullCalibP, w = sim.data$w,
              w.res = sim.data$w.res, weib.params = calib.weib.params))
# Calculate derivative matrices
px.deriv.shape <- t(sapply(case.times, ICcalib::CalcWeibullCalibPderivShape,
w = sim.data$w, w.res = sim.data$w.res, weib.params = calib.weib.params))
px.deriv.scale <- t(sapply(case.times, ICcalib::CalcWeibullCalibPderivScale,
w = sim.data$w, w.res = sim.data$w.res, weib.params = calib.weib.params))
# Point estimate
est.weib.calib <- optimize(f = ICcalib::CoxLogLikX, tm = sim.data$obs.tm,
                          event = sim.data$delta, ps = px, interval = c(-50,50),
                          maximum = TRUE)$maximum
# Variance estimate (no additional covariates)
var.beta.wb <- CalcVarThetaWeib(beta = est.weib.calib, etas = calib.weib.params,
                                tm = sim.data$obs.tm, event = sim.data$delta,
                                ps = px, ps.deriv.shape = px.deriv.shape,
                                ps.deriv.scale = px.deriv.scale, w = sim.data$w,
                                w.res = sim.data$w.res)

print(est.weib.calib)
print(var.beta.wb)

```

---

CalcWeibullCalibP

---

*Calculating the probabilities of positive binary exposure status at a given time point using a Weibull calibration model*


---

## Description

For a given time point, calculate the probability of positive exposure value for multiple observations (participants). The function uses the results of a Weibull calibration model fit, and given collected data on the history of the binary exposure for each participant.

## Usage

```
CalcWeibullCalibP(w, w.res, point, weib.params)
```

## Arguments

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in w
point	The time point at which the probabilities are estimated.
weib.params	A bivariate vector. Shape and scale parameters of the Weibull calibration model.

## Value

A vector of estimated probabilities of positive exposure status at time point.



**See Also**[Weibull](#)**Examples**

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                             alpha = 0.25, beta0 = log(0.5),
                                             mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)

# Fit a Weibull calibration model for the covariate starting time distribution
calib.weib.params <- FitCalibWeibull(w = sim.data$w, w.res = sim.data$w.res)
# Calculate the conditional probabilities of binary covariate=1 at time one
probs <- CalcWeibullCalibP(w = sim.data$w, w.res = sim.data$w.res, point = 1,
                           weib.params = calib.weib.params)

summary(probs)
```

CalcWeibullRSP

*Calculating the probabilities of positive binary exposure status at a given time point using risk-set Weibull calibration models*

**Description**

For a given time point, calculate the probability of positive exposure value for multiple observations (participants). The function uses the results of a Weibull calibration model fit, and given collected data on the history of the binary exposure for each participant.

**Usage**

```
CalcWeibullRSP(w, w.res, point, weib.params)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in w
point	The time point at which the probabilities are estimated.
weib.params	A bivariate vector. Shape and scale parameters of the Weibull calibration model.

**Details**

At its present form this function is identical to CalcWeibullCalibP. This is because the current version of the ICcalib package (Version 1.0.005), the user loop over the main event times. Then, at each event time point, the user should include the appropriate Weibull parameters as estimated by FitCalibWeibullRS.

**Value**

A vector of estimated probabilities of positive exposure status at time point.

**See Also**

[Weibull](#)

**Examples**

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                             alpha = 0.25, beta0 = log(0.5),
                                             mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)

case.times <- sim.data$obs.tm[sim.data$delta==1]
# Fit Weibull risk-set calibration models
calib.weib.params <- FitCalibWeibullRS(w = sim.data$w, w.res = sim.data$w.res,
                                       tm = sim.data$obs.tm,
                                       event = sim.data$delta)

# Calculate the conditional probabilities of binary covariate=1 at time one
probs <- CalcWeibullRSP(w = sim.data$w, w.res = sim.data$w.res, point = 1,
                       weib.params = calib.weib.params)

summary(probs)
## Not run:
if(interactive()){
  #EXAMPLE1
}

## End(Not run)
```

---

FitCalibCox

---

*Fitting Proportional Hazards Calibration Models with Covariates*


---

**Description**

Fits a proportional hazards calibration model for time-to-exposure from interval-censored data with covariates. The exposure is a binary covariate measured in intermittent times. The covariates (Q) are associated with the time-to-exposure.

**Usage**

```
FitCalibCox(w, w.res, Q, hz.times, n.int = 5, order = 2)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. It corresponds to the time points in w

Q	Matrix of covariates for PH calibration model
hz.times	Times used for calculating the baseline hazard function from PH calibration model
n.int	The number of interior knots to be used, see <code>ICsurv::fast.PH.ICsurv.EM</code> , Default: 5
order	the order of the basis functions. See <code>ICsurv::fast.PH.ICsurv.EM</code> , Default: 2

**Value**

An object created by `ICsurv::fast.PH.ICsurv.EM`, with additional variables `knots` and `order`.

**See Also**

[fast.PH.ICsurv.EM](#)

**Examples**

```
sim.data <- ICcalib::SimCoxIntervalCensCox(n.sample = 200, lambda = 0.1,
                                           alpha = 0.25, beta0 = 0,
                                           gamma.q = c(log(0.75), log(2.5)),
                                           gamma.z = log(1.5), mu = 0.2,
                                           n.points = 2)
# The baseline hazard for the calibration model is calculated in observation times
cox.hz.times <- sort(unique(sim.data$obs.tm))
# Fit proportional hazards calibration model
FitCalibCox(w = sim.data$w, w.res = sim.data$w.res, Q = sim.data$Q,
            hz.times = cox.hz.times, n.int = 5, order = 2)
```

---

FitCalibCoxRS

---

*Fitting Proportional Hazards Risk-Set Calibration Models with Co-*  
*variates*


---

**Description**

`FitCalibCoxRS` fits proportional hazards risk-set calibration models for time-to-exposure from interval-censored data with covariates. The exposure is a binary covariate measured in intermittent times. The covariates (Q) are associated with the time-to-exposure. This function fits a calibration model at each main event time point, using only members of the risk set at that time point. model is fitted (for all the data) and used for that time point.

`FitCalibCoxRSInts` fits proportional hazards grouped risk-set calibration models for time-to-exposure from interval-censored data with covariates. The exposure is a binary covariate measured in intermittent times. The covariates (Q) are associated with the time-to-exposure. Unlike `FitCalibCoxRS`, this function fits a calibration model at each of the given points for `pts.for.ints`.

**Usage**

```
FitCalibCoxRS(w, w.res, Q, hz.times, tm, n.int = 5, order = 2, event)
```

```
FitCalibCoxRSInts(w, w.res, Q, hz.times, n.int = 5, order = 2, tm, event,  
pts.for.ints)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. It corresponds to the time points in w
Q	Matrix of covariates for PH calibration model
hz.times	Times used for calculating the baseline hazard function of a PH calibration model
tm	Vector of observed main event time or censoring time
n.int	The number of interior knots to be used, see <code>ICsurv::fast.PH.ICsurv.EM</code> , Default: 5
order	the order of the basis functions. See <code>ICsurv::fast.PH.ICsurv.EM</code> , Default: 2
event	Vector of censoring indicators. 1 for event 0 for censored
pts.for.ints	Points defining the intervals for grouping risk-sets (first one has to be zero). Should be sorted from zero up. <code>pts.for.ints</code> is used only for <code>FitCalibCoxRSInts</code> .

**Details**

In case of an error in the model-fitting at a certain time point, a proportional hazards calibration model (for all the data) is fitted and used for that time point.

**Value**

A list of Cox PH model fits, each supplemented with the knots and order used for the I-splines.

**See Also**

[fast.PH.ICsurv.EM](#), [FitCalibCox](#)

[fast.PH.ICsurv.EM](#), [FitCalibCox](#)

**Examples**

```
set.seed(2)
sim.data <- ICcalib::SimCoxIntervalCensCox(n.sample = 50, lambda = 0.1,
                                           alpha = 0.25, beta0 = log(0.2),
                                           gamma.q = c(log(0.75), log(2.5)),
                                           gamma.z = log(1.5), mu = 0.2,
                                           n.points = 2)

# The baseline hazard for the calibration model is calculated in observation times
cox.hz.times <- sort(unique(sim.data$obs.tm))
```

```
# Fit proportional hazards grouped risk-sets calibration models
calib.ph.rs.fit <- FitCalibCoxRSInts(w = sim.data$w, w.res = sim.data$w.res, Q = sim.data$Q,
                                     hz.times = cox.hz.times, tm = sim.data$obs.tm,
                                     event = sim.data$delta, pts.for.ints = seq(0, 3, 1.5),
                                     n.int = 5, order = 2)

# Below is a more time consuming option (no grouping of risk-sets)
# FitCalibCoxRS(w = sim.data$w, w.res = sim.data$w.res, Q = sim.data$Q,
#               hz.times = cox.hz.times, obs.tm = sim.data$obs.tm,
#               event = sim.data$delta, n.int = 5, order = 1)
```

FitCalibNpmle

*Fitting Nonparametric Calibration Models***Description**

Fits a Weibull calibration model for time-to-exposure from interval-censored data. The exposure is a binary covariate measured in intermittent times.

**Usage**

```
FitCalibNpmle(w, w.res)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. It corresponds to the time points in w

**Value**

The result of NPMLE model fitting for interval-censored data, as obtained by `icenReg::ic_np`

**See Also**

[ic\\_np](#)

**Examples**

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                              alpha = 0.25, beta0 = log(0.5),
                                              mu = 0.2, n.points = 2,
                                              weib.shape = 1, weib.scale = 2)

# Calculate the NPMLE for a nonparametric calibration model for the covariate
# starting-time distribution
npmle.fit <- ICcalib::FitCalibNpmle(w = sim.data$w, w.res = sim.data$w.res)
# Plot the estimated survival function
plot(npmle.fit)
```

FitCalibWeibull

*Fitting Weibull Calibration Models***Description**

Fits a Weibull calibration model for time-to-exposure from interval-censored data. The exposure is a binary covariate measured in intermittent times.

**Usage**

```
FitCalibWeibull(w, w.res)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. It corresponds to the time points in w

**Value**

A bivariate vector: the estimated Weibull shape and scale parameters

**See Also**

[fitdistcens](#)

**Examples**

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                             alpha = 0.25, beta0 = log(0.5),
                                             mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)

# Fit a Weibull calibration model for the covariate starting time distribution
ICcalib::FitCalibWeibull(w = sim.data$w, w.res = sim.data$w.res)
```

FitCalibWeibullRS

*Fitting Weibull Risk-Set Calibration Models***Description**

Fits Weibull risk-set calibration models for time-to-exposure from interval-censored data. The exposure is a binary covariate measured in intermittent times. This function fits a calibration model at each main event time point, using only members of the risk set at that time point.

**Usage**

```
FitCalibWeibullRS(w, w.res, tm, event, lower = 1e-04, upper = 200)
```

**Arguments**

w	A matrix of time points when measurements on the binary covariate were obtained.
w.res	A matrix of measurement results of the binary covariate. Each measurement corresponds to the time points in w
tm	Vector of observed main event time or censoring time.
event	Vector of censoring indicators. 1 for event 0 for censored
lower	A value to replace zero in the left point of the interval, Default: 1e-04
upper	A value to replace infinity in the right point of the interval, Default: 200

**Details**

In case of an error in the model-fitting at a certain time point, a Weibull calibration model is fitted and used for that time point.

**Value**

A 2-column matrix with the shape and scale parameter for each time-point at which a calibration model was fitted.

**See Also**

[fitdistcens](#), [FitCalibWeibull](#)

**Examples**

```
# Simulate data set
sim.data <- ICcalib::SimCoxIntervalCensSingle(n.sample = 200, lambda = 0.1,
                                             alpha = 0.25, beta0 = log(0.5),
                                             mu = 0.2, n.points = 2,
                                             weib.shape = 1, weib.scale = 2)

# Fit Weibull risk-set calibration models for the conditional covariate
# starting-time distributions
ICcalib::FitCalibWeibullRS(w = sim.data$w, w.res = sim.data$w.res,
                           tm = sim.data$obs.tm, event = sim.data$delta)
```

# Index

CalcCoxCalibP, [2](#)  
CalcCoxCalibRSIntsP, [3](#)  
CalcNpmleCalibP, [4](#)  
CalcNpmleRSP, [5](#)  
CalcVarNpmle (CalcVarParam), [6](#)  
CalcVarNpmleRS (CalcVarParam), [6](#)  
CalcVarParam, [6](#)  
CalcVarParamRSInts (CalcVarParam), [6](#)  
CalcVarThetaWeib (CalcVarParam), [6](#)  
CalcVarThetaWeibRS (CalcVarParam), [6](#)  
CalcWeibullCalibP, [8](#)  
CalcWeibullRSP, [9](#)  
  
fast.PH.ICsurv.EM, [11](#), [12](#)  
FitCalibCox, [10](#), [12](#)  
FitCalibCoxRS, [11](#)  
FitCalibCoxRSInts (FitCalibCoxRS), [11](#)  
FitCalibNpmle, [13](#)  
FitCalibWeibull, [14](#), [15](#)  
FitCalibWeibullRS, [14](#)  
fitdistcens, [14](#), [15](#)  
  
ic\_np, [5](#), [13](#)  
  
Weibull, [9](#), [10](#)