Package 'cureAssess'

September 14, 2026

Type Package

Title Assessing Cure Model Appropriateness for Survival Data

Version 0.1.0

Description Assesses whether cure models are appropriate for right-censored survival data, where a fraction of subjects may never experience the event of interest. Implements a two-stage workflow combining Kaplan-Meier visualization and comparison of parametric cure and non-cure models by the Akaike information criterion with formal diagnostics for sufficient follow-up and for the presence of a cured fraction. The diagnostics include the statistics of Maller and Zhou (1992) <doi:10.1093/biomet/79.4.731> and Maller and Zhou (1994) <doi:10.1080/01621459.1994.10476889>, the test of Shen (2000) <doi:10.1016/S0167-7152(00)00063-8>, and the ratio estimation of censored uncured subjects ('RECeUS') method of Selukar and Othus (2023) <doi:10.1002/sim.9610>.

License MIT + file LICENSE

URL https://github.com/GeethanjaleeM/cureAssess

BugReports https://github.com/GeethanjaleeM/cureAssess/issues

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports survival, flexsurv, flexsurvcure, survminer, ggplot2, dplyr, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Geethanjalee Mudunkotuwa [aut, cre, cph], Durbadal Ghosh [aut]

Maintainer Geethanjalee Mudunkotuwa <geethanjaleem@gmail.com>

Repository CRAN

Date/Publication 2026-09-14 15:50:02 UTC

Contents

`cure.appropriateness` *Assess cure model appropriateness in two stages*

Description

Performs a two-stage workflow for cure model assessment.

Usage

```
cure.appropriateness(
  data,
  time,
  status,
  time_scale = c("none", "days_to_years"),
  dist = NULL,
  plot_km = TRUE,
  run_tests = c("auto", "yes", "no"),
  include_lognormal = FALSE
)
```

Arguments

`data`	A data frame.
`time`	Character string giving the survival time column name.
`status`	Character string giving the event indicator column name.
`time_scale`	Either `"none"` or `"days_to_years"`.
`dist`	Optional distribution to use for the RECeUS method. If `NULL`, the function automatically selects the best-fitting cure-model distribution from the Stage 1 AIC table.
`plot_km`	Logical; if `TRUE`, include Kaplan-Meier plot object.
`run_tests`	One of `"auto"`, `"yes"`, or `"no"`.
`include_lognormal`	Logical; passed to `model.fitting()`. If `TRUE`, the lognormal cure and non-cure models are added to the Stage 1 candidate set. Defaults to `FALSE` so the candidate set matches the four distributions used in the tutorial. Because the automatic RECeUS distribution (`dist = NULL`) is selected as the smallest-AIC cure model, enabling lognormal can change both the selected model and the RECeUS conclusion.

Details

Stage 1:

- Prepares the data
- Fits a Kaplan-Meier curve
- Fits candidate cure and non-cure models
- Compares them using AIC
- Selects the model with the smallest AIC
- Provides an initial recommendation

Stage 2:

- Runs cure-appropriateness tests such as Maller-Zhou statistics, Shen's test, immune summaries and RECeUS-style outputs.

If the smallest-AIC model is a non-cure model, the function reports that a cure model is not supported by the initial model-comparison step, while still allowing the user to proceed with additional tests if desired.

Value

An object of class `"cure.appropriateness"` containing:

data The standardized survival dataset returned by `prepare.surv.data()`, containing the survival time (`Y`) and event indicator (`D`).

screening A list containing results from the model screening step.

screening$kmfit Kaplan-Meier estimate of the survival function (`survival::survfit` object).

screening$kmplot Kaplan-Meier plot generated using `survminer::ggsurvplot`, returned when `plot_km = TRUE`.

screening$aic_table A data frame summarizing the fitted candidate models, including model name, model type (`"cure"` or `"non-cure"`), AIC value, and parameter estimates.

screening$best_model The model with the smallest AIC.

screening$best_model_type Indicates whether the best model is a `"cure"` or `"non-cure"` model.

screening$initial_decision Text summarizing the initial interpretation based on the AIC comparison.

selected_receus_model The cure model selected for the RECeUS method, based on the smallest AIC among cure models.

selected_receus_dist The distribution code used by the RECeUS procedure.

tests A list containing results from additional cure-appropriateness diagnostics.

tests_run Logical indicator specifying whether additional diagnostic tests were executed.

tests_reason Explanation describing why the diagnostic tests were or were not executed.

final_recommendation A text summary combining the screening results and optional diagnostic tests to provide a final interpretation regarding cure model appropriateness.

See Also

`prepare.surv.data`, `model.fitting`, `run.cure.tests`

Examples

```
library(survival)

# Stage 1 only: Kaplan-Meier screening and AIC model comparison
res <- cure.appropriateness(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years",
  plot_km = FALSE,
  run_tests = "no"
)
res

# Stage 1 and Stage 2: also run the cure-appropriateness diagnostics
res_full <- cure.appropriateness(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years",
  plot_km = FALSE,
  run_tests = "yes"
)
summary(res_full)
```

`immune.test` *Compute immune-related summary quantities*

Description

Computes Kaplan-Meier-based summary quantities used to assess whether right-censored survival data may contain a subgroup of long-term survivors (sometimes referred to as "immune" or "cured" individuals).

Usage

```
immune.test(dat)
```

Arguments

`dat`	A data frame containing columns `Y` and `D`, where `Y` is the observed survival time and `D` is the event indicator (`1` = event, `0` = censoring).

Details

The method evaluates the behavior of the Kaplan-Meier survival curve at the end of follow-up. In particular, it considers:

- the estimated event probability by the end of follow-up,
- the proportion of censored observations, and
- whether the largest observed follow-up time corresponds to a censored observation.

If the largest observed time is censored and the Kaplan-Meier curve appears to level off above zero, this is consistent with the presence of a survival plateau and may suggest a cure fraction. In contrast, if the largest observed time is an event, the evidence for a survival plateau is weaker.

This function provides a descriptive summary rather than a formal hypothesis test and is intended to be interpreted together with other diagnostics such as the Maller-Zhou test, qn statistic, Shen test, and RECeUS method.

Value

An object of class `"immune.test.result"` containing:

method Name of the diagnostic summary.

p_hat Estimated event probability by the end of follow-up, derived from the Kaplan-Meier curve.

p_cens Observed proportion of censored observations.

last_observation Largest observed follow-up time.

last_observation_censored Logical indicator of whether the largest observed follow-up time is censored.

interpretation Text describing the implication of the summary quantities for the presence of a survival plateau and possible cure fraction.

References

Maller RA, Zhou S (1992). *Estimating the proportion of immunes in a censored sample.* Biometrika, 79(4), 731–739. doi:10.1093/biomet/79.4.731

Maller RA, Zhou S (1994). *Testing for sufficient follow-up and outliers in survival data.* Journal of the American Statistical Association, 89(428), 1499–1506. doi:10.1080/01621459.1994.10476889

Maller RA, Zhou S (1995). *Testing for the presence of immune or cured individuals.* Biometrics, 51, 1197–1205. doi:10.2307/2533253

Maller RA, Zhou X (1996). *Survival Analysis with Long-term Survivors.* Wiley.

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

res <- immune.test(dat)
res
```

`model.fitting` *Fit candidate cure and non-cure survival models*

Description

Fits a set of parametric survival models and corresponding cure models to right-censored survival data, compares them using the Akaike Information Criterion (AIC), and identifies the model with the smallest AIC.

Usage

```
model.fitting(data, plot_km = TRUE, include_lognormal = FALSE)
```

Arguments

`data`	A data frame containing columns `Y` and `D`, where `Y` is the observed survival time and `D` is the event indicator (`1` = event, `0` = censoring).
`plot_km`	Logical; if `TRUE`, also returns a Kaplan-Meier plot created using `survminer::ggsurvplot`.
`include_lognormal`	Logical; if `TRUE`, the lognormal cure and non-cure models are added to the candidate set. Defaults to `FALSE` so that the default candidate set matches the four distributions used in the tutorial (exponential, Weibull, gamma, log-logistic). The lognormal distribution has a heavy tail that can substantially change the RECeUS remaining-uncured ratio and the selected model, so it is opt-in.

Details

The default candidate distributions are exponential, Weibull, gamma, and log-logistic, matching the four distributions used in the tutorial worked example. The lognormal distribution is available as an optional addition via `include_lognormal = TRUE`.

For each distribution, both a non-cure model and a cure model are fitted. A Kaplan-Meier estimate is also computed, and a Kaplan-Meier plot can be optionally returned.

Value

An object of class `"cure.model.fit"` containing:

kmfit A `survival::survfit` object representing the Kaplan-Meier estimate of the survival function.

kmplot A Kaplan-Meier plot object created using `survminer::ggsurvplot`. This is returned only when `plot_km = TRUE`.

fits A named list of fitted model results. Each element contains:

- **fit** The fitted model object, or `NULL` if fitting failed.
- **AIC** The Akaike Information Criterion value for the fitted model.
- **error** An error message if model fitting failed, otherwise `NULL`.

aic_table A data frame summarizing the fitted models, including model name, model type (`"cure"` or `"non-cure"`), AIC value, parameter estimates, and fitting errors. The table is ordered by increasing AIC.

best_model A character string giving the name of the model with the smallest AIC.

best_model_type A character string indicating whether the best model is a `"cure"` or `"non-cure"` model.

Objects of class `"cure.model.fit"` represent the model-screening stage of cure model assessment and are typically used as input for downstream functions such as `run.cure.tests()` and `cure.appropriateness()`.

See Also

`prepare.surv.data`, `run.cure.tests`, `cure.appropriateness`, `print.cure.model.fit`, `summary.cure.model.fit`, `plot.cure.model.fit`

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

fit_res <- model.fitting(dat, plot_km = FALSE)
```

```
fit_res$aic_table
fit_res$best_model
fit_res$best_model_type
```

`mz.test` *Compute the Maller-Zhou test statistic*

Description

Computes the Maller-Zhou test statistic used to assess whether a cure fraction may exist in right-censored survival data. The test is based on examining the behavior of events near the end of follow-up.

Usage

```
mz.test(dat, alpha = 0.05)
```

Arguments

`dat`	A data frame containing columns `Y` and `D`, where `Y` is the observed survival time and `D` is the event indicator (`1` = event, `0` = censoring).
`alpha`	Significance level used for interpretation. Default is `0.05`.

Details

The key idea is that if a cure fraction exists, the Kaplan-Meier survival curve will eventually reach a plateau above zero because a subset of individuals will never experience the event. In contrast, if no cure fraction exists, events should continue to occur toward the end of follow-up.

The Maller-Zhou statistic evaluates the number of events occurring in a time window near the largest observed event time. If relatively few events occur in this region, it provides evidence consistent with the presence of a cure fraction.

The test can only be computed when the largest observed time corresponds to a censored observation (i.e., follow-up extends beyond the last event).

Value

An object of class `"cure.test.result"` containing:

method Name of the test.

statistic Computed Maller-Zhou statistic.

alpha Significance level used for interpretation.

interpretation Text describing the implication of the test result for the presence of a cure fraction.

References

Maller RA, Zhou S (1992). *Estimating the proportion of immunes in a censored sample.* Biometrika, 79(4), 731–739. doi:10.1093/biomet/79.4.731

Maller RA, Zhou S (1994). *Testing for sufficient follow-up and outliers in survival data.* Journal of the American Statistical Association, 89(428), 1499–1506. doi:10.1080/01621459.1994.10476889

Maller RA, Zhou X (1996). *Survival Analysis with Long-Term Survivors.* Wiley.

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

res <- mz.test(dat)
res
```

---

plot.cure.model.fit *Plot a fitted cure model object*

Description

Displays the Kaplan-Meier plot stored in a `cure.model.fit` object.

Usage

```
## S3 method for class 'cure.model.fit'
plot(x, ...)
```

Arguments

`x`	An object of class `"cure.model.fit"`.
`...`	Additional arguments passed to other methods.

Value

The input object `x`, returned invisibly.

Examples

```
library(survival)

dat <- prepare.surv.data(gbsg, "rfstime", "status", "days_to_years")
fit_res <- model.fitting(dat, plot_km = TRUE)

plot(fit_res)
```

`prepare.surv.data` *Prepare survival data for cure model assessment*

Description

Standardizes survival data into a format required by the package. The returned data frame always contains columns `Y` for survival time and `D` for event indicator, D = 1 for event and D = 0 for censoring.

Usage

```
prepare.surv.data(data, time, status, time_scale = c("none", "days_to_years"))
```

Arguments

`data`	A data frame.
`time`	Character string giving the survival time column name.
`status`	Character string giving the event indicator column name.
`time_scale`	Either `"none"` or `"days_to_years"`.

Value

A data frame with standardized columns `Y` for survival time and `D` for event indicator, D = 1 for event and D = 0 for censoring.

See Also

`model.fitting`, `cure.appropriateness`

Examples

```
library(survival)

# `rfstime` is recorded in days, so convert it to years
dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
```

```
)

head(dat[, c("Y", "D")])
```

`print.cure.appropriateness` *Print cure model appropriateness analysis results*

Description

Prints a summary of a `cure.appropriateness` object, including the screening results, RECeUS model selection, testing status, and final recommendation.

Usage

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

Arguments

`x`	An object of class `"cure.appropriateness"`.
`...`	Additional arguments passed to other methods.

Value

The input object `x`, returned invisibly.

`print.cure.model.fit` *Print a fitted cure model object*

Description

Prints a summary of a `cure.model.fit` object, including the best model identified by AIC and the full AIC comparison table.

Usage

```
## S3 method for class 'cure.model.fit'
print(x, ...)
```

Arguments

`x`	An object of class `"cure.model.fit"`.
`...`	Additional arguments passed to other methods.

Value

The input object x, returned invisibly.

`print.cure.test.result` *Print cure model test results*

Description

Prints the results of a cure model diagnostic test, including the test statistic, significance level (if available), and interpretation.

Usage

```
## S3 method for class 'cure.test.result'
print(x, ...)
```

Arguments

x	An object of class "cure.test.result".
...	Additional arguments passed to other methods.

Value

The input object x, returned invisibly.

`print.immune.test.result` *Print immune test results*

Description

Prints the results of the immune-based cure model diagnostic test, including estimated proportions and censoring information.

Usage

```
## S3 method for class 'immune.test.result'
print(x, ...)
```

Arguments

x	An object of class "immune.test.result".
...	Additional arguments passed to other methods.

Value

The input object x, returned invisibly.

print.receus.output *Print RECeUS cure model assessment results*

Description

Prints the results from a RECeUS-based cure model assessment, including the estimated cure fraction, remaining uncured ratio, and the decision regarding cure model appropriateness.

Usage

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

Arguments

x	An object of class "receus.output".
...	Additional arguments passed to other methods.

Value

The input object x, returned invisibly.

print.summary.cure.appropriateness *Print a summary of cure model appropriateness results*

Description

Prints the condensed summary produced by summary.cure.appropriateness(), including the best model identified during screening, the AIC comparison table, whether diagnostic tests were run, and the final recommendation.

Usage

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

Arguments

x	An object of class "summary.cure.appropriateness".
...	Additional arguments passed to other methods.

Value

The input object x, returned invisibly.

`qn.test` *Compute the qn statistic*

Description

Computes the **qn statistic** proposed by Maller and Zhou (1996), which is a descriptive diagnostic used to assess whether follow-up is sufficient to detect a survival plateau in right-censored survival data. A survival plateau may indicate the presence of a *cure fraction*, meaning that a subset of individuals will never experience the event of interest.

Usage

```
qn.test(dat)
```

Arguments

`dat`	A data frame containing columns `Y` and `D`, where `Y` is the observed survival time and `D` is the event indicator (`1` = event, `0` = censoring).

Details

The statistic is `qn = Nn / n`, where `Nn` is the number of events falling in the late-time window `(2*Y* - Y_max, Y*]`, `Y*` is the largest event time, and `Y_max` is the largest observed time. The width of this window equals the gap `Y_max - Y*` between the last event and the end of follow-up. A long plateau (sufficient follow-up) produces a wide window that captures many events, so **larger** values of `qn` indicate stronger evidence of sufficient follow-up and a survival plateau. **Smaller** values indicate that events continue up to the end of follow-up, providing weaker evidence for a plateau.

For a formal decision, this implementation uses the companion Maller-Zhou statistic `alpha_n = (1 - qn)^n` (Maller & Zhou 1994, their eq. 5): sufficient follow-up is supported when `alpha_n < 0.05`, equivalently when `qn` exceeds `1 - 0.05^(1/n)` (a threshold that depends on the sample size `n`). Note this `alpha_n`-equivalent rule is the same decision used by `mz.test()`; the exact finite-sample critical values for `qn` derived by Maller, Resnick and Shemehsavar (2024) are not implemented here. The `qn` statistic itself is reported and is directionally informative regardless of the cutoff used.

The statistic can only be computed when the largest observed follow-up time is a censored observation (i.e., follow-up extends beyond the last event).

Value

An object of class `"cure.test.result"` containing:

method Name of the statistic.

statistic Computed qn statistic.

interpretation Text describing the implication of the statistic for the presence of a survival plateau and possible cure fraction.

References

Maller RA, Resnick S, Shemehsavar S (2024). *Finite sample and asymptotic distributions of a statistic for sufficient follow-up in cure models.* Canadian Journal of Statistics, 52(2), 359–379. doi:10.1002/cjs.11771

Maller RA, Zhou S (1992). *Estimating the proportion of immunes in a censored sample.* Biometrika, 79(4), 731–739. doi:10.1093/biomet/79.4.731

Maller RA, Zhou S (1994). *Testing for sufficient follow-up and outliers in survival data.* Journal of the American Statistical Association, 89(428), 1499–1506. doi:10.1080/01621459.1994.10476889

Maller RA, Zhou X (1996). *Survival Analysis with Long-term Survivors.* Wiley.

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

res <- qn.test(dat)
res
```

`receus.method` *Compute RECeUS cure model diagnostic outputs*

Description

Computes cure-fraction-related summary quantities using a selected parametric survival model and evaluates cure model appropriateness using the RECeUS method proposed by Selukar and Othus (2023).

Usage

```
receus.method(data, dist = "exp", whichTau = NULL)
```

Arguments

`data`	A data frame containing columns `Y` and `D`, where `Y` is the observed survival time and `D` is the event indicator (`1` = event, `0` = censoring).
`dist`	Character string specifying the parametric model used in the RECeUS calculation. Supported values are: `"exp"` Exponential cure model. `"expUnc"` Exponential non-cure model.

"wei" Weibull cure model.
"weiUnc" Weibull non-cure model.
"llogis" Log-logistic cure model.
"llogisUnc" Log-logistic non-cure model.
"gam" Gamma cure model.
"gamUnc" Gamma non-cure model.
"lnorm" Lognormal cure model.
"lnormUnc" Lognormal non-cure model.

whichTau Optional evaluation time. If NULL, the largest observed follow-up time is used.

Details

The RECeUS method is based on two quantities:

- pi_hat: the estimated cure fraction
- r_hat: the estimated proportion of uncured subjects remaining censored at the end of follow-up

This diagnostic is typically applied after model screening when a cure model is selected as the preferred model based on information criteria such as AIC (e.g., "exp", "wei", "llogis", "gam", "lnorm"). However, the method can also be applied using non-cure parametric models.

When a non-cure model is used (e.g., "expUnc", "weiUnc", "llogisUnc", "gamUnc", "lnormUnc"), the cure fraction is constrained to zero by the model specification. In this case the method will always return:

- pi_hat = 0
- r_hat = 1

A cure model is considered appropriate when both of the following conditions are satisfied:

- pi_hat > 0.025
- r_hat < 0.05

The first condition indicates that the estimated cure fraction is meaningfully greater than zero, while the second condition indicates that only a small proportion of uncured subjects remain censored at the end of follow-up. Together, these conditions suggest both the presence of a cure fraction and sufficient follow-up for reliable cure model estimation.

This function can be applied using either cure or non-cure parametric model specifications, depending on the value of dist.

Value

An object of class "receus.output" containing:

method Name of the diagnostic procedure.

dist Distribution used in the RECeUS calculation.

tau Evaluation time used in the calculation.

estimates Raw vector of estimated model quantities returned by the internal RECeUS calculation.

pi_hat Estimated cure fraction.

r_hat Estimated proportion of uncured subjects remaining censored at the end of follow-up.

cure_fraction_condition Logical indicator of whether `pi_hat > 0.025`.

followup_condition Logical indicator of whether `r_hat < 0.05`.

decision Text summary of whether the RECeUS criteria support cure model appropriateness.

interpretation Text describing the implication of the RECeUS quantities for cure model appropriateness and sufficiency of follow-up.

References

Selukar S, Othus M (2023). *RECeUS: Ratio estimation of censored uncured subjects.* Statistics in Medicine, 42(3), 209–227. doi:10.1002/sim.9610

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

res <- receus.method(dat, dist = "lnorm")
res
```

`run.cure.tests` *Run cure model appropriateness diagnostics*

Description

Runs a set of diagnostic procedures used to assess whether a cure model may be appropriate for right-censored survival data. The diagnostics include the Maller-Zhou test, qn statistic, Shen test, immune summary, and RECeUS method.

Usage

```
run.cure.tests(data, dist = NULL)
```

Arguments

`data`	A data frame containing columns `Y` and `D`, where `Y` is the observed survival time and `D` is the event indicator (`1` = event, `0` = censoring).
`dist`	Character string giving the distribution to be used for the RECeUS method. If `NULL`, the distribution must be supplied elsewhere or selected before calling this function.

Value

An object of class "cure.tests" containing:

mz Result of the Maller-Zhou diagnostic test.

qn Result of the qn statistic.

shen Result of Shen's test.

immune Result of the immune summary diagnostic.

receus Result of the RECeUS method.

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

res <- run.cure.tests(dat, dist = "lnorm")
res$mz
res$receus
```

---

shen.test *Compute Shen's test statistic*

Description

Computes the Shen (2000) diagnostic test statistic used to assess whether follow-up in right-censored survival data is sufficient to detect a potential cure fraction.

Usage

```
shen.test(dat, alpha = 0.05)
```

Arguments

dat	A data frame containing columns Y and D, where Y is the observed survival time and D is the event indicator (1 = event, 0 = censoring).
alpha	Significance level used for interpretation. Default is 0.05.

Details

The test examines the pattern of events near the end of follow-up. If a cure fraction exists, the Kaplan-Meier survival curve will tend to flatten (plateau) because a subset of individuals will never experience the event of interest. In contrast, if no cure fraction exists, events should continue to occur late in follow-up and the survival curve will continue to decline.

Shen's test evaluates whether the number of events occurring in a late follow-up interval is consistent with the presence of such a plateau. Smaller values of the test statistic provide stronger evidence supporting the presence of a cure fraction.

The test can only be computed when the largest observed follow-up time corresponds to a censored observation (i.e., follow-up extends beyond the last observed event time).

Value

An object of class `"cure.test.result"` containing:

method Name of the diagnostic test.

statistic Computed Shen test statistic.

alpha Significance level used for interpretation.

interpretation Text describing the implication of the statistic for the presence of a survival plateau and possible cure fraction.

References

Shen P-S (2000). *Testing for sufficient follow-up in survival data.* Statistics & Probability Letters, 49(4), 313–322. doi:10.1016/S01677152(00)000638

Examples

```
library(survival)

dat <- prepare.surv.data(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years"
)

res <- shen.test(dat)
res
```

summary.cure.appropriateness *Summarize cure model appropriateness results*

Description

Returns a summary of a `cure.appropriateness` object, including the best model identified during screening, the AIC comparison table, tests performed, and the final recommendation.

Usage

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

Arguments

`object`	An object of class `"cure.appropriateness"`.
`...`	Additional arguments passed to other methods.

Value

An object of class `"summary.cure.appropriateness"` containing key results from the cure model appropriateness analysis.

Examples

```
library(survival)

res <- cure.appropriateness(
  data = gbsg,
  time = "rfstime",
  status = "status",
  time_scale = "days_to_years",
  plot_km = FALSE,
  run_tests = "no"
)

summary(res)
```

summary.cure.model.fit *Summarize a fitted cure model object*

Description

Returns the AIC comparison table from a `cure.model.fit` object.

Usage

```
## S3 method for class 'cure.model.fit'
summary(object, ...)
```

Arguments

`object`	An object of class `"cure.model.fit"`.
`...`	Additional arguments passed to other methods.

Value

A data frame containing the AIC comparison of candidate models.

Examples

```
library(survival)

dat <- prepare.surv.data(gbsg, "rfstime", "status", "days_to_years")
fit_res <- model.fitting(dat, plot_km = FALSE)

summary(fit_res)
```

Index