Package 'gcemod'

November 14, 2026

Type Package

Title Generalized Competing Event Models with Lunn-McNeil Testing

Version 0.4.0

Date 2026-11-13

Description Fits generalized competing event (GCE) models and estimates covariate effects on omega-plus, the ratio of the hazard for an event of interest to the hazard for a competing event, on both the cause-specific (Cox) and subdistribution (Fine-Gray) hazard scales. Confidence intervals and p-values are obtained from the Lunn-McNeil (1995) stacked (augmented) data approach. The package builds GCE risk scores from the model linear predictor, identifies risk-score cutpoints that maximize the separation in omega-plus between groups, and produces cumulative incidence (``alligator'') plots by risk group and calibration plots of predicted versus observed omega-plus. It also compares covariate effects across the primary, competing, and total (composite) events, and estimates covariate effects on the ratio of cumulative incidence functions (a cumulative-incidence-scale GCE model). Models are specified through a formula interface and return classed objects with summary, extractor, prediction, and plot methods. Methods follow Carmona et al. (2014) <doi:10.1016/j.ijrobp.2014.03.047>, Mell et al. (2024) <doi:10.1016/j.eururo.2023.01.020>, and Lunn and McNeil (1995) <doi:10.2307/2532940>.

License GPL (>= 2)

Encoding UTF-8

Language en-US

LazyData true

Depends R (>= 3.5.0)

Imports survival, cmprsk, ggplot2 (>= 3.5.2), patchwork, stats

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Loren Mell [aut, cre],
Morten Overgaard [ctb] (proposed the stacked pseudo-observation estimator used by gcecif())

Maintainer Loren Mell <lmell@ucsd.edu>

Repository CRAN

Date/Publication 2026-11-14 19:00:08 UTC

Contents

gcecif . . . 2
gcecox . . . 5
gcefg . . . 7
gcemod-methods . . . 9
gce_alligator . . . 13
gce_calibration . . . 15
gce_cutpoints . . . 16
gce_omegatime . . . 18
gce_performance . . . 20
gce_riskscore . . . 21
gce_scaling . . . 22
gce_tidy . . . 22
hn . . . 23
lunnmcneil . . . 24
prostate . . . 27

Index **29**

`gcecif` *Generalized competing event model on the cumulative-incidence-ratio scale*

Description

Fits a generalized competing event (GCE) model on the *cumulative incidence* scale, estimating covariate effects on the ratio of cumulative incidences $\rho(t) = F_1(t)/F_2(t)$ at a landmark time `t`, where F_1 and F_2 are the cumulative incidence functions of the event of interest and the competing event. $\rho(t)$ is the odds that a subject's *realized* event by time `t` is the event of interest rather than the competing event, so `gcecif()` is a cumulative-incidence-scale companion to the cause-specific (`gcecox`) and subdistribution (`gcefg`) omega+ models and returns a compatible `"gcemod"` object.

Effects are estimated by a *stacked pseudo-observation regression*: each subject contributes one jackknife pseudo-observation for $F_1(t)$ and one for $F_2(t)$ (both mean-type functionals of the Aalen-Johansen estimator, for which conditional unbiasedness of pseudo-observations is established), and a single log-link GEE is fit to the stacked pairs with an event-type indicator. The main effects model $\log F_2(t \mid z)$; the event-type interaction coefficients estimate $\log \rho(t \mid z) = \log F_1(t \mid z) - \log F_2(t \mid$

z) directly. Because each subject appears twice, a cluster-robust (subject) sandwich variance is used, which captures the dependence between a subject's two pseudo-observations induced by the shared all-cause survivor. This is the cumulative-incidence-scale analogue of the stacked Lunn-McNeil construction used on the two hazard scales. A GCE risk score is formed from the interaction linear predictor, so `exp(riskscore)` is a subject's ρ relative to the average subject (see `gce_riskscore`, `gce_scaling`).

Usage

```
gcecif(formula, ...)

## S3 method for class 'formula'
gcecif(
  formula,
  data,
  subset,
  na.action,
  t,
  event = NULL,
  competing = NULL,
  conf.level = 0.95,
  contrasts = NULL,
  model = TRUE,
  ...
)

## Default S3 method:
gcecif(formula, Ind, Cov, M = NULL, t, conf.level = 0.95, ...)
```

Arguments

`formula`	a formula with a `Surv(time, status)` response (formula method), or the numeric vector of follow-up times (default method, where it may also be named `Time`).
`...`	unused.
`data`	a data frame in which the formula is evaluated.
`subset, na.action`	as in `model.frame`.
`t`	landmark time at which ρ is evaluated (required).
`event, competing`	code (or factor level) marking the event of interest and the competing event; see `lunnmcneil`.
`conf.level`	confidence level (default `0.95`).
`contrasts`	optional contrasts specification passed to `model.matrix`.
`model`	logical; keep the model frame in the fitted object (default `TRUE`).
`Ind`	(default method) a data frame of event indicators. Column 1 = event of interest, column 2 = competing event (`0`/`1`).

`Cov`	(default method) a data frame of covariates.
`M`	deprecated and ignored. The number of bins is now an argument of the displays (`plot(fit, which = "riskscore", bins = )`, `gce_calibration`).

Details

The model is specified with a formula whose response is `Surv(time, status)`, as for `gcecox`; the landmark time `t` is part of the estimand and must be supplied. The default method is the matrix interface of earlier versions, retained for backward compatibility.

Because $F_k(t) = \int_0^t S(u^-)\, h_k(u)\, du$ with S the all-cause survivor, $\rho(t)$ is a survival-weighted time-average of the instantaneous cause-specific hazard ratio $\omega^+(u)$; under a time-constant ω^+ it equals ω^+ exactly. The ratio of cumulative incidences is a non-collapsible, marginal-over-time measure, so `exp(coef)` is interpreted as a conditional association with the realized event mix, not a mechanism-level per-cause effect.

The estimating function is *linear* in the two per-cause (mean-type) pseudo-observations, so its conditional unbiasedness follows from the standard pseudo-observation theory (Graw et al. 2009; Overgaard et al. 2017); pseudo-observations of the nonlinear ratio itself are not used. The working model comprises a log-linear model for $F_2(t \mid z)$ (the nuisance main effects, reported as `$coef_f2`) and a log-linear model for $\rho(t \mid z)$ (the target interactions, reported as `$coef`).

The estimand is defined for a genuine competing-risks setting, in which both $F_1(t)$ and $F_2(t)$ are appreciable at the landmark `t`. If the competing event is effectively absent, or `t` precedes appreciable competing-event incidence, $\log \rho$ is weakly identified and the fit may fail or return very large standard errors; this should be read as a signal that the competing-risks premise does not hold.

Value

An object of class `c("gcecif", "gcemod")`; see `gcemod-methods` for the available methods. The list contains `coef` and `var` (log ρ ratios and their covariance matrix), `omnibus` (Wald test that all ρ ratios equal 1), `riskscore` (interaction linear predictor, mean-centered), `scaling` (covariate means/SDs and per-SD effects), `F1`, `F2`, `rho0` (marginal $F_1(t)/F_2(t)$), `rho` (per-subject fitted ρ), `pseudo` (a two-column data frame of per-cause pseudo-observations), `coef_rho` and `coef_f2` (the full log-linear models for $\rho(t \mid z)$ and the nuisance $F_2(t \mid z)$, intercepts included), `converged`, and the model `terms`, `xlevels`, `contrasts`, and `model` frame.

References

Andersen PK, Klein JP, Rosthoj S (2003). Generalized linear models for correlated pseudo-observations, with applications to multi-state models. *Biometrika* 90:15-27.

Graw F, Gerds TA, Schumacher M (2009). On pseudo-values for regression analysis in competing risks models. *Lifetime Data Analysis* 15:241-255.

Overgaard M, Parner ET, Pedersen J (2017). Asymptotic theory of generalized estimating equations based on jack-knife pseudo-observations. *The Annals of Statistics* 45:1988-2015.

Overgaard M, Parner ET, Pedersen J (2018). Estimating the variance in a pseudo-observation scheme with competing risks. *Scandinavian Journal of Statistics* 45:923-940.

See Also

`gcecox`, `gcefg`, `gcemod-methods`, `gce_tidy`

Examples

```
data(prostate)
fit <- gcecif(Surv(time, status) ~ age + psa + gleason + t2b + comorbidity,
              data = prostate, t = 10)
summary(fit)
predict(fit, newdata = prostate[1:3, ], type = "ratio")
```

---

gcecox *Generalized competing event model via cause-specific Cox regression*

Description

Fits a generalized competing event (GCE) model on the cause-specific hazard scale and estimates covariate effects on ω^+ (event of interest vs. competing event) using the Lunn-McNeil stacked model (see lunnmcneil), with confidence intervals, Wald tests, and an omnibus test. A GCE risk score is formed from the model linear predictor.

Usage

```
gcecox(formula, ...)

## S3 method for class 'formula'
gcecox(
  formula,
  data,
  subset,
  na.action,
  t = NULL,
  event = NULL,
  competing = NULL,
  conf.level = 0.95,
  cluster.se = TRUE,
  standardize = FALSE,
  contrasts = NULL,
  model = TRUE,
  ...
)

## Default S3 method:
gcecox(
  formula,
  Ind,
  Cov,
  M = NULL,
  t = NULL,
  conf.level = 0.95,
  cluster.se = TRUE,
```

```
  standardize = FALSE,
  ...
)
```

Arguments

formula	a formula with a `Surv(time, status)` response (formula method), or the numeric vector of follow-up times (default method, where it may also be named `Time`).
...	passed to `coxph` for the stacked model.
data	a data frame in which the formula is evaluated.
subset, na.action	as in `model.frame`.
t	default readout time for time-dependent quantities. If `NULL` (default), the largest observed event time is used.
event, competing	code (or factor level) marking the event of interest and the competing event; see `lunnmcneil`.
conf.level	confidence level (default `0.95`).
cluster.se	logical; cluster-robust SEs in the Lunn-McNeil model (default `TRUE`).
standardize	logical; controls only the *reported* omega+ ratio scale (default `FALSE` = per natural covariate unit; `TRUE` = per 1 SD). The risk score is *always* the normalized (mean-centered) linear predictor regardless of this setting, so cutpoints, plots, and calibration are unaffected. Covariate means/SDs are returned by `gce_scaling`.
contrasts	optional contrasts specification passed to `model.matrix`.
model	logical; keep the model frame in the fitted object (default `TRUE`).
Ind	(default method) a data frame of event indicators. Column 1 = event of interest, column 2 = competing event (`0`/`1`).
Cov	(default method) a data frame of covariates.
M	deprecated and ignored. The number of bins is now an argument of the displays (`plot(fit, which = "riskscore", bins = )`, `gce_calibration`).

Details

The model is specified with a formula whose response is `Surv(time, status)`; see the *Interfaces* section of `lunnmcneil` for the accepted status codings. The fitted object stores the model terms, factor levels, and contrasts, so that `predict()` can evaluate the risk score and the predicted ω^+ for new data.

Fitting does not depend on a time point. The argument `t` only sets the default readout time for quantities that are functions of time: the predicted absolute $\omega^+(t)$ and $\omega(t)$ (`predict(type = "ratio")`, `type = "share"`), calibration (`gce_calibration`), and the time axis of the displays. It can be overridden in each of those functions.

The default method is the matrix interface of earlier versions, `gcecox(Time, Ind, Cov, t = )`, retained for backward compatibility.

Value

An object of class `c("gcecox", "gcemod")`. The components most often needed are available through methods rather than by name; see `gcemod-methods` (`print`, `summary`, `coef`, `vcov`, `confint`, `logLik`, `nobs`, `formula`, `terms`, `model.frame`, `model.matrix`, `predict`, `fitted`, `update`, `plot`). The list also contains `coef` and `var` (log omega+ ratios and their covariance matrix on the reported scale), `riskscore` (mean-centered linear predictor), `omegaplus` and `omega` (per-subject predictions at `t`), `omnibus`, `lunnmcneil` (the stacked-model fit), `scaling`, and the model `terms`, `xlevels`, `contrasts`, and `model` frame.

See Also

`gcefg`, `gcecif`, `lunnmcneil`, `gcemod-methods`, `gce_cutpoints`

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + smoker + t_cat + n_cat + p16,
              data = hn)
fit
summary(fit)
confint(fit)

# risk score and predicted omega+ at 5 years for new subjects
predict(fit, newdata = hn[1:3, ], type = "lp")
predict(fit, newdata = hn[1:3, ], type = "ratio", times = 5)

# matrix interface
Ind <- data.frame(event = as.integer(hn$status == 1),
                  competing = as.integer(hn$status == 2))
gcecox(hn$time, Ind, hn[, c("age", "smoker", "p16")], t = 5)
```

`gcefg` *Generalized competing event model via Fine-Gray regression*

Description

Fits a generalized competing event (GCE) model on the subdistribution hazard scale and estimates covariate effects on $\tilde{\omega}^{+}$ using a Lunn-McNeil-style stacked Fine-Gray model (see `lunnmcneil` with `type = "finegray"`), giving confidence intervals, Wald tests, and an omnibus test from one joint model. A GCE risk score is formed from the model linear predictor.

Usage

```
gcefg(formula, ...)

## S3 method for class 'formula'
gcefg(
  formula,
```

```
  data,
  subset,
  na.action,
  t = NULL,
  event = NULL,
  competing = NULL,
  conf.level = 0.95,
  standardize = FALSE,
  se = c("robust", "boot"),
  B = 200,
  contrasts = NULL,
  model = TRUE,
  ...
)

## Default S3 method:
gcefg(
  formula,
  Ind,
  Cov,
  M = NULL,
  t = NULL,
  conf.level = 0.95,
  standardize = FALSE,
  se = c("robust", "boot"),
  B = 200,
  ...
)
```

Arguments

`formula`	a formula with a `Surv(time, status)` response (formula method), or the numeric vector of follow-up times (default method, where it may also be named `Time`).
`...`	passed to `coxph` for the stacked model.
`data`	a data frame in which the formula is evaluated.
`subset, na.action`	as in `model.frame`.
`t`	default readout time for time-dependent quantities. If `NULL` (default), the largest observed event time is used.
`event, competing`	code (or factor level) marking the event of interest and the competing event; see `lunnmcneil`.
`conf.level`	confidence level (default `0.95`).
`standardize`	logical; controls only the *reported* omega+ ratio scale (default `FALSE` = per natural covariate unit; `TRUE` = per 1 SD). The risk score is *always* the normalized (mean-centered) linear predictor regardless of this setting, so cutpoints,

	plots, and calibration are unaffected. Covariate means/SDs are returned by gce_scaling.
se	standard-error method passed to lunnmcneil: "robust" (default; cluster-robust on subject, censoring weights treated as fixed, which errs on the conservative side) or "boot" (subject bootstrap that re-estimates the censoring weights within each resample, so the variance additionally accounts for their estimation, at higher computational cost).
B	number of bootstrap resamples when se = "boot" (default 200).
contrasts	optional contrasts specification passed to model.matrix.
model	logical; keep the model frame in the fitted object (default TRUE).
Ind	(default method) a data frame of event indicators. Column 1 = event of interest, column 2 = competing event (0/1).
Cov	(default method) a data frame of covariates.
M	deprecated and ignored. The number of bins is now an argument of the displays (plot(fit, which = "riskscore", bins =), gce_calibration).

Details

The interface, the role of t, and the returned object are as for gcecox. Predicted absolute $\tilde{\omega}^{+}(t)$ values are obtained from two separate Fine-Gray fits (crr), stored as $fit1 and $fit2.

Value

An object of class c("gcefg", "gcemod") with the same components and methods as a gcecox fit, on the subdistribution scale; see gcemod-methods.

See Also

gcecox, lunnmcneil, gcemod-methods

Examples

```
data(hn)
fit <- gcefg(Surv(time, status) ~ age + smoker + p16, data = hn, t = 5)
summary(fit)
```

---

gcemod-methods *Methods for fitted generalized competing event models*

Description

Methods for objects of class "gcemod" returned by gcecox, gcefg, and gcecif, and for the "gce_lunnmcneil" objects returned by lunnmcneil.

Usage

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

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

## S3 method for class 'gcemod'
summary(object, conf.level = object$conf.level, ...)

## S3 method for class 'gce_lunnmcneil'
summary(object, conf.level = object$conf.level, ...)

## S3 method for class 'summary.gcemod'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)

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

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

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

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

## S3 method for class 'gcemod'
confint(object, parm, level = 0.95, exponentiate = FALSE, ...)

## S3 method for class 'gce_lunnmcneil'
confint(object, parm, level = 0.95, exponentiate = FALSE, ...)

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

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

## S3 method for class 'gcemod'
formula(x, ...)
```

```
## S3 method for class 'gce_lunnmcneil'
formula(x, ...)

## S3 method for class 'gcemod'
terms(x, ...)

## S3 method for class 'gce_lunnmcneil'
terms(x, ...)

## S3 method for class 'gcemod'
model.frame(formula, ...)

## S3 method for class 'gcemod'
model.matrix(object, ...)

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

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

## S3 method for class 'gcemod'
predict(
  object,
  newdata = NULL,
  type = c("lp", "risk", "ratio", "share"),
  times = NULL,
  ...
)

## S3 method for class 'gcemod'
fitted(object, type = c("lp", "risk", "ratio", "share"), ...)

## S3 method for class 'gcemod'
plot(
  x,
  which = c("riskscore", "omegatime", "alligator", "calibration"),
  bins = 5,
  ...
)
```

Arguments

x, object	a fitted model object.
digits	number of significant digits to print.
...	for plot, further arguments passed to the display function; otherwise unused.
conf.level	confidence level of the intervals in the summary (defaults to the level used when fitting).

signif.stars	logical; print significance stars.
parm	coefficients for which intervals are wanted (names or indices).
level	confidence level.
exponentiate	logical; return intervals for the ratios exp(coef) instead of the log ratios.
formula	a fitted model object (the argument name is that of the model.frame generic).
newdata	optional data frame of new subjects. If omitted, predictions for the fitting data are returned.
type	type of prediction; see *Prediction*.
times	time(s) at which "ratio" and "share" are evaluated.
which	display to draw: "riskscore" (predicted ratio against the risk score, in bins quantile groups), "omegatime" (gce_omegatime), "alligator" (gce_alligator), or "calibration" (gce_calibration).
bins	number of quantile groups for which = "riskscore".

Details

print a short display: the call, the sample and event counts, and the estimated ratios exp(coef).

summary returns an object of class "summary.gcemod" (or "summary.gce_lunnmcneil") holding the full coefficient table (estimate, ratio, standard error, Wald z, p-value), the confidence intervals for the ratios, the omnibus Wald test, and the baseline quantity of the model ($\omega^{+}(t)$ of the average subject, or $F_1(t)$, $F_2(t)$, and $\rho(t)$). Its print method formats the table with printCoefmat.

coef, vcov, confint the log ratios (log ω^{+} ratios; log ρ ratios for gcecif), their covariance matrix, and Wald confidence intervals, all on the reported coefficient scale (per unit, or per SD if the model was fitted with standardize = TRUE). confint(exponentiate = TRUE) returns intervals for the ratios.

logLik the maximized log partial likelihood of the stacked model for gcecox fits, and the corresponding weighted log partial pseudo-likelihood for gcefg fits (useful for comparing nested specifications descriptively; it is not a genuine likelihood). gcecif is estimated by generalized estimating equations and has no likelihood, so logLik signals an error.

nobs, formula, terms, model.frame, model.matrix the number of subjects and the stored model specification. update() works through the stored call.

predict, fitted see below.

plot dispatches to the display functions; see which.

Value

summary returns an object of class "summary.gcemod" or "summary.gce_lunnmcneil"; coef a named numeric vector; vcov a matrix; confint a two-column matrix; logLik an object of class "logLik"; predict and fitted a numeric vector (or matrix, for several times); plot a ggplot object, invisibly. print methods return their argument invisibly.

Prediction

`predict()` rebuilds the design matrix for `newdata` from the stored terms, factor levels, and contrasts. `type` selects:

`"lp"` the GCE risk score: the linear predictor centered at the covariate means of the fitting data.

`"risk"` `exp(lp)`, the subject's ratio relative to the average subject.

`"ratio"` the absolute predicted ratio: $\omega^{+}(t)$ for `gcecox` (the marginal Nelson-Aalen ratio $H_1(t)/H_2(t)$ times `exp(lp)`), $\tilde{\omega}^{+}(t)$ for `gcefg`, and $\rho(t)$ for `gcecif`.

`"share"` `ratio / (1 + ratio)`: $\omega(t)$, or the share $\pi_1(t)$ for `gcecif`.

`times` defaults to the readout time `t` of the fit. With several `times` a matrix with one column per time is returned. For `gcecif` the landmark time is part of the fitted model and cannot be changed. `fitted()` is `predict()` on the fitting data.

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + smoker + p16, data = hn, t = 5)

fit                       # short display
s <- summary(fit)         # full table, returned as an object
s$coefficients
coef(fit); vcov(fit)
confint(fit, exponentiate = TRUE)
logLik(fit); nobs(fit); formula(fit)

head(fitted(fit))                                  # risk score
predict(fit, hn[1:3, ], type = "ratio", times = c(2, 5))

fit2 <- update(fit, . ~ . - smoker)
plot(fit, which = "riskscore", bins = 5)
```

---

gce_alligator *Alligator plot: cumulative incidence by risk group and event type*

Description

Plots the cause-specific cumulative incidence functions (CIFs) of the event of interest and the competing event over time on a single panel, with risk groups distinguished by color and event types by line type (event of interest solid, competing event dashed). In higher-risk groups the event-of-interest CIF rises above the competing-event CIF and the curves separate like an alligator's open jaws.

The x-axis is truncated at `xmax` (by default the model's readout time `t`), with tick marks at 1-unit (e.g. 1-year) intervals. The y-axis auto-scales its cap to the smallest of 25/50/75/100% that covers the peak cumulative incidence (so low-incidence settings are not compressed), with ticks at 10% (or 5% when capped at 25%). Numbers at risk for each group are shown in an aligned panel below the plot. Gray's test across groups is attached.

Usage

```
gce_alligator(
  object,
  groups = 2,
  method = c("optimal", "quantile"),
  criterion = c("omegaplus", "cif"),
  xlab = "Time (years)",
  ylab = "Cumulative incidence",
  group.lab = "Risk group",
  risk.table = TRUE,
  ymax = NULL,
  xmax = NULL
)

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

## S3 method for class 'gce_alligator'
plot(x, ...)
```

Arguments

object	a "gcemod" object from gcecox, gcefg, or gcecif.
groups	a gce_cutpoints result, a grouping factor of length n, or a single integer giving the number of risk groups (default 2). When an integer is given, the groups are derived with gce_cutpoints using method.
method	grouping rule used when groups is an integer: "optimal" (default) or "quantile" (equal-sized groups). Ignored when groups is a cutpoints object or a factor.
criterion	optimal-search criterion when groups is an integer and method = "optimal": "omegaplus" (default, log-ω^+ separation; not tail-sensitive) or "cif" (joint Gray statistic on the observed cumulative incidence of both events). See gce_cutpoints.
xlab, ylab	axis labels.
group.lab	legend title for the risk groups (default "Risk group").
risk.table	logical; show numbers at risk in an aligned panel below the plot (default TRUE).
ymax	optional numeric upper cap for the y-axis (0-1). If NULL (default) the cap auto-scales to 25/50/75/100% based on the peak incidence.
xmax	upper limit of the time axis; defaults to the readout time t of the fit.
x	a "gce_alligator" object.
...	unused.

Value

An object of class "gce_alligator" with print and plot methods. It is a list holding the cumulative incidence curves ($data), the numbers at risk ($n.risk), Gray's tests across groups ($grays_test), and the cutpoints. Printing shows the group sizes and Gray's tests; plot draws the figure and returns the ggplot/patchwork object invisibly. The display arguments (xlab, ylab, group.lab, risk.table, ymax) are stored and used by plot.

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + t_cat + p16, data = hn, t = 5)
al <- gce_alligator(fit, groups = 2)
al
plot(al)
```

`gce_calibration` *Calibration of predicted vs. observed omega+ by risk rank*

Description

Groups subjects into equal-sized rank groups on the GCE risk score and compares the model-predicted quantity with an observed estimate in each group. For omega+ models (`gcecox`, `gcefg`) the predicted ω^+ (or ω) is compared with observed cause-specific Nelson-Aalen cumulative hazards at time `t`: $\omega^+_{obs} = H_1(t)/H_2(t)$ and $\omega_{obs} = H_1(t)/(H_1(t) + H_2(t))$. For a `gcecif` fit the predicted cumulative-incidence ratio ρ (or share π_1) is compared with the observed Aalen-Johansen cumulative incidence within the group: $\rho_{obs} = F_1(t)/F_2(t)$ and $\pi_{1,obs} = F_1(t)/(F_1(t)+F_2(t))$. A calibration plot (predicted vs. observed with the identity line) and the underlying table are returned.

Usage

```
gce_calibration(object, which = NULL, groups = 5, t = NULL)

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

## S3 method for class 'gce_calibration'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'gce_calibration'
plot(x, ...)
```

Arguments

`object`	a `"gcemod"` object from `gcecox`, `gcefg`, or `gcecif`.
`which`	quantity to calibrate. For omega+ models `"omegaplus"` (default) or `"omega"`; for a `gcecif` fit `"rho"` (default) or `"share"`. If `NULL` the model-appropriate default is used.
`groups`	number of rank groups (default 5).
`t`	time at which predicted and observed quantities are compared; defaults to the readout time of the fit. Fixed at the landmark time for a `gcecif` fit.
`x`	a `"gce_calibration"` object.
`digits`	number of significant digits to print.
`...`	unused.
`row.names, optional`	ignored; present for compatibility with the generic.

Value

An object of class "gce_calibration" with print, plot, and as.data.frame methods. Printing shows the calibration table (group, n, predicted, observed); plot draws predicted against observed with the identity line and returns the ggplot object invisibly.

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + t_cat + p16, data = hn, t = 5)
cal <- gce_calibration(fit, which = "omegaplus", groups = 5)
cal
plot(cal)
```

---

gce_cutpoints *Risk-score cutpoints for GCE risk stratification*

Description

Splits subjects into ordered risk groups on the GCE risk score. With method = "optimal" the cutpoint(s) maximize a separation criterion by a greedy forward search; with method = "quantile" equal-sized groups are used.

The default criterion, "omegaplus", maximizes the between-group separation in **log** ω^+ (equivalently, the risk score / linear predictor). This is the additive scale on which the joint event-of-interest-vs-competing ω^+ contrast is defined, and – unlike separation measured on the exponential ω^+ scale – it is **not** sensitive to a few extreme high-risk subjects, so the search does not chase high-risk tails and tends to return balanced, well-separated groups. The alternative "cif" maximizes a joint Gray's test statistic (summed across both competing events) on the observed cumulative incidence.

Usage

```
gce_cutpoints(
  object,
  groups = 2,
  method = c("optimal", "quantile"),
  criterion = c("omegaplus", "cif"),
  min.prop = 0.1,
  nperm = 0,
  labels = NULL
)

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

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

## S3 method for class 'summary.gce_cutpoints'
```

```
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'gce_cutpoints'
plot(x, ...)
```

Arguments

object	a "gcemod" object from gcecox, gcefg, or gcecif (for the summary method, a "gce_cutpoints" object). For a gcecif fit the per-group summary reports mean ρ and share instead of ω^+ / ω; the "omegaplus" criterion still means separation on the risk-score (linear-predictor) scale.
groups	number of groups (default 2).
method	"optimal" (default) or "quantile".
criterion	optimal-search criterion: "omegaplus" (default, log-ω^+ separation) or "cif" (joint Gray statistic on observed cumulative incidence of both events).
min.prop	minimum proportion of subjects per group (default 0.1).
nperm	number of permutations for the selection-adjusted p-value (default 0 = not computed). When nperm > 0, the maximally-selected joint Gray statistic is permuted (risk score vs. outcome) to give a p-value accounting for the data-driven choice of cutpoint.
labels	optional character labels for the groups (low to high risk).
x	a "gce_cutpoints" object.
digits	number of significant digits to print.
...	unused.

Value

An object of class "gce_cutpoints", with print, summary, and plot methods. print shows the cutpoints, the per-group table, and the group contrast; summary returns the per-group table (a data frame) together with the test results as an object of class "summary.gce_cutpoints"; plot draws the distribution of the risk score with the cutpoints marked. The object is a list with: groups (ordered factor), cutpoints (on the risk-score scale), score (the risk score), summary (per-group n, events, mean ω^+ / ω), and test. The test element reports the Lunn-McNeil ω^+ contrast between groups – a test that the ratio of cause-specific (or subdistribution) hazards differs across groups – with the univariable ω^+ ratio estimate and CI ($omegaplus), the joint test ($statistic, $df, $p.value), the per-event Gray tests for comparison ($individual_gray), and a selection-adjusted $p.perm when nperm > 0.

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + t_cat + p16, data = hn, t = 5)
cp <- gce_cutpoints(fit, groups = 2)
cp
summary(cp)$groups
plot(cp)
table(cp$groups)
```

```
# with a selection-adjusted permutation p-value (slower)
gce_cutpoints(fit, groups = 2, nperm = 200)
```

`gce_omegatime` *GCE contrasts over time: a visual check of relative-hazard constancy*

Description

Plots a nonparametric estimate of a chosen GCE contrast over follow-up, as a visual assessment of the proportional relative-hazards (PRH) assumption and of the time range over which it holds approximately. All estimates are built from the Nelson-Aalen cause-specific cumulative hazards $\Lambda_k(t)$ and the Aalen-Johansen cumulative incidences $F_k(t)$:

`"omega"` **(default)** $\Lambda_1(t)/\{\Lambda_1(t)+\Lambda_2(t)\}$, the bounded cumulative hazard share, plotted on $[0,1]$.

`"omegaplus"` $\omega^+(t) = \Lambda_1(t)/\Lambda_2(t)$, the cause-specific relative hazard, on a log scale.

`"omegatilde"` $\tilde{\Lambda}_1(t)/\{\tilde{\Lambda}_1(t)+\tilde{\Lambda}_2(t)\}$, the subdistribution analogue of `"omega"`, with $\tilde{\Lambda}_k(t) = -\log\{1-F_k(t)\}$, on $[0,1]$.

`"omegatildeplus"` $\tilde{\omega}^+(t) = \tilde{\Lambda}_1(t)/\tilde{\Lambda}_2(t)$, the subdistribution relative hazard, on a log scale.

`"rho"` $\rho(t) = F_1(t)/F_2(t)$, the cumulative-incidence ratio, on a log scale.

`"pi"` $\pi(t) = F_1(t)/\{F_1(t)+F_2(t)\} = \rho(t)/\{1+\rho(t)\}$, the bounded share form of ρ (so $\operatorname{logit}\pi = \log\rho$), i.e. the probability that a realized event by t is the event of interest, on $[0,1]$.

If the cause-specific relative hazard is constant in time then $\Lambda_1(t) = \omega^+\Lambda_2(t)$ and, because $F_k(t) = \int_0^t S(u^-)\lambda_k(u)\,du$, also $F_1(t) = \omega^+F_2(t)$: the `"omega"`, `"omegaplus"`, `"rho"`, and `"pi"` displays are then exactly flat, and drift in any of them delineates where a single time-constant summary is and is not adequate. The subdistribution displays are flat only approximately (exactly so when both incidences are small, since $\tilde{\Lambda}_k \approx F_k$).

With `groups`, one curve is drawn per risk group. Approximately parallel group curves on a log-scaled contrast indicate a time-constant group *contrast*—the modeling assumption of the stacked estimators—even where the individual curves drift.

Usage

```
gce_omegatime(
  object,
  which = c("omega", "omegaplus", "omegatilde", "omegatildeplus", "rho", "pi"),
  groups = NULL,
  method = "optimal",
  xmax = NULL
)

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

```
## S3 method for class 'gce_omegatime'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'gce_omegatime'
plot(x, ...)
```

Arguments

object	a "gcemod" object from gcecox, gcefg, or gcecif.
which	the contrast to plot: "omega" (default), "omegaplus", "omegatilde", "omegatildeplus", "rho", or "pi". See Description.
groups	optional grouping: a gce_cutpoints result, a grouping factor of length n, or a single integer number of risk groups (derived with gce_cutpoints). NULL (default) draws one curve for the whole sample.
method	grouping rule used when groups is an integer: "optimal" (default) or "quantile".
xmax	optional upper limit of the time axis; the curves are truncated at xmax. Defaults to NULL (full follow-up). Late follow-up is often carried by few subjects at risk, so truncating (e.g. at or somewhat beyond the readout time t) can give a fairer picture of the range that matters for the fitted summary.
x	a "gce_omegatime" object.
digits	number of significant digits to print.
...	unused.
row.names, optional	ignored; present for compatibility with the generic.

Value

An object of class "gce_omegatime" with print, plot, and as.data.frame methods. The step-function estimates (group, time, value) are in $data. Printing gives, per group, the range of the contrast and its value at the readout time t; plot draws the curves (the readout time is marked with a dashed vertical line) and returns the ggplot object invisibly.

See Also

gce_cutpoints, gce_alligator

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + t_cat + p16, data = hn, t = 5)
ot <- gce_omegatime(fit)
ot
plot(ot)
plot(gce_omegatime(fit, which = "omegaplus", groups = 2, xmax = 10))
```

`gce_performance` *Discrimination (C-index) for a GCE model*

Description

Computes the cause-specific concordance (C-index) of the GCE risk score for the event of interest. For calibration (predicted vs. observed ω^{+}), see `gce_calibration`.

Usage

```
gce_performance(object, conf.level = 0.95)
```

Arguments

`object`	a `"gcemod"` object.
`conf.level`	confidence level for the C-index interval (default `0.95`).

Details

The C-index is computed with `concordance` using the event of interest as the outcome and the GCE risk score as the predictor. Higher risk scores denote higher risk; the statistic is oriented so that values above 0.5 indicate discrimination in the expected direction, and the `"flipped"` attribute of the result records whether the orientation was reversed.

Value

An object of class `"gce_performance"`: a named numeric vector with elements `estimate`, `se`, `lower`, and `upper`, carrying attributes `conf.level` and `flipped`. Printing the object displays the C-index with its confidence interval.

See Also

`gce_calibration`

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + t_cat + p16, data = hn, t = 5)
gce_performance(fit)
```

gce_riskscore *Normalized risk score and relative omega+ for new subjects*

Description

Computes the GCE risk score for new subjects using the covariate means/SDs and coefficients stored in a fitted model. This is a convenience wrapper around `predict()`, which offers the same quantities individually (`type = "lp"`, `"risk"`, `"ratio"`, `"share"`) and at arbitrary times. The score is the normalized (mean-centered) linear predictor, so `exp(riskscore)` is the subject's omega+ relative to the *average* subject in the fitting data (a value of 1 = same as average, 2 = twice the average omega+). For cause-specific (`gcecox`) models the absolute predicted omega+ and omega at the model's time `t` are also returned.

Usage

```
gce_riskscore(object, newdata)
```

Arguments

`object`	a `"gcemod"` object from `gcecox`, `gcefg`, or `gcecif`.
`newdata`	a data frame containing the covariates used to fit `object` (same column names and, for factors, levels seen in the fitting data).

Value

A data frame with `riskscore` and `exp(riskscore)` relative to the average subject: `omegaplus_rel` for omega+ models (with absolute `omegaplus` and `omega` at time `t` for cause-specific fits), or `rho_rel` for a `gcecif` fit.

See Also

`gce_scaling`

Examples

```
data(prostate)
fit <- gcecox(Surv(time, status) ~ age + psa + gleason, data = prostate, t = 10)
gce_riskscore(fit, newdata = prostate[1:5, ])
```

gce_scaling *Covariate scaling (means and SDs) used by the GCE risk score*

Description

Returns each covariate's mean and standard deviation (from the data used to fit the model), together with the per-SD log-omega+ coefficient and omega+ ratio. These are the quantities needed to reproduce the normalized risk score for new subjects (see `gce_riskscore`) and to interpret how far a subject sits from the average patient.

Usage

```
gce_scaling(object)
```

Arguments

`object`	a "gcemod" object from `gcecox`, `gcefg`, or `gcecif`.

Value

A data frame with columns `term`, `mean`, `sd`, `coef_perSD` (log ratio per 1 SD) and the per-SD ratio (`omegaplus_ratio_perSD` for omega+ models, `rho_ratio_perSD` for a `gcecif` fit).

See Also

`gce_riskscore`

Examples

```
data(prostate)
fit <- gcecox(Surv(time, status) ~ age + psa + gleason, data = prostate, t = 10)
gce_scaling(fit)
```

gce_tidy *Tidy a GCE model into a data frame of estimates*

Description

With `effects = FALSE` (the default) the omega+ ratio estimates are returned. With `effects = TRUE` a covariate-effects comparison table is returned that juxtaposes, for each covariate, the hazard ratio for the primary (event-of-interest) hazard, the competing-event hazard, the total (composite, any-event) hazard, and the relative hazard ratio (the omega+ ratio) from the GCE model, each with a confidence interval. This lets the investigator see whether a covariate acts mainly on the event of interest, the competing event, both, or on their balance.

Usage

```
gce_tidy(object, effects = FALSE, ...)
```

Arguments

object	a "gcemod" object from gcecox, gcefg, or gcecif. For a gcecif fit the relative column is the cumulative-incidence ratio (rho ratio) and the primary/competing/total columns are cause-specific hazard ratios.
effects	logical; if TRUE, return the primary/competing/total/ relative effects comparison table instead of the omega+ ratio table. Requires an object fit with **gcemod** >= 0.3.0 (which stores the design matrix). Default FALSE.
...	unused.

Details

The individual- and total-event columns are estimated on the same hazard scale as the fitted model: cause-specific Cox models for a gcecox fit, and Fine-Gray subdistribution models for a gcefg fit. The total (composite) event has no competing event, so its subdistribution and cause-specific hazards coincide and a Cox model on the composite endpoint is used in both cases.

Value

If effects = FALSE, a data frame with columns term, log_ratio, ratio, conf.low, conf.high, p.value. If effects = TRUE, a data frame with columns term, primary, competing, total, and relative, each a formatted "HR (low, high)" string.

Examples

```
data(hn)
fit <- gcecox(Surv(time, status) ~ age + t_cat + p16, data = hn, t = 5)
gce_tidy(fit)
gce_tidy(fit, effects = TRUE)
```

---

hn *Head-and-neck cancer competing-risks example dataset*

Description

An example head-and-neck cancer competing-risks dataset used to illustrate the **gcemod** functions. The data are provided for illustration only and do not represent real patients. The event of interest is cancer recurrence and the competing event is death without recurrence; recurrence is the more frequent event (recurrence-dominant, $\omega^+ > 1$). Contrast with prostate.

Usage

```
hn
```

Format

A data frame with 1000 rows and 11 variables:

id sequential subject identifier

time follow-up time in years

status competing-risks status: 0 = censored, 1 = cancer recurrence (event of interest), 2 = death without recurrence (competing event)

age age at diagnosis (years)

ps ECOG performance status (factor: 0, 1, 2)

female sex indicator (1 = female)

smoker smoking indicator (1 = more than 10 pack-years)

t_cat T category (factor: 1-4)

n_cat N category (factor: 0-3)

p16 p16 status (1 = positive)

site primary site (factor: oropharynx, larynx, hypopharynx, oralcavity)

`lunnmcneil` *Lunn-McNeil stacked-model estimation of omega+ ratios*

Description

Estimates covariate effects on ω^+ (the ratio of the hazard for the event of interest to the hazard for the competing event) from a single stacked ("augmented") model, following Lunn and McNeil (1995). Each subject contributes one record per event type; the model is stratified by event type and the covariate-by-event-type interaction estimates $\log(\omega^+ \text{ ratio})$ with model-based standard errors, per-covariate Wald tests, and an omnibus multivariate Wald test.

`lunnmcneil()` is the estimation engine behind `gcecox` and `gcefg`; use it directly when only the coefficient table and tests are needed, and the GCE fitters when the risk score, predictions, and displays are wanted as well.

Usage

```
lunnmcneil(formula, ...)

## S3 method for class 'formula'
lunnmcneil(
  formula,
  data,
  subset,
  na.action,
  event = NULL,
  competing = NULL,
  type = c("coxph", "finegray"),
```

```
  cluster.se = TRUE,
  conf.level = 0.95,
  standardize = FALSE,
  se = c("robust", "boot"),
  B = 200,
  contrasts = NULL,
  ...
)

## Default S3 method:
lunnmcneil(
  formula,
  cause,
  Cov,
  event = 1,
  competing = 2,
  type = c("coxph", "finegray"),
  cluster.se = TRUE,
  conf.level = 0.95,
  standardize = FALSE,
  se = c("robust", "boot"),
  B = 200,
  ...
)
```

Arguments

formula	a formula with a `Surv(time, status)` response (formula method), or the numeric vector of follow-up times (default method, where it may also be named `Time`).
...	passed to coxph.
data	a data frame in which the formula is evaluated.
subset, na.action	as in model.frame.
event	code (or factor level) marking the event of interest.
competing	code (or factor level) marking the competing event.
type	`"coxph"` (cause-specific, default) or `"finegray"` (subdistribution).
cluster.se	logical; cluster-robust (subject) SEs (default `TRUE`).
conf.level	confidence level (default `0.95`).
standardize	logical; controls only the *reported* coefficient scale (default `FALSE`). `FALSE` reports omega+ ratios per natural covariate unit; `TRUE` reports them per 1 SD. This is a reparameterization, so Wald tests, p-values, and the omnibus test are identical either way. The covariate means/SDs (`$center`, `$scale`) are always returned.
se	standard-error method: `"robust"` (default; model-based, cluster-robust on subject, with the censoring weights treated as fixed) or `"boot"` (subject bootstrap:

	the stacked model is refit on `B` resamples of subjects, re-estimating the censoring weights within each resample, so the variance additionally accounts for their estimation). Mainly relevant for `type = "finegray"`; because estimating the weights reduces the asymptotic variance, the fixed-weights default errs, if at all, on the conservative side.
`B`	number of bootstrap resamples when `se = "boot"` (default `200`).
`contrasts`	optional contrasts specification passed to `model.matrix`.
`cause`	(default method) event indicator per subject: `0` = censored, and the codes in `event` and `competing` for the two event types.
`Cov`	(default method) a data frame of covariates (factors are expanded to indicators).

Details

Two hazard scales are supported:

`type = "coxph"` Cause-specific hazards. The augmented data set duplicates each subject into an "event" stratum and a "competing" stratum; the interaction coefficient is exactly the difference in cause-specific log-hazards, i.e. $\log(\omega^+)$. This is the original Lunn-McNeil construction and the interaction MLE equals the difference of two separate cause-specific Cox fits.

`type = "finegray"` Subdistribution hazards. Risk-set-weighted data are generated for each event type with `finegray`, stacked, and fit with a single weighted, event-type-stratified `coxph`. The interaction estimates the difference in subdistribution log-hazards. Because subjects recur across the weighted risk sets, cluster-robust (subject) standard errors are used. This is an extension of Lunn-McNeil to the subdistribution scale.

Subject records are duplicated in both cases, so cluster-robust standard errors (clustered on subject) are the default.

Value

An object of class `"gce_lunnmcneil"`: a list with the fitted stacked model (`$fit`), the omega+ coefficient table (`$omegaplus`), the coefficient vector and its covariance matrix on the reported scale (`$coef`, `$var`), the reference (competing-event) coefficients (`$beta_competing`), the omnibus test (`$omnibus`), the model `terms`, `xlevels`, `contrasts` and `formula`, and the hazard scale (`$type`). Methods are available for `print`, `summary`, `coef`, `vcov`, `confint`, `nobs`, `logLik`, `formula`, and `terms`.

Interfaces

The formula method is the recommended interface. The response is written `Surv(time, status)`, where `status` is either numeric with `0` = censored and the codes given by `event` and `competing` (defaults `1` and `2`), or a factor whose first level denotes censoring (the convention of `Surv` for multi-state data), in which case `event` and `competing` default to the second and third levels. The right-hand side accepts the usual model-formula syntax (factors, interactions, transformations, splines). The model terms, factor levels, and contrasts are stored in the fitted object.

The default method is the matrix interface of earlier versions (`lunnmcneil(Time, cause, Cov)`); it is retained for backward compatibility and returns the same object.

References

Lunn M, McNeil D (1995) Applying Cox regression to competing risks. *Biometrics* 51:524-32.

See Also

gcecox, gcefg, gcemod-methods

Examples

```
data(hn)
fit <- lunnmcneil(Surv(time, status) ~ age + smoker + t_cat + n_cat + p16,
                  data = hn)
fit
summary(fit)
confint(fit)

# subdistribution scale
lunnmcneil(Surv(time, status) ~ age + smoker + p16, data = hn,
           type = "finegray")

# matrix interface
lunnmcneil(hn$time, hn$status, hn[, c("age", "smoker", "p16")])
```

---

prostate *Prostate cancer competing-risks example dataset*

Description

An example prostate cancer competing-risks dataset used to illustrate the **gcemod** functions. The data are provided for illustration only and do not represent real patients. The event of interest is distant metastasis or prostate-cancer death and the competing event is death from other causes; the competing event is the more frequent (competing-death-dominant, $\omega^{+} < 1$). Contrast with hn.

Usage

```
prostate
```

Format

A data frame with 1000 rows and 9 variables:

id sequential subject identifier

time follow-up time in years

status competing-risks status: 0 = censored, 1 = distant metastasis or prostate-cancer death (event of interest), 2 = death from other causes (competing event)

age age at diagnosis (years)

ps performance status (factor: 0, 1)

psa pre-treatment PSA (ng/mL, capped at 20)

gleason Gleason grade group (factor: "<=6", "3+4", "4+3", ">=8")

t2b stage indicator (1 = T2b or higher)

comorbidity significant comorbidity indicator (1 = yes)

Index

∗ **datasets**
hn, 23
prostate, 27

as.data.frame.gce_calibration *(*gce_calibration*)*, 15
as.data.frame.gce_omegatime *(*gce_omegatime*)*, 18

coef.gce_lunnmcneil *(*gcemod-methods*)*, 9
coef.gcemod *(*gcemod-methods*)*, 9
concordance, *20*
confint.gce_lunnmcneil *(*gcemod-methods*)*, 9
confint.gcemod *(*gcemod-methods*)*, 9
coxph, *6*, *8*, *25, 26*
crr, *9*

finegray, *26*
fitted.gcemod *(*gcemod-methods*)*, 9
formula.gce_lunnmcneil *(*gcemod-methods*)*, 9
formula.gcemod *(*gcemod-methods*)*, 9

gce_alligator, *12*, 13, *19*
gce_calibration, *4*, *6*, *9*, *12*, 15, *20*
gce_cutpoints, *7*, *14*, 16, *19*
gce_omegatime, *12*, 18
gce_performance, 20
gce_riskscore, *3*, 21, *22*
gce_scaling, *3*, *6*, *9*, *21*, 22
gce_tidy, *4*, 22
gcecif, 2, *7*, *9*, *14, 15*, *17*, *19*, *21–23*
gcecox, *2*, *4*, 5, *9*, *14, 15*, *17*, *19*, *21–24*, *27*
gcefg, *2*, *4*, *7*, 7, *9*, *14, 15*, *17*, *19*, *21–24*, *27*
gcemod-methods, 9

hn, 23, *27*

logLik.gce_lunnmcneil *(*gcemod-methods*)*, 9
logLik.gcemod *(*gcemod-methods*)*, 9
lunnmcneil, *3*, *5–9*, 24

model.frame, *3*, *6*, *8*, *12*, *25*
model.frame.gcemod *(*gcemod-methods*)*, 9
model.matrix, *3*, *6*, *9*, *26*
model.matrix.gcemod *(*gcemod-methods*)*, 9

nobs.gce_lunnmcneil *(*gcemod-methods*)*, 9
nobs.gcemod *(*gcemod-methods*)*, 9

plot.gce_alligator *(*gce_alligator*)*, 13
plot.gce_calibration *(*gce_calibration*)*, 15
plot.gce_cutpoints *(*gce_cutpoints*)*, 16
plot.gce_omegatime *(*gce_omegatime*)*, 18
plot.gcemod *(*gcemod-methods*)*, 9
predict, *6*, *21*
predict.gcemod *(*gcemod-methods*)*, 9
print.gce_alligator *(*gce_alligator*)*, 13
print.gce_calibration *(*gce_calibration*)*, 15
print.gce_cutpoints *(*gce_cutpoints*)*, 16
print.gce_lunnmcneil *(*gcemod-methods*)*, 9
print.gce_omegatime *(*gce_omegatime*)*, 18
print.gcemod *(*gcemod-methods*)*, 9
print.summary.gce_cutpoints *(*gce_cutpoints*)*, 16
print.summary.gcemod *(*gcemod-methods*)*, 9
printCoefmat, *12*
prostate, *23*, 27

summary.gce_cutpoints *(*gce_cutpoints*)*, 16
summary.gce_lunnmcneil *(*gcemod-methods*)*, 9
summary.gcemod *(*gcemod-methods*)*, 9
Surv, *26*

terms.gce_lunnmcneil *(*gcemod-methods*)*, 9
terms.gcemod *(*gcemod-methods*)*, 9

vcov.gce_lunnmcneil *(*gcemod-methods*)*, 9
vcov.gcemod *(*gcemod-methods*)*, 9