Package ‘merlin’

January 15, 2020

Version 0.0.2

Title Mixed Effects Regression for Linear, Non-Linear and User-Defined Models

Description Fits linear, non-linear, and user-defined mixed effects regression models following the framework developed by Crowther (2017) <arXiv:1710.02223>. 'merlin' can fit multivariate outcome models of any type, each of which could be repeatedly measured (longitudinal), with any number of levels, and with any number of random effects at each level. Standard distributions/models available include the Bernoulli, Gaussian, Poisson, beta, negative-binomial, and time-to-event/survival models include the exponential, Gompertz, Royston-Parmar, Weibull and general hazard model. 'merlin' provides a flexible predictor syntax, allowing the user to define variables, random effects, spline and fractional polynomial functions, functions of other outcome models, and any interaction between each of them. Non-linear and time-dependent effects are seamlessly incorporated into the predictor. 'merlin' allows multivariate normal random effects, which are integrated out using Gaussian quadrature or Monte-Carlo integration. Note, 'merlin' is based on the 'Stata' package of the same name, described in Crowther (2018) <arXiv:1806.01615>.

License GPL (>= 3)

Depends R (>= 2.10)

Imports MASS, randtoolbox, statmod, stats, survival, tools

Suggests knitr, nlme, rmarkdown, testthat

URL https://www.mjcrowther.co.uk/software/merlin

VignetteBuilder knitr

RoxygenNote 7.0.2

LazyData true

Encoding UTF-8

ByteCompile true

NeedsCompilation no

Author Emma Martin [aut], Alessandro Gasparini [aut], Michael Crowther [aut, cre]
Maintainer  Michael Crowther <michael.crowther@le.ac.uk>  
Repository  CRAN  
Date/Publication  2020-01-15 18:00:02 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>coef.merlin</td>
<td>3</td>
</tr>
<tr>
<td>coef.mlsurv</td>
<td>3</td>
</tr>
<tr>
<td>logLik.merlin</td>
<td>4</td>
</tr>
<tr>
<td>logLik.mlsurv</td>
<td>4</td>
</tr>
<tr>
<td>merlin</td>
<td>5</td>
</tr>
<tr>
<td>merlin_util_ap</td>
<td>9</td>
</tr>
<tr>
<td>merlin_util_ap_mod</td>
<td>10</td>
</tr>
<tr>
<td>merlin_util_depvar</td>
<td>11</td>
</tr>
<tr>
<td>merlin_util_ev</td>
<td>12</td>
</tr>
<tr>
<td>merlin_util_ev_deriv</td>
<td>12</td>
</tr>
<tr>
<td>merlin_util_ev_deriv2</td>
<td>13</td>
</tr>
<tr>
<td>merlin_util_ev_deriv2_mod</td>
<td>14</td>
</tr>
<tr>
<td>merlin_util_ev_deriv_mod</td>
<td>15</td>
</tr>
<tr>
<td>merlin_util_ev_integ</td>
<td>15</td>
</tr>
<tr>
<td>merlin_util_ev_integ_mod</td>
<td>16</td>
</tr>
<tr>
<td>merlin_util_ev_mod</td>
<td>17</td>
</tr>
<tr>
<td>merlin_util_timevar</td>
<td>18</td>
</tr>
<tr>
<td>merlin_util_xzb</td>
<td>18</td>
</tr>
<tr>
<td>merlin_util_xzb_deriv</td>
<td>19</td>
</tr>
<tr>
<td>merlin_util_xzb_deriv2</td>
<td>20</td>
</tr>
<tr>
<td>merlin_util_xzb_deriv2_mod</td>
<td>21</td>
</tr>
<tr>
<td>merlin_util_xzb_deriv_mod</td>
<td>22</td>
</tr>
<tr>
<td>merlin_util_xzb_integ</td>
<td>22</td>
</tr>
<tr>
<td>merlin_util_xzb_integ_mod</td>
<td>23</td>
</tr>
<tr>
<td>merlin_util_xzb_mod</td>
<td>24</td>
</tr>
<tr>
<td>mlsurv</td>
<td>25</td>
</tr>
<tr>
<td>pbc</td>
<td>26</td>
</tr>
<tr>
<td>predict.merlin</td>
<td>28</td>
</tr>
<tr>
<td>print.merlin</td>
<td>30</td>
</tr>
<tr>
<td>print.mlsurv</td>
<td>30</td>
</tr>
<tr>
<td>sin3</td>
<td>31</td>
</tr>
<tr>
<td>summary.merlin</td>
<td>31</td>
</tr>
<tr>
<td>summary.mlsurv</td>
<td>32</td>
</tr>
<tr>
<td>vcov.merlin</td>
<td>32</td>
</tr>
<tr>
<td>vcov.mlsurv</td>
<td>33</td>
</tr>
</tbody>
</table>

Index  34
coef.merlin

**Extract Model Coefficients**

**Description**

coef extracts model coefficients from a merlin model fit. coefficients is an alias for it.

**Usage**

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

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

**Arguments**

- **object**
  - An object of class merlin or summary.merlin.

- **...**
  - Not used.

---

coef.mlsurv

**Extract Model Coefficients**

**Description**

coef extracts model coefficients from a mlsurv model fit. coefficients is an alias for it.

**Usage**

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

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

**Arguments**

- **object**
  - An object of class mlsurv or summary.mlsurv.

- **...**
  - Not used.
logLik.merlin  
*Extract Log-Likelihood*

**Description**

Extract log-likelihood of a merlin model.

**Usage**

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

```r
## S3 method for class 'summary.merlin'
logLik(object, ...)
```

**Arguments**

- `object`: An object of class merlin or summary.merlin.
- `...`: Not used.

logLik.mlsurv  
*Extract Log-Likelihood*

**Description**

Extract log-likelihood of a mlsurv model.

**Usage**

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

```r
## S3 method for class 'summary.mlsurv'
logLik(object, ...)
```

**Arguments**

- `object`: An object of class mlsurv or summary.mlsurv.
- `...`: Not used.
merlin - Mixed Effects Regression for Linear, Nonlinear and User-defined models

Description

merlin fits linear, non-linear and user-defined mixed effects regression models. merlin can fit multivariate outcome models of any type, each of which could be repeatedly measured (longitudinal), with any number of levels, and with any number of random effects at each level. Standard distributions/models available include the Bernoulli, Gaussian, Poisson, beta, negative-binomial, and time-to-event/survival models include the exponential, Gompertz, Weibull, Royston-Parmar, and general hazard model. merlin provides a flexible predictor syntax, allowing the user to define variables, random effects, spline and fractional polynomial functions, functions of other outcome models, and any interaction between each of them. Non-linear and time-dependent effects are seamlessly incorporated into the predictor. merlin allows multivariate normal random effects, which are integrated out using Gaussian quadrature or Monte-Carlo integration. Relative survival (excess hazard) models are supported. Utility functions are provided to allow user-defined models to be specified, in conjunction with the complex predictor.

Usage

merlin(
  model,
  from = NULL,
  family = "gaussian",
  link = NULL,
  timevar = NULL,
  covariance = "diagonal",
  data,
  userf = NULL,
  sweights = NULL,
  levels = NULL,
  predict = FALSE,
  predtype = NULL,
  predmodel = NULL,
  causes = NULL,
  control = list()
)

Arguments

model

specify the fixed and random elements for each model outcome. Where there are multiple outcomes, the models should be specified in a list. Each model should be specified as a formula (e.g. \( y \sim x \)). A number of different element types can be specified, including

- varname - an independent variable from the data set
- random-effects - a random-effect at the cluster level can be specified using \( M^\text{[cluster level]} \), for example \( M^\text{id} \) would define a random intercept at the ID level. Each independent random-effect should be given a unique name, if two random-effects are given the same name they will be treated as shared random-effects.

- \( rcs() \) - restricted cubic spline terms, this option can be used to include a restricted cubic spline function, with the degrees of freedom (number of spline terms) specified using the df sub-option, with the boundary knots assumed to be at the minimum and maximum of the variable, with internal knots placed at equally spaced centiles. Other default options orthog = TRUE, which by default orthogonalises the splines, log = FALSE, which can be used to calculate splines of the log of the variable and event = F, which can be used to calculate the internal knots based only on observations that experienced the event of interest (for survival models).

- \( srcs() \) is a shorthand element, equivalent to \( rcs(...,\text{log}=\text{TRUE},\text{event}=\text{TRUE}) \), for use with survival models.

- \( fp() \) - fractional polynomials of order 1 or 2 can be specified, the sub-option powers is used to specify the powers of the fp model.

- \( \text{EV}[\text{depvar}] \) - the expected value of the response of a submodel.

- \( d\text{EV}[\text{depvar}] \) - the first derivative with respect to time of the expected value of the response of a submodel.

- \( d2\text{EV}[\text{depvar}] \) - the second derivative with respect to time of the expected value of the response of a submodel.

- \( \text{iEV}[\text{depvar}] \) - the integral with respect to time of the expected value of the response of a submodel.

- \( \text{XB}[\text{depvar}] \) - the expected value of the complex predictor of a submodel.

- \( d\text{XB}[\text{depvar}] \) - the first derivative with respect to time of the expected value of the complex predictor of a submodel.

- \( d2\text{XB}[\text{depvar}] \) - the second derivative with respect to time of the expected value of the complex predictor of a submodel.

- \( \text{iXB}[\text{depvar}] \) - the integral with respect to time of the expected value of the complex predictor of a submodel.

- \( \text{bhazard}([\text{varname}]) \) - invokes a relative survival (excess hazard) model. \( \text{varname} \) specifies the expected hazard rate at the event time.

- \( \text{exposure}([\text{varname}]) \) - include \( \log(\text{varname}) \) in the linear predictor, with a coefficient of 1. For use with family = "poisson".

- \( \text{ap}(\#) \) - defines the number of ancillary parameters. Used with family="user".

\textbf{from} this is an optional argument giving the initial values for the full parameter vector, for more details on how to specify the initial estimates see the vignette.

\textbf{family} a vector of strings specifying the family for each outcome specified in model. The currently available models include,

\* gaussian - Gaussian distribution
\* bernoulli - Bernoulli distribution
\* poisson - Poisson distribution
\* beta - Beta distribution
• negbinomial - Negative binomial distribution
  with survival models including,
  • exponential - exponential distribution
  • weibull - Weibull distribution
  • gompertz - Gompertz distribution
  • rp - Royston-Parmar model (complex predictor on the log cumulative hazard scale)
  • loghazard - general log hazard model (complex predictor on the log hazard scale)

and user-defined,
  • user - fit a user-defined model, which should be written using merlin’s utility functions. The name of your user-defined function should be passed through the userf option.
  • null - is a convenience tool to define additional complex predictors, that do not contribute to the log likelihood. For use with family = “user”.

link string vector defining the link functions for each model. Default is "identity" for all models. If specified, you must define a link function for all submodels. Options include "identity", "log" and "logit".

timevar specifies the variable which represents time, this is necessary when a function of time is used in the linear predictor of a survival model as it may interact with other elements of the model.

covariance the structure of the variance-covariance matrix can be varied, the default is diagonal where all diagonal elements of the variance-covariance matrix are estimated uniquely, identity assumes all diagonal elements are equal and unstructured estimates all elements of the variance-covariance matrix.

data a data frame containing all variables required for fitting the model. Can be a tibble object.

userf string vector defining the name of the user-written functions for each family="user". Each function must be in memory and should return the observation level contribution to the log-likelihood.

sweight Not documented.

timevar if the model contains random-effects then a vector giving the order of levels must be specified, from the highest level to the lowest, e.g. levels=c("practice","id").
predict Not documented.
predtype Not documented.
predmodel Not documented.
causes Not documented.
control A list of parameters that control the estimation algorithm. Generally it should not be modified, unless there are convergence issues. Possible values are:
  • ip An optional vector of integers specifying the number of integration points to be used when integrating out the random effects. A different number of ip can be specified for each level (from highest to lowest level).
If only a single number is given then merlin will assume that number of integration points at all levels. Default is \( ip = 7 \) for each level using Gauss-Hermite quadrature, or \( ip = 100 \) for each level using Monte-Carlo integration:

- \( \text{intmethod} \) The method used for numerically integrating out the random-effects in order to calculate the likelihood for a mixed effects model which includes random effects. Options include ghermite for non-adaptive Gauss-Hermite quadrature, halton for Monte-Carlo integration using Halton sequences, sobol for Monte-Carlo integration using Sobol sequences, or mc for standard Monte-Carlo integration using normal draws. The default is ghermite. Level-specific integration techniques can be specified, for example, with a three level model, we may use Gauss-Hermite quadrature at the highest level, and Monte-Carlo integration with Halton sequences at level 2, using \( \text{intmethod} = c( \text{"ghermite"}, \text{"halton"}) \).

- \( \text{debug} \) Not documented.

- \( \text{verbose} \) Not documented.

- \( \text{optim.method} \) The optim method to be used. Defaults to Nelder-Mead, see optim for available methods.

- \( \text{maxit} \) The maximum number of iterations for the optimisation routine. Defaults to 5000.

### Author(s)

Emma C. Martin, Alessandro Gasparini and Michael J. Crowther

### References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear, non-linear and user-defined models.

### See Also

predict.merlin
merlin_util_depvar, merlin_util_timevar, merlin_util_xzb, merlin_util_xzb_mod, merlin_util_xzb_deriv, merlin_util_xzb_deriv_mod, merlin_util_xzb_deriv2, merlin_util_xzb_deriv2_mod, merlin_util_xzb_integ, merlin_util_xzb_integ_mod, merlin_util_ev, merlin_util_ev_mod, merlin_util_ev_deriv, merlin_util_ev_deriv_mod, merlin_util_ev_deriv2, merlin_util_ev_deriv2_mod, merlin_util_ev_integ, merlin_util_ev_integ_mod, merlin_util_ap, merlin_util_ap_mod

### Examples

```r
## Not run:
library(merlin)
data(pbc.merlin, package = "merlin")
```
# Linear fixed-effects model
merlin(logb ~ year,
     family = "gaussian",
     data = pbc.merlin)

# Linear mixed-effects model with random intercept and slope at ID level
merlin(logb ~ year + M1[id] * 1 + year:M2[id] * 1,
     family = "gaussian",
     levels = "id",
     data = pbc.merlin)

# Joint longitudinal and survival model with shared random effects
merlin(model = list(logb ~ year + M1[id] * 1,
                 Surv(stime, died) ~ trt + M1[id]),
     family = c("gaussian", "weibull"),
     levels = "id",
     data = pbc.merlin)

# Joint longitudinal and survival model with expected value
merlin(model = list(logb ~ year + M1[id] * 1,
                 Surv(stime, died) ~ trt + EV[logb]),
     family = c("gaussian", "weibull"),
     levels = "id",
     timevar = c("year", "stime"),
     data = pbc.merlin)

# Gaussian distribution - implemented as a user-written family
logl_gaussian <- function(gml)
{
  y <- merlin_util_depvar(gml)
  xzb <- merlin_util_xzb(gml)
  se <- exp(merlin_util_ap(gml, 1))

  mu <- (sweep(xzb, 1, y, "+", "+") + xzb)
  logl <- ((-0.5 * log(2*pi) - log(se)) - (mu/(2 * se^2)))
  return(logl)
}

merlin(logb ~ year + ap(1), family = "user", data = pbc.merlin,
       userf = "logl_gaussian")

# 3-level Weibull model
merlin(Surv(stime1, dead1) ~ age + M1[id1]*1 + M2[id2]*1,
       levels=c("id1", "id2"), family="weibull", data=sim3)

# merlin_util_ap - returns the current estimate of the ith ancillary parameter for the associated model
**Description**

*merlin_util_ap* - returns the current estimate of the *i*th ancillary parameter for the associated model.

**Usage**

```
merlin_util_ap(gml, i)
```

**Arguments**

- `gml`: merlin object - should not be edited
- `i`: index for which ancillary parameter to extract

**Author(s)**

Michael J. Crowther

**References**


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

---

**Description**

*merlin_util_ap_mod* - returns the current estimate of the *i*th ancillary parameter for the specified model.

**Usage**

```
merlin_util_ap_mod(gml, m, i)
```

**Arguments**

- `gml`: merlin object - should not be edited
- `m`: specifies the merlin submodel
- `i`: index for which ancillary parameter to extract

**Author(s)**

Michael J. Crowther
References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

---

merlin_util_depvar

merlin_util_depvar - returns the response variable(s)

Description

Utility function to extract the dependent variable(s) for the current model. If the model is a survival/time-to-event model, then it will contain the event times in the first column and the event indicator in the second.

Usage

merlin_util_depvar(gml)

Arguments

gml

merlin object - should not be edited

Author(s)

Emma C. Martin, Alessandro Gasparini and Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.
merlin_util_ev

Description
Utility function to extract the expected value of the outcome at the current parameter estimates for a particular model.

Usage
merlin_util_ev(gml, t = NULL)

Arguments
gml merlin object - should not be edited
t specifies the variable which represents time

Author(s)
Michael J. Crowther

References
Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

merlin_util_ev_deriv

Description
Utility function to extract d/dt of the expected value of the outcome at the current parameter estimates for a particular model.

Usage
merlin_util_ev_deriv(gml, t = NULL)
Argument

- **gml**: merlin object - should not be edited
- **t**: specifies the variable which represents time

Author(s)

Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

Examples

```r
library(merlin)
data(pbc.merlin, package = "merlin")

# Linear fixed-effects model
merlin(model = list(logb ~ year),
       family = "gaussian",
       data = pbc.merlin)
```

Description

merlin_util_ev_deriv2 - returns the second derivative with respect to time of the observation-level expected value of the outcome

Usage

```r
merlin_util_ev_deriv2(gml, t = NULL)
```

Arguments

- **gml**: merlin object - should not be edited
- **t**: specifies the variable which represents time
merlin_util_ev_deriv2_mod

merlin_util_ev_deriv2_mod - returns the second derivative with respect to time of the observation-level expected value of the specified outcome

Description

merlin_util_ev_deriv2_mod - returns the second derivative with respect to time of the observation-level expected value of the specified outcome

Usage

merlin_util_ev_deriv2_mod(gml, m, t = NULL)

Arguments

gml merlin object - should not be edited
m specifies the merlin submodel
t specifies the variable which represents time

Author(s)

Michael J. Crowther

References

Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.
merlin_util_ev_deriv_mod

merlin_util_ev_deriv - returns the first derivative with respect to time of the observation-level expected value of the specified outcome

Description

Utility function to extract d/dt of the expected value of the outcome at the current parameter estimates for a specified model.

Usage

merlin_util_ev_deriv_mod(gml, m, t = NULL)

Arguments

gml merlin object - should not be edited
m specifies the merlin submodel
t specifies the variable which represents time

Author(s)

Michael J. Crowther

References

Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

merlin_util_ev_integ

merlin_util_ev_integ - returns the integral with respect to time of the observation-level expected value of the outcome

Description

merlin_util_ev_integ - returns the integral with respect to time of the observation-level expected value of the outcome

Usage

merlin_util_ev_integ(gml, t = NULL)
merlin_util_ev_integ_mod

Arguments

- **gml**: merlin object - should not be edited
- **t**: specifies the variable which represents time

Author(s)

- Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

---

**Description**

merlin_util_ev_integ_mod - returns the integral with respect to time of the observation-level expected value of the specified outcome

**Usage**

```r
merlin_util_ev_integ_mod(gml, m, t = NULL)
```

**Arguments**

- **gml**: merlin object - should not be edited
- **m**: specifies the merlin submodel
- **t**: specifies the variable which represents time

**Author(s)**

- Michael J. Crowther
References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

**merlin_util_ev_mod**

*merlin_util_ev_mod* - returns the observation-level expected value of the outcome for a specified model

Description

Utility function to extract the expected value of the outcome at the current parameter estimates for a specified model.

Usage

```r
merlin_util_ev_mod(gml, m, t = NULL)
```

Arguments

- `gml`: merlin object - should not be edited
- `m`: specifies the model
- `t`: specifies the variable which represents time

Author(s)

Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.
merlin_util_timevar - returns the time variable

Description
Utility function to extract the time variable(s) for the current model.

Usage
merlin_util_timevar(gml)

Arguments
- gml: merlin object - should not be edited

Author(s)
Michael J. Crowther

References
Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

merlin_util_xzb - returns the observation-level complex predictor

Description
Utility function to extract the complex predictor evaluated at the current parameter estimates for a particular model.

Usage
merlin_util_xzb(gml, t = NULL)

Arguments
- gml: merlin object - should not be edited
- t: specifies the variable which represents time
**merlin_util_xzb_deriv**

**Author(s)**

Emma C. Martin, Alessandro Gasparini and Michael J. Crowther

**References**


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

**Examples**

```r
library(merlin)
data(pbc.merlin, package = "merlin")

# Linear fixed-effects model
merlin(model = list(logb ~ year),
      family = "gaussian",
      data = pbc.merlin)
```

---

**merlin_util_xzb_deriv**  *merlin_util_xzb_deriv - returns the first derivative with respect to time of the observation-level complex predictor*

**Description**

Utility function to extract d/dt of the complex predictor evaluated at the current parameter estimates for a particular model.

**Usage**

```r
merlin_util_xzb_deriv(gml, t = NULL)
```

**Arguments**

- `gml` merlin object - should not be edited
- `t` specifies the variable which represents time

**Author(s)**

Michael J. Crowther
References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

Examples

```r
library(merlin)
data(pbc.merlin, package = "merlin")

# Linear fixed-effects model
merlin(model = list(logb ~ year),
       family = "gaussian",
       data = pbc.merlin)
```

---

`merlin_util_xzb_deriv2`

`merlin_util_xzb_deriv2` - returns the second derivative with respect to time of the observation-level complex predictor

Description

`merlin_util_xzb_deriv2` - returns the second derivative with respect to time of the observation-level complex predictor

Usage

`merlin_util_xzb_deriv2(glm, t = NULL)`

Arguments

- `glm` merlin object - should not be edited
- `t` specifies the variable which represents time

Author(s)

Michael J. Crowther
References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

merlin_util_xzb_deriv2_mod

*merlin_util_xzb_deriv2_mod* - returns the second derivative with respect to time of the observation-level complex predictor of the specified model

Description

merlin_util_xzb_deriv2_mod - returns the second derivative with respect to time of the observation-level complex predictor of the specified model

Usage

```
merlin_util_xzb_deriv2_mod(glm, m, t = NULL)
```

Arguments

- **glm**: merlin object - should not be edited
- **m**: specifies the merlin submodel
- **t**: specifies the variable which represents time

Author(s)

Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.
merlin_util_xzb_deriv_mod

merlin_util_xzb_deriv_mod - returns the first derivative with respect to time of the observation-level complex predictor of a specified model

Description

merlin_util_xzb_deriv_mod - returns the first derivative with respect to time of the observation-level complex predictor of a specified model

Usage

merlin_util_xzb_deriv_mod(gml, m, t = NULL)

Arguments

gml merlin object - should not be edited
m specifies the merlin submodel
t specifies the variable which represents time

Author(s)

Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

merlin_util_xzb_integ

merlin_util_xzb_integ - returns the integral with respect to time of the observation-level complex predictor

Description

merlin_util_xzb_integ - returns the integral with respect to time of the observation-level complex predictor

Usage

merlin_util_xzb_integ(gml, t = NULL)
**Arguments**

- `gml` merlin object - should not be edited
- `t` specifies the variable which represents time

**Author(s)**

Michael J. Crowther

**References**


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

---

**Description**

`merlin_util_xzb_integ_mod` - returns the integral with respect to time of the observation-level complex predictor of the specified model.

**Usage**

`merlin_util_xzb_integ_mod(gml, m, t = NULL)`

**Arguments**

- `gml` merlin object - should not be edited
- `m` specifies the merlin submodel
- `t` specifies the variable which represents time

**Author(s)**

Michael J. Crowther
References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.

merlin_util_xzb_mod

Description

Utility function to extract the complex predictor evaluated at the current parameter estimates for a particular model.

Usage

merlin_util_xzb_mod(gml, m, t = NULL)

Arguments

gml merlin object - should not be edited
m specifies the merlin submodel
t specifies the variable which represents time

Author(s)

Michael J. Crowther

References


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear and non-linear models.
mlsurv

Fit proportional hazards survival models

Description

Fit proportional hazards survival models

Usage

mlsurv(
  formula,
  distribution,
  df = NULL,
  powers = NULL,
  rcs = TRUE,
  data,
  from.null = NULL,
  ...
)

Arguments

- **formula**: A model formula, where the left-hand side is a Surv object.
- **distribution**: A parametric distribution for the baseline hazard. Possible values are exponential, weibull, gompertz, rp, logchazard, and loghazard. rp is equivalent to a logchazard model with restricted cubic splines (argument rcs = TRUE).
- **df**: Represents the number of degrees of freedom used for the restricted cubic splines when flexibly modelling the baseline hazard. Only required when rcs = TRUE.
- **powers**: A vector representing the degree of the fractional polynomials used to model the baseline hazard (with a maximum degree of 2). Only required when rcs = FALSE.
- **rcs**: Use restricted cubic splines when flexibly modelling the baseline hazard? Defaults to TRUE, and the alternative is using fractional polynomials.
- **data**: A data frame containing all variables required for fitting the model. Can be a tibble object.
- **from.null**: A vector of starting values for the null model (used to get improved starting values). This is mostly useful when experiencing issues with default starting values or convergence issues.
- **...**: Further arguments passed to merlin.

Value

An object of class mlsurv.
Examples

# Weibull model
library(survival)
data("pbc.merlin", package = "merlin")
fit <- mlsurv(
  formula = Surv(stime, died) ~ trt,
  distribution = "weibull",
  data = pbc.merlin
)
summary(fit)

# Royston-Parmar model with 3 degrees of freedom
fit <- mlsurv(
  formula = Surv(stime, died) ~ trt,
  distribution = "rp",
  df = 3,
  data = pbc.merlin
)
summary(fit)

## Not run:
# Flexible parametric model on the log-hazard scale with fractional polynomials
fit <- mlsurv(
  formula = Surv(stime, died) ~ trt,
  distribution = "loghazard",
  powers = c(0, 1),
  rcs = FALSE,
  data = pbc.merlin
)
summary(fit)

## End(Not run)

---

**pbc**  
*Mayo Clinic primary biliary cirrhosis data*

**Description**

This data is from the Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver conducted between 1974 and 1984. A total of 424 PBC patients, referred to Mayo Clinic during that ten-year interval met eligibility criteria for the randomized placebo controlled trial of the drug D-penicillamine, but only the first 312 cases in the data set participated in the randomized trial. Therefore, the data here are for the 312 patients with largely complete data.

**Usage**

data(pbc)

data(pbc.merlin)
**Format**

A data frame with 1945 observations on the following 20 variables:

- **id** patients identifier; in total there are 312 patients.
- **years** number of years between registration and the earlier of death, transplantation, or study analysis time.
- **status** a factor with levels alive, transplanted and dead.
- **drug** a factor with levels placebo and D-penicil.
- **age** at registration in years.
- **sex** a factor with levels male and female.
- **year** number of years between enrollment and this visit date, remaining values on the line of data refer to this visit.
- **ascites** a factor with levels No and Yes.
- **hepatomegaly** a factor with levels No and Yes.
- **spiders** a factor with levels No and Yes.
- **edema** a factor with levels No edema (i.e. no edema and no diuretic therapy for edema), edema no diuretics (i.e. edema present without diuretics, or edema resolved by diuretics), and edema despite diuretics (i.e. edema despite diuretic therapy).
- **serBilir** serum bilirubin in mg/dl.
- **serChol** serum cholesterol in mg/dl.
- **albumin** albumin in mg/dl.
- **alkaline** alkaline phosphatase in U/liter.
- **SGOT** SGOT in U/ml.
- **platelets** platelets per cubic ml/1000.
- **prothrombin** prothrombin time in seconds.
- **histologic** histologic stage of disease.
- **status2** a numeric vector with the value 1 denoting if the patient was dead, and 0 if the patient was alive or transplanted.

**Note**

`pbc.merlin` is a version of the `pbc` dataset in merlin format.

**Source**

`pbc`.

**References**


predict.merlin - post-estimation tools for merlin

Description
predictions following the fit of a merlin model

Usage
```r
## S3 method for class 'merlin'
predict(
  object,
  stat = "eta",
  type = "fixedonly",
  predmodel = 1,
  causes = NULL,
  ...
)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>merlin model object</td>
</tr>
<tr>
<td>stat</td>
<td>specifies which prediction, which can be one of:</td>
</tr>
<tr>
<td>type</td>
<td>specifies which prediction, which can be one of:</td>
</tr>
<tr>
<td>predmodel</td>
<td>specifies which prediction, which can be one of:</td>
</tr>
<tr>
<td>causes</td>
<td>specifies which prediction, which can be one of:</td>
</tr>
</tbody>
</table>

- `eta` the expected value of the complex predictor
- `mu` the expected value of the response variable
- `hazard` the hazard function
- `chazard` the cumulative hazard function
- `logchazard` the log cumulative hazard function
- `survival` the survival function
- `cif` the cumulative incidence function
- `rmst` calculates the restricted mean survival time, which is the integral of the survival function within the interval (0,t], where t is the time at which predictions are made. If multiple survival models have been specified in your merlin model, then it will assume all of them are cause-specific competing risks models, and include them in the calculation. If this is not the case, you can override which models are included by using the `causes` option. `rmst = t - totaltimelost`.
- `timelost` calculates the time lost due to a particular event occurring, within the interval (0,t]. In a single event survival model, this is the integral of the cif between (0,t]. If multiple survival models are specified in the merlin model then by default all are assumed to be cause-specific event time models contributing to the calculation. This can be overridden using the `causes` option.
predict.merlin

- `totaltimelost` total time lost due to all competing events, within \((0, t]\). If multiple survival models are specified in the merlin model then by default all are assumed to be cause-specific event time models contributing to the calculation. This can be overridden using the `causes` option. 
- `totaltimelost` is the sum of the `timelost` due to all causes.

**type**
- the type of prediction, either:
  - `fixedonly` prediction calculated based only on the fixed effects; the default.
  - `marginal` prediction calculated marginally with respect to the latent variables. The stat is calculated by integrating the prediction function with respect to all the latent variables over their entire support.

**predmodel**
- specifies which model to obtain predictions from; default is `predmodel=1`

**causes**
- is for use when calculating predictions from a competing risks merlin model. By default, `cif`, `rmst`, `timelost` and `totaltimelost` assume that all survival models included in the merlin model are cause-specific hazard models contributing to the calculation. If this is not the case, then you can specify which models (indexed using the order they appear in your merlin model by using the `causes` option, e.g. `causes=c(1,2)`.

... other options

**Author(s)**
Emma C. Martin, Alessandro Gasparini and Michael J. Crowther

**References**


Martin EC, Gasparini A, Crowther MJ. merlin - an R package for mixed effects regression of linear, non-linear and user-defined models.

**See Also**
merlin

**Examples**

```r
library(merlin)
data(pbc.merlin, package = "merlin")

# Linear fixed-effects model
mod <- merlin(model = list(logb ~ year),
              family = "gaussian",
              data = pbc.merlin)
predict(mod, stat="eta", type="fixedonly")
```
print.merlin  

Print merlin Fits

Description
Print the coefficients from a merlin fit.

Usage
```r
## S3 method for class 'merlin'
print(x, digits = max(3L, getOption("digits") - 3L), ...)  
```

Arguments
- `x`  An object of class merlin.
- `digits`  The number of significant digits to use when printing.
- `...`  Not used.

print.mlsurv  

Print mlsurv Fits

Description
Print the coefficients from a mlsurv fit.

Usage
```r
## S3 method for class 'mlsurv'
print(x, digits = max(3L, getOption("digits") - 3L), simplify = TRUE, ...)  
```

Arguments
- `x`  An object of class mlsurv.
- `digits`  The number of significant digits to use when printing.
- `simplify`  Should non-interpretible coefficients be hidden (e.g. splines and flexible polynomials terms)? Defaults to TRUE.
- `...`  Not used.
**Description**

Simulated 3-level survival data

**Usage**

data(sim3)

**Format**

A data frame...

**summary.merlin**

**Summarizing merlin Fits**

**Description**

These functions are all methods for class merlin or summary.merlin objects.

**Usage**

```r
## S3 method for class 'merlin'
summary(object, sig = 0.95, ...)
```

```r
## S3 method for class 'summary.merlin'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

**Arguments**

- **object**: An object of class merlin
- **sig**: Significance level for confidence intervals. Defaults to 0.95.
- **...**: Not used.
- **x**: An object of class summary.merlin
- **digits**: The number of significant digits to use when printing.
summary.mlsurv  
**Summarizing mlsurv Fits**

### Description
These functions are all methods for class `mlsurv` or `summary.mlsurv` objects.

### Usage

```r
## S3 method for class 'mlsurv'
summary(object, sig = 0.95, ...)

## S3 method for class 'summary.mlsurv'
print(x, digits = max(3, getOption("digits") - 3), simplify = TRUE, ...)
```

### Arguments

- `object`: An object of class `mlsurv`
- `sig`: Significance level for confidence intervals. Defaults to 0.95.
- `...`: Not used.
- `x`: An object of class `summary.mlsurv`
- `digits`: The number of significant digits to use when printing.
- `simplify`: Should non-interpretable coefficients be hidden (e.g. splines and flexible polynomials terms)? Defaults to TRUE.

vcov.merlin  
**Calculate Variance-Covariance Matrix for a merlin Model Object**

### Description
Returns the variance-covariance matrix of all estimated parameters of a fitted `merlin` model.

### Usage

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

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

### Arguments

- `object`: An object of class `merlin` or `summary.merlin`.
- `...`: Not used.
Calculate Variance-Covariance Matrix for a mlsurv Model Object

Description

Returns the variance-covariance matrix of all estimated parameters of a fitted mlsurv model.

Usage

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

## S3 method for class 'summary.mlsurv'
vcov(object, ...)

Arguments

- **object**: An object of class `mlsurv` or `summary.mlsurv`.
- **...**: Not used.
Index

*Topic datasets
  pbc, 26
  sim3, 31
coef.merlin, 3
coeff.mlsurv, 3
coeff.summary.merlin(coef.merlin), 3
coeff.summary.mlsurv(coef.mlsurv), 3
logLik.merlin, 4
logLik.mlsurv, 4
logLik.summary.merlin(logLik.merlin), 4
logLik.summary.mlsurv(logLik.mlsurv), 4
merlin, 5, 25, 29
merlin_util_ap, 8, 9
merlin_util_ap_mod, 8, 10
merlin_util_depvar, 8, 11
merlin_util_ev, 8, 12
merlin_util_ev_deriv, 8, 12
merlin_util_ev_deriv2, 8, 13
merlin_util_ev_deriv2_mod, 8, 14
merlin_util_ev_deriv_mod, 8, 15
merlin_util_ev_integ, 8, 15
merlin_util_ev_integ_mod, 8, 16
merlin_util_ev_mod, 8, 17
merlin_util_timevar, 8, 18
merlin_util_xzb, 8, 18
merlin_util_xzb_deriv, 8, 19
merlin_util_xzb_deriv2, 8, 20
merlin_util_xzb_deriv2_mod, 8, 21
merlin_util_xzb_deriv_mod, 8, 22
merlin_util_xzb_integ, 8, 22
merlin_util_xzb_integ_mod, 8, 23
merlin_util_xzb_mod, 8, 24
mlsurv, 25
optim, 8
pbc, 26, 27
predict.merlin, 8, 28
print.merlin, 30
print.mlsurv, 30
print.summary.merlin(summary.merlin),
  31
print.summary.mlsurv(summary.mlsurv),
  32
sim3, 31
summary.merlin, 31
summary.mlsurv, 32
Surv, 25
vcov.merlin, 32
vcov.mlsurv, 33
vcov.summary.merlin(vcov.merlin), 32
vcov.summary.mlsurv(vcov.mlsurv), 33
vcov.summary.mlsurv(summary.mlsurv), 32
vcov.summary.mlsurv(summary.mlsurv), 32