Package: YPPE (via r-universe)

August 27, 2024

Title Yang and Prentice Model with Piecewise Exponential Baseline Distribution

Version 1.1.0

Description Semiparametric modeling of lifetime data with crossing survival curves via Yang and Prentice model with piecewise exponential baseline distribution. Details about the model can be found in Demarqui and Mayrink (2021) <doi:10.1214/20-BJPS471>. Model fitting carried out via likelihood-based and Bayesian approaches. The package also provides point and interval estimation for the crossing survival times.

License MIT + file LICENSE

URL https://fndemarqui.github.io/YPPE/

BugReports https://github.com/fndemarqui/YPPE/issues

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.4.0), survival

Imports dplyr, methods, MASS, Formula, Rcpp (>= 0.12.0), Rdpack, rstan (>= 2.18.1), rstantools (>= 2.0.0)

RdMacros Rdpack

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

RoxygenNote 7.2.3

Suggests rmarkdown, knitr, testthat

VignetteBuilder knitr

Repository https://fndemarqui.r-universe.dev

RemoteUrl https://github.com/fndemarqui/yppe

RemoteRef HEAD

RemoteSha e733475087c16d502586e3c410b34d6f44fb04a8

Contents

`YPPE-package` *The 'YPPE' package.*

Description

Semiparametric modeling of lifetime data with crossing survival curves via Yang and Prentice model with piecewise exponential baseline distribution. Details about the model can be found in (Demarqui and Mayrink 2021) <doi.org/10.1214/20-BJPS471>. Model fitting carried out via likelihood-based and Bayesian approaches. The package also provides point and interval estimation for the crossing survival times.

References

Demarqui FN, Mayrink VD (2021). "Yang and Prentice model with piecewise exponential baseline distribution for modeling lifetime data with crossing survival curves." *Brazilian Journal of Probability and Statistics*, **35**(1), 172 – 186. doi:10.1214/20BJPS471.

Yang S, Prentice RL (2005). "Semiparametric analysis of short-term and long-term hazard ratios with two-sample survival data." *Biometrika*, **92**(1), 1-17.

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2. https://mc-stan.org

---

`AIC.yppe` *Akaike information criterion*

Description

Akaike information criterion

Usage

```
## S3 method for class 'yppe'
AIC(object, ..., k = 2)
```

Arguments

`object`	an object of the class yppe.
`...`	further arguments passed to or from other methods.
`k`	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.

Value

the Akaike information criterion

`anova.yppe` *anova method for yppe models*

Description

Compute analysis of variance (or deviance) tables for one or more fitted model objects.

Usage

```
## S3 method for class 'yppe'
anova(...)
```

Arguments

...	further arguments passed to or from other methods.

Value

the ANOVA table.

---

coef.yppe *Estimated regression coefficients*

Description

This function returns the estimated regression coefficients when the maximum likelihood estimation approach is used in the model fitting.

Usage

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

Arguments

object	an object of the class yppe.
...	further arguments passed to or from other methods.

Value

the estimated regression coefficients.

Examples

```
## Not run:
fit <- yppe(Surv(time, status)~arm, data=ipass, init = 0)
coef(fit)

## End(Not run)
```

confint.yppe *Confidence intervals for the regression coefficients*

Description

Confidence intervals for the regression coefficients

Usage

```
## S3 method for class 'yppe'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	an object of the class yppe.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	further arguments passed to or from other methods.

Value

100(1-alpha) confidence intervals for the regression coefficients.

crossTime *Generic S3 method crossTime*

Description

Generic S3 method crossTime

Usage

```
crossTime(object, ...)
```

Arguments

object	a fitted model object
...	further arguments passed to or from other methods.

Value

the crossing survival time

`crossTime.yppe` *Computes the crossing survival times*

Description

Computes the crossing survival times

Usage

```
## S3 method for class 'yppe'
crossTime(object, newdata1, newdata2, conf.level = 0.95, nboot = 1000, ...)
```

Arguments

`object`	an object of class yppe
`newdata1`	a data frame containing the first set of explanatory variables
`newdata2`	a data frame containing the second set of explanatory variables
`conf.level`	level of the confidence/credible intervals
`nboot`	number of bootstrap samples (default nboot=1000); ignored if approach="bayes".
`...`	further arguments passed to or from other methods.

Value

the crossing survival time

Examples

```
# ML approach:
library(YPPE)
mle <- yppe(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle", init = 0)
summary(mle)
newdata1 <- data.frame(arm=0)
newdata2 <- data.frame(arm=1)
tcross <- crossTime(mle, newdata1, newdata2, nboot = 10)
tcross
ekm <- survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(mle, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))
abline(v=tcross, col="blue")

# Bayesian approach:
bayes <- yppe(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes", chains=1, iter=10)
summary(bayes)
newdata1 <- data.frame(arm=0)
```

```
newdata2 <- data.frame(arm=1)
tcross <- crossTime(bayes, newdata1, newdata2)
tcross
ekm <- survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(bayes, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))
abline(v=tcross, col="blue")
```

---

gastric *Gastric cancer data set*

Description

Data set from a clinical trial conducted by the Gastrointestinal Tumor Study Group (GTSG) in 1982. The data set refers to the survival times of patients with locally nonresectable gastric cancer. Patients were either treated with chemotherapy combined with radiation or chemotherapy alone.

Format

A data frame with 90 rows and 3 variables:

- time: survival times (in days)
- status: failure indicator (1 - failure; 0 - otherwise)
- trt: treatments (1 - chemotherapy + radiation; 0 - chemotherapy alone)

Author(s)

Fabio N. Demarqui <fndemarqui@est.ufmg.br>

References

Gastrointestinal Tumor Study Group. (1982) A Comparison of Combination Chemotherapy and Combined Modality Therapy for Locally Advanced Gastric Carcinoma. Cancer 49:1771-7.

ipass *IRESSA Pan-Asia Study (IPASS) data set*

Description

Reconstructed IPASS clinical trial data reported in Argyropoulos and Unruh (2015). Although reconstructed, this data set preserves all features exhibited in references with full access to the observations from this clinical trial. The data base is related to the period of March 2006 to April 2008. The main purpose of the study is to compare the drug gefitinib against carboplatin/paclitaxel doublet chemotherapy as first line treatment, in terms of progression free survival (in months), to be applied to selected non-small-cell lung cancer (NSCLC) patients.

Format

A data frame with 1217 rows and 3 variables:

- time: progression free survival (in months)
- status: failure indicator (1 - failure; 0 - otherwise)
- arm: (1 - gefitinib; 0 - carboplatin/paclitaxel doublet chemotherapy)

Author(s)

Fabio N. Demarqui <fndemarqui@est.ufmg.br>

References

Argyropoulos, C. and Unruh, M. L. (2015). Analysis of time to event outcomes in randomized controlled trials by generalized additive models. PLOS One 10, 1-33.

logLik.yppe *Extract Log-Likelihood*

Description

Extract Log-Likelihood

Usage

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

Arguments

object	an object of the class yppe.
...	further arguments passed to or from other methods.

Value

the log-likelihood associated with the fitted model.

`model.matrix.yppe` *Model.matrix method for yppe models*

Description

Reconstruct the model matrix (or matrices if the alternative formulation of the YP model is used) for a yppe model.

Usage

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

Arguments

`object`	an object of the class yppe.
`...`	further arguments passed to or from other methods.

Value

The model matrix (or matrices) for the fit.

Examples

```
## Not run:
fit <- yppe(Surv(time, status)~arm, data=ipass)
model.matrix(fit)

## End(Not run)
```

`pehaz` *Hazard and cumulative hazard functions of the PE distribution*

Description

Hazard and cumulative hazard functions of the PE distribution

Usage

```
hpexp(x, rho, rates)

Hpexp(x, rho, rates)
```

Arguments

x	vector of time points.
rho	vector of time grid knots.
rates	vector of failure rates.

Value

hpexp gives the hazard function and Hpexp gives the cumulative hazard function of the PE distribution.

---

pexp *Probability function, distribution function, quantile function and random generation for the Piecewise Exponential (PE) distribution.*

Description

Probability function, distribution function, quantile function and random generation for the Piecewise Exponential (PE) distribution.

Usage

```
dpexp(x, rho, rates, log = FALSE)

ppexp(q, rho, rates, lower.tail = TRUE, log.p = FALSE)

qpexp(p, rho, rates, lower.tail = TRUE, log.p = FALSE)

rpexp(n, rho, rates)
```

Arguments

x	vector of time points.
rho	vector of time grid knots.
rates	vector of failure rates.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
q	vector of quantiles.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$; otherwise, $P[X > x]$.
p	vector of probabilities.
n	number of random values to return.

Value

dpexp gives the (log) probability function, ppexp gives the (log) distribution function, qpexp gives the quantile function, and rpexp generates random deviates.

Examples

```
n <- 10
rho <- c(0, 1, 3, 7, Inf)
rates <- c(0.5, 4, 0.8, 0.1)
x <- sort(rpexp(n, rho=rho, rates=rates))
Fx <- ppexp(x, rho, rates)
y <- qpexp(Fx, rho, rates)
# checking:
x==y
```

---

phpe *phpe: Fit Proportional Hazards Regression Model with Piecewise Exponential baseline distribution.*

Description

phpe: Fit Proportional Hazards Regression Model with Piecewise Exponential baseline distribution.

Usage

```
phpe(
  formula,
  data,
  n_int = NULL,
  rho = NULL,
  tau = NULL,
  hessian = TRUE,
  approach = c("mle", "bayes"),
  hyper_parms = list(h1_gamma = 0, h2_gamma = 4, mu_beta = 0, sigma_beta = 4),
  ...
)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which phpe is called.
n_int	number of intervals of the PE distribution. If NULL, default value (square root of n) is used.
rho	the time grid of the PE distribution. If NULL, the function timeGrid is used to compute rho.
tau	the maximum time of follow-up. If NULL, tau = max(time), where time is the vector of observed survival times.

hessian	logical; If TRUE (default), the hessian matrix is returned when approach="mle".
approach	approach to be used to fit the model (mle: maximum likelihood; bayes: Bayesian approach).
hyper_parms	a list containing the hyper-parameters of the prior distributions (when approach = "bayes"). If not specified, default values are used.
...	Arguments passed to either ‘rstan::optimizing‘ or ‘rstan::sampling‘ .

Value

phpe returns an object of class "phpe" containing the fitted model.

Examples

```
# ML approach:
library(YPPE)
mle <- phpe(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle")
summary(mle)

# Bayesian approach:
bayes <- phpe(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes")
summary(bayes)
```

---

pope *pope: Fit Proportional Odds Regression Model with Piecewise Exponential baseline distribution.*

Description

pope: Fit Proportional Odds Regression Model with Piecewise Exponential baseline distribution.

Usage

```
pope(
  formula,
  data,
  n_int = NULL,
  rho = NULL,
  tau = NULL,
  hessian = TRUE,
  approach = c("mle", "bayes"),
  hyper_parms = list(h1_gamma = 0, h2_gamma = 4, mu_beta = 0, sigma_beta = 4),
  ...
)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which pope is called.
n_int	number of intervals of the PE distribution. If NULL, default value (square root of n) is used.
rho	the time grid of the PE distribution. If NULL, the function timeGrid is used to compute rho.
tau	the maximum time of follow-up. If NULL, tau = max(time), where time is the vector of observed survival times.
hessian	logical; If TRUE (default), the hessian matrix is returned when approach="mle".
approach	approach to be used to fit the model (mle: maximum likelihood; bayes: Bayesian approach).
hyper_parms	a list containing the hyper-parameters of the prior distributions (when approach = "bayes"). If not specified, default values are used.
...	Arguments passed to either ‘rstan::optimizing‘ or ‘rstan::sampling‘ .

Value

pope returns an object of class "pope" containing the fitted model.

Examples

```
# ML approach:
library(YPPE)
mle <- pope(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle")
summary(mle)

# Bayesian approach:
bayes <- pope(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes")
summary(bayes)
```

---

print.summary.phpe *Print the summary.phpe output*

Description

Print the summary.phpe output

Usage

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

Arguments

x	an object of the class summary.phpe.
...	further arguments passed to or from other methods.

Value

a summary of the fitted model.

`print.summary.pope` *Print the summary.pope output*

Description

Print the summary.pope output

Usage

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

Arguments

x	an object of the class summary.pope.
...	further arguments passed to or from other methods.

Value

a summary of the fitted model.

print.summary.yppe *Print the summary.yppe output*

Description

Print the summary.yppe output

Usage

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

Arguments

x	an object of the class summary.yppe.
...	further arguments passed to or from other methods.

Value

a summary of the fitted model.

rates *Generic S3 method rates*

Description

Generic S3 method rates

Usage

```
rates(object, ...)
```

Arguments

object	a fitted model object.
...	further arguments passed to or from other methods.

Details

Method only available for ML approach.

Value

the estimated failure rates for the PE distribution.

rates.phpe *Estimated failure rates for the PE distribution*

Description

Estimated failure rates for the PE distribution

Usage

```
## S3 method for class 'phpe'
rates(object, ...)
```

Arguments

object	a fitted model object.
...	further arguments passed to or from other methods.

Details

Method only available for ML approach.

Value

the estimated failure rates for the PE distribution.

rates.pope *Estimated failure rates for the PE distribution*

Description

Estimated failure rates for the PE distribution

Usage

```
## S3 method for class 'pope'
rates(object, ...)
```

Arguments

object	a fitted model object.
...	further arguments passed to or from other methods.

Details

Method only available for ML approach.

Value

the estimated failure rates for the PE distribution.

`rates.yppe` *Estimated failure rates for the PE distribution*

Description

Estimated failure rates for the PE distribution

Usage

```
## S3 method for class 'yppe'
rates(object, ...)
```

Arguments

`object`	a fitted model object.
`...`	further arguments passed to or from other methods.

Details

Method only available for ML approach.

Value

the estimated failure rates for the PE distribution.

`rsurv` *Random generation of survival data*

Description

Function to generate a random sample of survival data.

Usage

```
rsurv(
  formula,
  covariates,
  baseline = "weibull",
  gamma,
  psi = NULL,
  phi = NULL,
  max_fu
)
```

Arguments

formula	formula specifying the linear predictors
covariates	data frame containing the covariates used to generate the survival times
baseline	baseline model (currently only the Weibull distribution is available)
gamma	baseline parameters
psi	short-term regression coefficients
phi	long-term regression coefficients
max_fu	maximum follow-up time

---

summary.phpe *Summary for the yppe model*

Description

Summary for the yppe model

Usage

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

Arguments

object	an objecto of the class 'yppe'.
...	further arguments passed to or from other methods.

---

summary.pope *Summary for the yppe model*

Description

Summary for the yppe model

Usage

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

Arguments

object	an objecto of the class 'yppe'.
...	further arguments passed to or from other methods.

summary.yppe *Summary for the yppe model*

Description

Summary for the yppe model

Usage

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

Arguments

object	an objecto of the class 'yppe'.
...	further arguments passed to or from other methods.

survfit.phpe *survfit method for phpe models*

Description

Computes the predicted survivor function for a phpe model.

Usage

```
## S3 method for class 'phpe'
survfit(formula, newdata, ...)
```

Arguments

formula	an object of the class phpe
newdata	a data frame containing the set of explanatory variables.
...	further arguments passed to or from other methods.

Value

a list containing the estimated survival probabilities.

Examples

```
# ML approach:
library(YPPE)
mle <- phpe(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle", init = 0)
summary(mle)
ekm <- survival::survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(mle, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))

# Bayesian approach:
bayes <- phpe(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes")
summary(bayes)
ekm <- survival::survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(bayes, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))
```

---

survfit.pope *survfit method for pope models*

Description

Computes the predicted survivor function for a pope model.

Usage

```
## S3 method for class 'pope'
survfit(formula, newdata, ...)
```

Arguments

formula	an object of the class pope
newdata	a data frame containing the set of explanatory variables.
...	further arguments passed to or from other methods.

Value

a list containing the estimated survival probabilities.

Examples

```
# ML approach:
library(YPPE)
mle <- pope(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle", init = 0)
summary(mle)
ekm <- survival::survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(mle, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))

# Bayesian approach:
bayes <- pope(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes")
summary(bayes)
ekm <- survival::survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(bayes, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))
```

---

survfit.yppe *survfit method for yppe models*

Description

Computes the predicted survivor function for a yppe model.

Usage

```
## S3 method for class 'yppe'
survfit(formula, newdata, ...)
```

Arguments

formula	an object of the class yppe
newdata	a data frame containing the set of explanatory variables.
...	further arguments passed to or from other methods.

Value

a list containing the estimated survival probabilities.

Examples

```
# ML approach:
library(YPPE)
mle <- yppe(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle")
summary(mle)
ekm <- survival::survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(mle, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))

# Bayesian approach:
bayes <- yppe(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes")
summary(bayes)
ekm <- survival::survfit(Surv(time, status)~arm, data=ipass)
newdata <- data.frame(arm=0:1)
St <- survfit(bayes, newdata)
plot(ekm, col=1:2)
with(St, lines(time, surv[[1]]))
with(St, lines(time, surv[[2]], col=2))
```

---

timeGrid *Time grid*

Description

Time grid

Usage

```
timeGrid(time, status, n_int = NULL)
```

Arguments

time	Vector of failure times
status	Vector of failure indicators
n_int	Optional. Number of intervals. If NULL, the number of intervals is set to be equal to the number of distinct observed failure times.

Value

Time grid.

vcov.yppe *Variance-covariance matrix for a yppe model*

Description

This function extracts and returns the variance-covariance matrix associated with the regression coefficients when the maximum likelihood estimation approach is used in the model fitting.

Usage

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

Arguments

object	an object of the class yppe.
...	further arguments passed to or from other methods.

Value

the variance-covariance matrix associated with the regression coefficients.

yppe *yppe: Fit the Yang and Prentice Regression Model with Piecewise Exponential baseline distribution.*

Description

yppe: Fit the Yang and Prentice Regression Model with Piecewise Exponential baseline distribution.

Usage

```
yppe(
  formula,
  data,
  n_int = NULL,
  rho = NULL,
  tau = NULL,
  hessian = TRUE,
  approach = c("mle", "bayes"),
 hyper_parms = list(h1_gamma = 0, h2_gamma = 4, mu_psi = 0, sigma_psi = 4, mu_phi = 0,
    sigma_phi = 4, mu_beta = 0, sigma_beta = 4),
  ...
)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which yppe is called.
n_int	number of intervals of the PE distribution. If NULL, default value (square root of n) is used.
rho	the time grid of the PE distribution. If NULL, the function timeGrid is used to compute rho.
tau	the maximum time of follow-up. If NULL, tau = max(time), where time is the vector of observed survival times.
hessian	logical; If TRUE (default), the hessian matrix is returned when approach="mle".
approach	approach to be used to fit the model (mle: maximum likelihood; bayes: Bayesian approach).
hyper_parms	a list containing the hyper-parameters of the prior distributions (when approach = "bayes"). If not specified, default values are used.
...	Arguments passed to either ‘rstan::optimizing‘ or ‘rstan::sampling‘ .

Value

yppe returns an object of class "yppe" containing the fitted model.

Examples

```
# ML approach:
library(YPPE)
mle <- yppe(Surv(time, status)~arm, data=ipass, n_int=10, approach="mle")
summary(mle)

# Bayesian approach:
bayes <- yppe(Surv(time, status)~arm, data=ipass, n_int=10, approach="bayes")
summary(bayes)
```

Index