Package ‘PAmeasures’

February 5, 2025

Type Package

Title Prediction and Accuracy Measures for Nonlinear Models and for Right-Censored Time-to-Event Data

Version 0.1.0

Date 2018-01-18

Author Xiaoyan Wang, Gang Li

Maintainer Xiaoyan Wang<xywang@ucla.edu>

Description We propose a pair of summary measures for the predictive power of a prediction function based on a regression model. The regression model can be linear or nonlinear, parametric, semi-parametric, or nonparametric, and correctly specified or mis-specified. The first measure, R-squared, is an extension of the classical R-squared statistic for a linear model, quantifying the prediction function's ability to capture the variability of the response. The second measure, L-squared, quantifies the prediction function's bias for predicting the mean regression function. When used together, they give a complete summary of the predictive power of a prediction function. Please refer to Gang Li and Xiaoyan Wang (2016) <arXiv:1611.03063> for more details.

Depends R (>= 3.1)

Imports survival, stats

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-22 10:06:41 UTC

Contents

`moore` *Moore's Law data*

Description

A dataset containing the number of transistors and the corresponding #' years. The Moore's Law #'states that the number of transistors in a dense integrated circuit doubles approximately every two #'years. moore.

Usage

```
moore
```

Format

A data frame with 48 rows and 3 variables:

year year, from 1973 to 2011

time time starting from 1973

count number of transistors

`pam.Brier` *Brier Score Calculation*

Description

The Brier Score, proposed by Glenn W. Brier in 1950, is a proper score function used to measure the accuracy of probabilistic predictions. It is commonly applied to assess model fits for survival data. The Brier Score can be calculated at any time point, regardless of whether it coincides with the event time.

The Brier Score represents the mean squared difference between true classes and predicted probabilities, effectively serving as a cost function. A lower Brier Score indicates better-calibrated predictions. Its values range from zero to one, as it reflects the maximum possible squared difference between predicted probabilities and actual outcomes.

In the context of censored samples, where the exact time of an event (e.g., death) is unknown, direct calculation of residuals is not feasible. Thus, the Brier Score is widely utilized in survival analysis.

The Brier Score is considered a strictly proper score (Gneiting and Raftery, 2007), meaning it achieves its minimum value only when the predicted probabilities align with empirical probabilities. Empirical evidence suggests that predictions of survival duration can be inaccurate; however, incorporating patient-specific survival probabilities along with the Brier Score improves the ability to differentiate between future survivors and failures.

Usage

```
pam.Brier(object, pre_sp, t_star = -1)
```

Arguments

`object` An object of class `Surv`, created by the `Surv` function, or a fitted survival model, such as those produced by `coxph`, `survreg`, or `rfsrc`.

`pre_sp` If `object` is a fitted survival model, this parameter should be a dataset on which you want to calculate the Brier Score. If `object` is a survival object, this parameter should be a vector of predicted survival probabilities for each observation at time `t_star`.

`t_star` A specified time point for calculating the Brier Score. This is necessary when `object` is a fitted survival model, as it indicates when the survival probability is predicted. If `object` is a survival object, this parameter can be ignored and does not affect the function's outcome.

Value

The Brier Score at time `t_star`, representing the difference between true classes and predicted probabilities.

References

Graf, E., Schmoor, C., Sauerbrei, W., & et al. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistical Medicine*, 18(17-18), 2529-2545.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78.

Gneiting, T., & Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation.

Zhou, H., Cheng, X., Wang, S., Zou, Y., & Wang, H. (2022). SurvMetrics: Predictive Evaluation Metrics in Survival Analysis. R package version 0.5.0. Available at https://github.com/skyee1/SurvMetrics.

Examples

```
library(survival)

# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)

# Fit an exponential model with bilirubin
fit.coxph.full <- coxph(Surv(time, status) ~ age + log_albumin +
                                    log_bili + log_protime + edema,
                                data = pbc,x=TRUE,y=TRUE)
```

```
taulist <- seq(0, max(pbc$time), 300)
median_time <- median(pbc$time)
pam.Brier(fit.coxph.full, pbc, median_time)
```

`pam.Brier_metric` *Calculate Brier Score for Survival Data*

Description

Computes the Brier score for survival data to evaluate the accuracy of predicted survival probabilities at a specific time point. The Brier score measures the mean squared error between the predicted probabilities and the actual outcomes, adjusted for censoring using the Kaplan-Meier estimator.

Usage

```
pam.Brier_metric(predicted_data, suvival_time, t_star)
```

Arguments

`predicted_data` A numeric vector of predicted survival probabilities for each observation. The length must match the number of rows in `suvival_time`.

`suvival_time` A survival object created using the `Surv` function. It must include survival times and event status indicators.

`t_star` A positive numeric value specifying the time point at which the Brier score is calculated.

Details

The Brier score is calculated by partitioning the observations into two groups:

- Observations with event times less than `t_star`.
- Observations with event times greater than or equal to `t_star`.

The score is adjusted for censoring using the Kaplan-Meier estimate of the censoring distribution. Observations are weighted accordingly to account for the censoring bias.

Value

A single numeric value representing the Brier score, rounded to six decimal places. The score reflects the accuracy of the predicted survival probabilities at the specified time point.

References

Graf, E., Schmoor, C., Sauerbrei, W., & Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18(17-18), 2529-2545.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78, 1-3.

Gneiting, T., & Raftery, A. E. (2007). Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477), 359-378.

Zhou, H., Cheng, X., Wang, S., Zou, Y., & Wang, H. (2022). *SurvMetrics: Predictive Evaluation Metrics in Survival Analysis* (R package version 0.5.0). Available at https://github.com/skyee1/SurvMetrics.

Examples

```
library(survival)
# Example dataset
data(lung)
lung$SurvObj <- with(lung, Surv(time, status == 2))

# Simulated predicted probabilities
set.seed(123)
predicted_probs <- runif(nrow(lung), 0, 1)

# Calculate Brier score at t_star = 200
brier_score <- pam.Brier_metric(
  predicted_data = predicted_probs,
  suvival_time = lung$SurvObj,
  t_star = 200
)
print(brier_score)
```

pam.censor *Prediction Accuracy Measures for Regression Models of Right-Censored Data*

Description

This function calculates a pair of measures, R-Squared and L-Squared, for any regression models of right-censored data. R-squared is an extension of the classical R2 statistic for a linear model, quantifying the amount of variability in the response that is explained by a corrected prediction based on the original prediction function. L-squared is the proportion of the prediction error of the original prediction function that is explained by the corrected prediction function, quantifying the distance between the corrected and uncorrected predictions. When used together, they give a complete summary of the predictive power of a prediction function.

Usage

```
pam.censor(y, y.predict, delta)
```

Arguments

`y`	A numeric vector containing the response values.
`y.predict`	A numeric vector containing the predicted response values from a fitted model.
`delta`	A numeric vector indicating the status of the event, normally 0=alive, 1=dead.

Value

A list containing two components: R-squared and L-squared

Examples

```
library(survival)
library(PAmeasures)

# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)

# Fit an exponential model with bilirubin
fit1 <- survreg(Surv(time, status==2) ~ bili, data = pbc,dist="exponential" )

# Obtain predicted response from the fitted exponential model
predict.time<-predict(fit1,type="response")

# Recode status at endpoint, 0 for censored, 1 for dead
delta.pbc<- as.numeric(pbc$status == 2)

# R.squared and L.squared of log-linear model
pam.censor(pbc$time, predict.time, delta.pbc)
```

pam.concordance *Compute the Concordance Statistic*

Description

This function computes the concordance statistic, which measures the agreement between an observed response and a predictor. It is closely related to Kendall's tau-a and tau-b, Goodman's gamma, and Somers' d, all of which can also be calculated from the results of this function. This function handles different model types and can be used with new data to evaluate but not refit the models.

Usage

```
pam.concordance(object, ...)
```

Arguments

`object`	A fitted model or a formula of the form y ~ x or y ~ x + strata(z) with a single numeric or survival response and a single predictor. Counts of concordant, discordant and tied pairs are computed separately per stratum and then added.
`data`	A data.frame used to interpret variables named in the formula. Only applicable if object is a formula.
`newdata`	Optional new data frame for evaluating models.
`weights`	Optional vector of case weights, applicable only if object is a formula.
`subset`	An expression indicating which subset of the rows of data should be used in the fit. Only applicable if object is a formula.
`na.action`	A missing-data filter function, applied to the model.frame after any subset argument has been used. Default is options()$na.action. Only applicable if object is a formula.
`cluster`	Optional grouping vector for calculating robust variance.
`ymin`	Optional lower bound for y values in the calculation.

ymax Optional upper bound for y values in the calculation.

timewt Weighting to be applied. The overall statistic is a weighted mean over event times, with options like "n", "S", "S/G", "n/G2", "I".

influence Level of influence output to return, where 1=return the dfbeta vector, 2=return the full influence matrix, 3=return both.

ranks If TRUE, returns a data frame with ranks.

reverse If TRUE, assume larger x values predict smaller response values y.

timefix If TRUE, correct for possible rounding errors related to tied times.

keepstrata Either TRUE, FALSE, or an integer value. Computations are always done within stratum, then added. If the total number of strata is greater than keepstrata, or keepstrata=FALSE, those subtotals are not kept in the output.

Value

An object of class ’concordance’ containing the estimated concordance value or values, counts of concordant, discordant, and tied pairs, number of observations, estimated variance of the concordance, and optionally, the data frame of ranks, dfbeta vector, and influence matrix.

References

F Harrell, R Califf, D Pryor, K Lee and R Rosati, Evaluating the yield of medical tests, J Am Medical Assoc, 1982.

R Peto and J Peto, Asymptotically efficient rank invariant test procedures (with discussion), J Royal Stat Soc A, 1972.

M Schemper, Cox analysis of survival data with non-proportional hazard functions, The Statistician, 1992.

H Uno, T Cai, M Pencina, R D’Agnostino and Lj Wei, On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data, Statistics in Medicine, 2011.

Therneau, T. M., Lumley, T., Atkinson, E., Crowson, C. (2024). survival: Survival Analysis. R package version 3.7-0. DOI: doi:10.32614/CRAN.package.survival. Available at https://CRAN.R-project.org/package=survival.

Examples

```
library(survival)
library(dplyr)

# Load the pbc dataset
data(pbc)

# Data preparation
pbc <- pbc %>%
  filter(!is.na(trt)) %>%
  mutate(
    log_albumin = log(albumin),
    log_bili = log(bili),
    log_protime = log(protime),
    status = ifelse(status == 2, 1, 0)
  )

# Fit a Cox Proportional Hazards model
```

```
cox_model <- cph(
  Surv(time, status) ~ age + log_albumin + log_bili + log_protime + edema,
  data = pbc,
  x = TRUE,
  y = TRUE
)

# Compute the concordance statistic
concordance_result <- pam.concordance(cox_model)
print(concordance_result)
```

pam.concordance_metric

Compute the Concordance Statistic for Survival Data

Description

This function calculates the Concordance Index (C-index), a performance metric used to evaluate survival prediction models. The C-index measures the ability of a model to correctly rank predicted survival times with observed survival outcomes.

Usage

```
pam.concordance_metric(
  predicted_time,
  survival_time,
  status,
  weight = "H",
  input_tau = NULL
)
```

Arguments

predicted_time Numeric vector of predicted survival times.

survival_time Numeric vector of observed survival times.

status Numeric vector indicating censoring status (1 = event occurred, 0 = censored).

weight Character string specifying the weighting method to use. Options are: - ‘"H"‘: Harrell’s C-index. - ‘"U"‘: Unweighted C-index with KM survival probability. - ‘"U_tau"‘: Unweighted C-index truncated at input_tau.

input_tau Optional numeric value specifying the truncation point for "U_tau" weighting.

Details

The function performs pairwise comparisons of predicted survival times and observed outcomes. For weighted calculations (‘"U"‘ and ‘"U_tau"‘), the Kaplan-Meier survival probability is used to weight the pairs. The ‘"U_tau"‘ method additionally truncates comparisons at a specified time point (input_tau).

Value

A numeric value representing the computed C-index. A value closer to 1 indicates better model performance, while a value closer to 0.5 indicates no discriminative power.

References

F Harrell, R Califf, D Pryor, K Lee and R Rosati, Evaluating the yield of medical tests, J Am Medical Assoc, 1982.

R Peto and J Peto, Asymptotically efficient rank invariant test procedures (with discussion), J Royal Stat Soc A, 1972.

M Schemper, Cox analysis of survival data with non-proportional hazard functions, The Statistician, 1992.

H Uno, T Cai, M Pencina, R D'Agnostino and Lj Wei, On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data, Statistics in Medicine, 2011.

Therneau, T. M., Lumley, T., Atkinson, E., Crowson, C. (2024). survival: Survival Analysis. R package version 3.7-0. DOI: doi:10.32614/CRAN.package.survival. Available at `https://CRAN.R-project.org/package=survival`.

Examples

```
library(PAmeasure)
predicted_time <- c(2.5, 3.2, 1.8, 4.1)
survival_time <- c(3, 4, 2, 5)
status <- c(1, 0, 1, 1)
pam.concordance_metric(predicted_time, survival_time, status, weight = "H")
```

`pam.coxph` *Prediction Accuracy Measures for Cox proportional hazards model*

Description

This function calculates a pair of measures, R-Squared and L-Squared, for Cox proportional hazards model. R-squared is an extension of the classical R2 statistic for a linear model, quantifying the amount of variability in the response that is explained by a corrected prediction based on the original prediction function. L-squared is the proportion of the prediction error of the original prediction function that is explained by the corrected prediction function, quantifying the distance between the corrected and uncorrected predictions. When used together, they give a complete summary of the predictive power of a prediction function.

Usage

```
pam.coxph(fit.cox)
```

Arguments

`fit.cox`	object inheriting from class coxph representing a fitted Cox proportional hazards regression model. Specifying x = TRUE and y=TRUE are required in the call to coxph( ) to include the design matrix and the response vector in the object fit.

Value

A list containing two components: R-squared and L-squared

References

Li, G., & Wang, X. (2016). Prediction Accuracy Measures for a Nonlinear Model and for Right-Censored Time-to-Event Data. arXiv preprint arXiv:1611.03063. Available at https://arxiv.org/abs/1611.03063

Examples

```
library(survival)
library(PAmeasures)

# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)

head(pbc)


# Fit a univariate Cox PH model with standardised blood clotting time
fit1 <- coxph(Surv(time, status==2) ~ protime, data = pbc,x=TRUE,y=TRUE)

# R.squared and L.squared of Cox PH model
pam.coxph(fit1)

# Fit a multiple Cox PH model with bilirubin and standardised blood clotting time
fit2 <- coxph(Surv(time, status==2) ~ bili + protime, data = pbc,x=TRUE,y=TRUE)

# R.squared and L.squared of Cox PH model
pam.coxph(fit2)
```

---

pam.metrics_summary_predicted *Summary of Performance Metrics for Predicted Survival Data*

Description

This function computes and summarizes various performance metrics for survival data using predicted values and observed survival data. Users can specify the desired metrics or compute all available metrics by default.

Usage

```
pam.metrics_summary_predicted(
  predicted_data,
  survival_time,
  metric,
  status = NULL,
  tau = NULL,
  t_star = NULL,
  start_time = NULL
)
```

Arguments

`predicted_data` A numeric vector of predicted survival probabilities or scores.

`survival_time` A numeric vector of observed survival times.

`status` A numeric or logical vector indicating event status (1 for event, 0 for censoring).

`tau` An optional numeric value for restricted time horizon. Default is NULL.

`t_star` An optional numeric value specifying the time point for certain metrics. Default is NULL.

`metrics` A character vector specifying the desired metrics to compute. Options include: - ‘"R_square"‘: \(R^2 \) metric. - ‘"L_square"‘: \(L^2 \) metric. - ‘"Pesudo_R"‘: Pseudo-\(R^2 \) metric. - ‘"Harrell’s C"‘: Harrell’s Concordance Index. - ‘"Uno’s C"‘: Uno’s Concordance Index. - ‘"R_sph"‘: Explained variation (\(R_sph \)). - ‘"R_sh"‘: Explained variation (\(R_sh \)). - ‘"Brier Score"‘: Brier Score. - ‘"Time Dependent Auc"‘: Time-dependent AUC.

Value

A data frame summarizing the requested performance metrics.

Examples

```
predicted_data <- c(0.8, 0.6, 0.4, 0.2)
survival_time <- c(5, 8, 3, 10)
status <- c(1, 0, 1, 1)
metric <- "R_square"
pam.metrics_summary_predicted(predicted_data, survival_time, metric, status)
```

`pam.nlm` *Prediction Accuracy Measures for Nonlinear Regression Models.*

Description

This function calculates a pair of measures, R-Squared and L-Squared, for any nonlinear regression model. R-squared is an extension of the classical R2 statistic for a linear model, quantifying the amount of variability in the response that is explained by a corrected prediction based on the original prediction function. L-squared is the proportion of the prediction error of the original prediction function that is explained by the corrected prediction function, quantifying the distance between the corrected and uncorrected predictions. When used together, they give a complete summary of the predictive power of a prediction function.

Usage

```
pam.nlm(y, y.predict)
```

Arguments

`y` A numeric vector containing the response values.

`y.predict` A numeric vector containing the predicted response values from a fitted model.

Value

A list containing two components: R-squared and L-squared

Examples

```
library(PAmeasures)

data(moore)

head(moore)

# Transistor count
count <- moore$count

time<-moore$time

# Fit a log-linear model
moore.glm= glm(log2(count) ~ time, family=gaussian(link = "identity") )

# Obtain predicted transistor count
count.predict<-2^(predict(moore.glm,newdata = data.frame(X = time),type = "response" ))

# R.squared and L.squared of log-linear model
pam.nlm(count, count.predict)
```

pam.performance_metrics *Performance Metrics for Survival Analysis Models*

Description

This function computes a comprehensive set of performance metrics for survival analysis models. It provides metrics such as R_square, L_square, Pseudo_R, Harrell's C, Uno's C, R_sph (distance-based estimator for survival predictive accuracy), R_sh, Brier Score, and Time-dependent AUC. Users can specify particular metrics and model types, enabling tailored performance evaluation for various survival models.

Usage

```
pam.performance_metrics(
  data,
  time_var,
  status_var,
  covariates,
  model = "coxph",
  metrics = "all",
  newdata = NULL
)
```

Arguments

`data` A data frame containing the survival data.

`time_var` The name of the time variable in ‘data‘ indicating survival time.

`status_var` The name of the status variable in ‘data‘ indicating event occurrence.

`covariates` A character vector of covariate names to include in the model.

`model` A character string or vector specifying the model types to fit (e.g., "coxph", "exp", "lognormal", "weibull"). Default is "coxph" to fit all models.

`metrics` A character string or vector specifying the metrics to compute. Default is "all" to compute all available metrics.

`newdata` (Optional) A data frame containing validation data. If ‘NULL‘, the function uses the same data as ‘data‘ for model evaluation.

Value

A data frame containing the selected model’s performance metrics.

Examples

```
library(PAmeasures)
library(survival)
library(rms)
library(dplyr)

Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)
pbc <- pbc %>%
  filter(is.na(trt)==F) %>%
mutate(log_albumin = log(albumin),
       log_bili = log(bili),
       log_protime = log(protime),
       status = ifelse(status==2, 1, 0))
time_var <- "time"
status_var <- "status"
covariates <- c("age", "log_albumin", "log_bili", "log_protime", "edema")
Call the function with all metrics and all models
results <- pam.performance_metrics(data = pbc, time_var = time_var, status_var = status_var, covariates = covar
results2 <- pam.performance_metrics(data = pbc, time_var = time_var, status_var = status_var, covariates = cova
```

`pam.r2_metrics` *Compute R^2, L^2, and Pseudo-R^2 for Survival Data*

Description

This function calculates \(R^2 \), \(L^2 \), and pseudo-\(R^2 \) for survival data using observed and predicted survival times.

Usage

```
pam.r2_metrics(predicted_data, survival_time, status, tau = NULL)
```

Arguments

`predicted_data`	A numeric vector of predicted survival times.
`survival_time`	A numeric vector of observed survival times.
`status`	A numeric vector indicating the event occurrence (1 for event, 0 for censoring).
`tau`	An optional numeric value for restricted time horizon. Default is NULL (no restriction).

Value

A list with \(R^2 \), \(L^2 \), and pseudo-\(R^2 \).

Examples

```
library(PAmeasure)
predicted_data <- c(5, 4, 8, 2)
survival_time <- c(6, 5, 10, 3)
status <- c(1, 1, 0, 1)
pam.r2_metrics(predicted_data, survival_time, status)
```

---

pam.rsh_metric *Compute R_sh Metric*

Description

This function calculates the \(R_sh \) metric, a measure of explained variation for survival models, using only predicted survival probabilities and observed survival data.

Usage

```
pam.rsh_metric(predicted_data, survival_time, status)
```

Arguments

`predicted_data`	A numeric vector of predicted survival probabilities.
`survival_time`	A numeric vector of observed survival times.
`status`	A numeric vector indicating event occurrence (1 for event, 0 for censoring).

Value

A list with the following components:

`D`	The total variation in the survival data.
`Dx`	The unexplained variation by the predictions.
`V`	The explained variation (\(R_sh \)).

References

Schemper, M. and R. Henderson (2000). Predictive accuracy and explained variation in Cox regression. Biometrics 56, 249–255.

Lusa, L., R. Miceli and L. Mariani (2007). Estimation of predictive accuracy in survival analysis using R and S-PLUS. Computer Methods and Programs in Biomedicine 87, 132–137.

Potapov, S., Adler, W., Schmid, M., Bertrand, F. (2024). survAUC: Estimating Time-Dependent AUC for Censored Survival Data. R package version 1.3-0. DOI: doi:10.32614/CRAN.package.survAUC. Available at `https://CRAN.R-project.org/package=survAUC`.

Examples

```
library(PAmeasure)
predicted_data <- c(0.8, 0.6, 0.4, 0.2)
survival_time <- c(5, 8, 3, 10)
status <- c(1, 0, 1, 1)
pam.rsh_metric(predicted_data, survival_time, status)
```

`pam.rsph` *RE Measure of Explained Variation of the Model*

Description

This package provides tools to evaluate the proportion of variability in survival times explained by covariates using Schempers' RE measure. The measure assesses how well the model explains the variability in the data, which is important for understanding the impact of covariates in survival analysis.

The RE measure is sourced from: `https://ibmi3.mf.uni-lj.si/ibmi-english/biostat-center/programje/Re.r`

It includes the following functions:

- `re` - Function for calculating the RE measure.
- `summary.re` - Function for calculating the RE measure over time (cumulative and in specified time windows).

The `re` function can be called with:

`re(fit)`

It works with various model objects, including `coxph`, `survreg`, and `aareg`.

Usage

```
pam.rsph(fit, ...)
```

Arguments

`fit`	A survival model object (e.g., `coxph`, `survreg`, `aareg`).
`...`	Additional arguments passed to methods.

Details

The re function calculates the RE measure, which consists of the following components: - **Re**: The primary RE measure. - **Re.imp**: The RE measure assuming constant coefficients after the last event time. - **Re.fix**: The RE measure assuming a constant value of coefficients throughout the follow-up time. (Note: Returns NA for counting process type times.) - **se**: Standard error of the RE measure. - **sen0**: Standard error calculated with expected Q. - **C**: C-index generalized to allow for time-dependent covariates, ensuring independence from the censoring process. - **r2nw**: An unweighted measure.

Value

A list containing the various RE measures and their associated statistics.

Note

Left-truncated data is not yet implemented. When calculating Re.imp, the dataset should not be split after the last event time; each data line should represent a different subject.

References

- Schemper, M. and R. Henderson (2000). Predictive accuracy and explained variation in Cox regression. Biometrics 56, 249–255. - LUSA, L., MICELI, R. and MARIANI, L. (2007). Estimation of predictive accuracy in survival analysis using R and S-PLUS. Computer methods and programs in biomedicine 87 132–137.

Examples

```
# Load necessary libraries
library(survival)
library(PAmeasures)
# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)

pbc <- pbc %>%
  filter(!is.na(trt)) %>%
  mutate(log_albumin = log(albumin),
         log_bili = log(bili),
         log_protime = log(protime),
         status = ifelse(status == 2, 1, 0))

# Fit a full Cox PH model
fit.coxph.full <- coxph(Surv(time, status) ~ age + log_albumin +
                         log_bili + log_protime + edema,
                         data = pbc, x = TRUE, y = TRUE)
# Calculate the RE measure
re_result <- pam.re(fit.coxph.full)
print(re_result$Re)

# Summarize the RE measure over time
summary_result <- pam.summary(re_result)
```

pam.rsph_metric *RE Measure of Explained Variation Using Predicted and Observed Data*

Description

This function computes the RE (explained variation) metric using predicted risk scores and observed survival times. It evaluates the proportion of variability in survival times explained by the covariates using a simplified approach that requires only the predicted values and observed survival data.

Usage

```
pam.rsph_metric(predicted_data, survival_time, status, start_time = NULL)
```

Arguments

`predicted_data`	A numeric vector of predicted risk scores or linear predictors from a model.
`survival_time`	A numeric vector of observed survival times.
`status`	A numeric or logical vector indicating event status (1 for event, 0 for censoring).

Details

This function assumes the input data is complete and properly formatted. It adjusts for censoring using Kaplan-Meier weights and computes ranks for the predicted values to evaluate the explained variation in survival times.

Value

A list containing the following components:

- `Re`: The RE measure, representing the proportion of explained variation.
- `numerator`: The numerator of the RE calculation, representing explained variation.
- `denominator`: The denominator of the RE calculation, representing total variation.

References

- Schemper, M., & Henderson, R. (2000). Predictive accuracy and explained variation in Cox regression. Biometrics, 56, 249–255. - Lusa, L., Miceli, R., & Mariani, L. (2007). Estimation of predictive accuracy in survival analysis using R and S-PLUS. Computer Methods and Programs in Biomedicine, 87, 132–137.

Examples

```
library(survival)

data("lung")

predicted_data <- lung$ph.ecog
survival_time <- lung$time
status <- lung$status - 1

complete_cases <- complete.cases(predicted_data, survival_time, status)
```

```
predicted_data <- predicted_data[complete_cases]
survival_time <- survival_time[complete_cases]
status <- status[complete_cases]

# Compute the RE measure
result <- pam.rsph_metricc(predicted_data, survival_time, status)

cat("RE Measure:", result$Re, "\n")
```

pam.schemper *Distance-based estimator of survival predictive accuracy proposed by Schemper and Henderson*

Description

This function calculates metrics (D, Dx, V) based on a fitted Cox model using the ‘cph‘ function from the ‘rms‘ package. It uses survival estimates to assess the concordance between observed and predicted survival outcomes.

Usage

```
pam.schemper(train.fit, traindata, newdata)
```

Arguments

`train.fit`	A fitted Cox model from ‘rms::cph‘.
`traindata`	A data frame containing training data.
`newdata`	A data frame with new data for prediction.

Value

A list containing the model call, D, Dx, V metrics, and other calculated components.

References

Schemper, M. and R. Henderson (2000). Predictive accuracy and explained variation in Cox regression. Biometrics 56, 249–255.

Lusa, L., R. Miceli and L. Mariani (2007). Estimation of predictive accuracy in survival analysis using R and S-PLUS. Computer Methods and Programs in Biomedicine 87, 132–137.

Potapov, S., Adler, W., Schmid, M., Bertrand, F. (2024). survAUC: Estimating Time-Dependent AUC for Censored Survival Data. R package version 1.3-0. DOI: doi:10.32614/CRAN.package.survAUC. Available at https://CRAN.R-project.org/package=survAUC.

Examples

```
library(PAmeasures)
library(survival)

# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)
pbc <- pbc %>%
  filter(is.na(trt)==F) %>%
mutate(log_albumin = log(albumin),
       log_bili = log(bili),
       log_protime = log(protime),
       status = ifelse(status==2, 1, 0))
#Schemper and Henderson's estimator of the absolute deviation between survival functions
schemper(train.fit.full, pbc, pbc)$Dx
```

pam.survivalROC *Time-dependent ROC Curve from Censored Survival Data*

Description

This function creates a time-dependent ROC curve from censored survival data using the Kaplan-Meier (KM) or Nearest Neighbor Estimation (NNE) method of Heagerty, Lumley, and Pepe (2000).

Usage

```
pam.survivalROC(
  Stime,
  status,
  marker,
  entry = NULL,
  predict.time,
  cut.values = NULL,
  method = "NNE",
  lambda = NULL,
  span = NULL,
  window = "symmetric"
)
```

Arguments

`Stime`	Event time or censoring time for subjects.
`status`	Indicator of status, 1 if death or event, 0 otherwise.
`marker`	Predictor or marker value.
`entry`	Entry time for the subjects.
`predict.time`	Time point of the ROC curve.
`cut.values`	Marker values to use as a cut-off for calculating sensitivity and specificity.
`method`	Method for fitting joint distribution of (marker, t), either "KM" or "NNE". Default is "NNE".
`lambda`	Smoothing parameter for NNE.
`span`	Span for the NNE. Either `lambda` or `span` is required for NNE.
`window`	Window type for NNE, either "symmetric" or "asymmetric".

Value

A list containing the following elements:

cut.values Unique marker values for calculating TP and FP.

TP True Positive corresponding to the cut-offs in the marker.

FP False Positive corresponding to the cut-offs in the marker.

predict.time Time point of interest.

Survival Kaplan-Meier survival estimate at `predict.time`.

AUC Area Under the ROC Curve at `predict.time`.

References

Heagerty, P. J., Lumley, T., & Pepe, M. S. (2000). Time-dependent ROC curves for censored survival data and a diagnostic marker. *Biometrics*, 56(2), 337-344.

Heagerty, P. J., Saha-Chaudhuri, P. (2022). survivalROC: Time-Dependent ROC Curve Estimation from Censored Survival Data. R package version 1.0.3.1. DOI: doi:10.32614/CRAN.package.survivalROC. Available at `https://CRAN.R-project.org/package=survivalROC`.

Examples

```
library(survival)
library(PAmeasures)
# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)
pbc <- pbc %>%
  filter(is.na(trt)==F) %>%
  mutate(log_albumin = log(albumin),
         log_bili = log(bili),
         log_protime = log(protime),
         status = ifelse(status==2, 1, 0))

time_dep_auc_full <- survivalROC(Stime = pbc$time,
            status = pbc$status,
            marker = predict(fit.coxph.full, newdata=pbc, type = "lp") ,
            predict.time = quantile(pbc$time, 0.5),
            method="KM")$AUC
```

`pam.survreg` *Prediction Accuracy Measures for Parametric Survival Regression Models*

Description

This function calculates a pair of measures, R-Squared and L-Squared, for parametric survival regression models. R-squared is an extension of the classical R2 statistic for a linear model, quantifying the amount of variability in the response that is explained by a corrected prediction based on the original prediction function. L-squared is the proportion of the prediction error of the original prediction function that is explained by the corrected prediction function, quantifying the distance between the corrected and uncorrected predictions. When used together, they give a complete summary of the predictive power of a prediction function.

Usage

```
pam.survreg(fit.survreg, validation_data = NULL)
```

Arguments

`fit.survreg`	object inheriting from class survreg representing a fitted parametric survival regression model. Specifying x = TRUE and y=TRUE are required in the call to survreg( ) to include the design matrix and the response vector in the object fit.

Value

A list containing two components: R-squared and L-squared

Examples

```
library(survival)
library(PAmeasures)

# Use Mayo Clinic Primary Biliary Cirrhosis Data
data(pbc)

head(pbc)

# Fit an exponential model with bilirubin
fit1 <- survreg(Surv(time, status==2) ~ bili, data = pbc,dist="exponential",x=TRUE,y=TRUE)

# R.squared and L.squared of exponential model
pam.survreg(fit1)

# Fit a lognormal model with standardised blood clotting time
fit2 <- survreg(Surv(time, status==2) ~ protime, data = pbc,dist="lognormal",x=TRUE,y=TRUE)

# R.squared and L.squared of lognormal model
pam.survreg(fit2)

# Fit a weibull model with bilirubin and standardised blood clotting time
fit3 <- survreg(Surv(time, status==2) ~ bili + protime, data = pbc,dist="weibull",x=TRUE,y=TRUE)

# R.squared and L.squared of weibull model
pam.survreg(fit3)
```

Index