

Package: spsurv (via r-universe)

October 11, 2026

Type Package

Title Bernstein Polynomial Based Semiparametric Survival Analysis

Version 1.1.0

Description Semiparametric survival analysis based on Bernstein polynomials. 'spsurv' includes proportional hazards, proportional odds and accelerated failure time frameworks for right-censored data. RV Panaro (2020)
<[doi:10.48550/arXiv.2003.10548](https://doi.org/10.48550/arXiv.2003.10548)>.

License GPL-3

Biarch true

Depends R (>= 4.1.0), survival (>= 2.44-1.1), coda (>= 0.19-3)

Imports methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), generics, viridis

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Suggests testthat (>= 2.1.0), knitr, rmarkdown (>= 2.31), ggplot2 (>= 3.3.0), KMSurv, parsnip (>= 1.0.0), censored (>= 0.3.0), workflows (>= 1.1.0), tidybayes (>= 3.0.0), posterior (>= 1.4.0), covr, rsample (>= 1.1.0), rsurv, rstantools (>= 1.5.1)

VignetteBuilder knitr

Encoding UTF-8

SystemRequirements GNU make

BugReports <https://github.com/rvpanaro/spsurv/issues>

URL <https://github.com/rvpanaro/spsurv>,
<https://rvpanaro.github.io/spsurv/>

Config/roxygen2/version 8.0.0

Config/rstantools/auto_config FALSE

Config/pak/sysreqs cmake make

Repository <https://rvpanaro.r-universe.dev>

Date/Publication 2026-10-04 19:57:36 UTC

RemoteUrl <https://github.com/rvpanaro/spsurv>

RemoteRef HEAD

RemoteSha ee5461d3e38dc13279752f30ecfe3bc3aad74fff

Contents

spsurv-package	3
AIC.sbp	4
anova.sbp	4
as_draws_df.sbp	5
augment.sbp	5
bernstein	6
bp.basis	6
bp_survival_reg	7
bpaft	7
bp	8
bpo	9
coef.sbp	11
confint.sbp	11
credint	12
credint.sbp	12
estimates	13
extractAIC.sbp	13
ggresiduals	14
glance.sbp	15
logLik.sbp	15
model.matrix.sbp	16
plot.sbp	16
predict.sbp	17
print.sbp	19
print.summary.bpaft.bayes	20
print.summary.bpaft.mle	20
print.summary.bp.bayes	21
print.summary.bp.mle	21
print.summary.bpo.bayes	22
print.summary.bpo.mle	22
print.summary.sbp.bayes	23
print.summary.sbp.mle	23
proportional_odds	24
pw.basis	24
rank_models	25
residuals.sbp	25
se	26
sbp	27
sbp.default	28
spread_surv_draws.sbp	30

summary.sbp	30
survfit.sbp	31
tidy.sbp	33
vcov.sbp	34
Index	35

spsurv-package	<i>The 'spsurv' package.</i>
----------------	------------------------------

Description

A set of flexible routines to allow semiparametric survival regression modeling based on Bernstein polynomial, including Bernstein PH model (BPPH), Bernstein PO model (BPPO), and Bernstein AFT model (BPAFT) for right-censored data.

Details

`sbp` fits semi-parametric models for time-to-event survival data. Non-informative right-censoring assumption is available. Any user-defined Bernstein polynomial can be user-defined using an arbitrary degree, i.e. highest basis polynomials order.

The framework takes advantage of fully likelihood methods since the polynomial parameters are used to estimate the baseline functions. Even so, this is said to be semi-parametric since this approach does not rely on any distribution. Unlike the Cox model, the BP based models provide smooth hazard and survival curve estimates. For Bayesian fits, users should routinely inspect divergences, split- $\hat{R}$, and effective sample sizes, and tighten NUTS controls (e.g., increasing `adapt_delta`, `iter`, and `warmup`) when warnings are present.

Value

none

Author(s)

rvpanaro@gmail.com

References

Panaro R.V. (2020). `spsurv`: An R package for semi-parametric survival analysis. arXiv preprint arXiv:2003.10548.

Demarqui, F. N., & Mayrink, V. D. (2019). A fully likelihood-based approach to model survival data with crossing survival curves. arXiv preprint arXiv:1910.02406.

Demarqui, F. N., Mayrink, V. D., & Ghosh, S. K. (2019). An Unified Semiparametric Approach to Model Lifetime Data with Crossing Survival Curves. arXiv preprint arXiv:1910.04475.

Osman, M., & Ghosh, S. K. (2012). Nonparametric regression models for right-censored data using Bernstein polynomials. Computational Statistics & Data Analysis, 56(3), 559-573.

Lorentz, G. G. (1953). Bernstein polynomials. American Mathematical Society.

AIC.spbp	<i>Akaike information criterion for fitted spbp models</i>
----------	--

Description

Akaike information criterion for fitted spbp models

Usage

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

Arguments

object	A fitted "spbp" object.
...	Additional fitted models for comparison.
k	Penalty per parameter (default 2).

anova.spbp	<i>Analysis of deviance table for nested spbp models</i>
------------	--

Description

Compare nested MLE fits. With one model, builds a sequential table by refitting intercept-only and nested submodels along the formula term order (same idea as `anova(fit)` for [survreg](#)). With multiple models, compares each consecutive pair (same convention as `anova(fit0, fit1)` for `survreg`).

Usage

```
## S3 method for class 'spbp'
anova(object, ..., test = "Chisq")
```

Arguments

object	A fitted "spbp" object.
...	Additional nested fitted models (MLE, same family recommended).
test	Which test to report (only "Chisq" is supported).

Value

An "anova" object (also a `data.frame`). Pairwise comparisons use `Terms`, `Resid. Df`, `-2*LL`, `Test`, `Df`, `Deviance`, and `Pr(>Chi)` columns (as in `survival::survreg`). Sequential single-model tables use `Df`, `Deviance`, `Resid. Df`, `-2*LL`, and `Pr(>Chi)` with row names `NULL` and term labels.

as_draws_df.sbp	<i>Convert sbp Bayesian fit to posterior draws</i>
-----------------	--

Description

Convert sbp Bayesian fit to posterior draws

Usage

```
as_draws_df.sbp(x, variables = c("beta", "gamma"), ...)
```

Arguments

x	A fitted "sbp" object from approach = "bayes".
variables	Character vector of posterior components to include ("beta", "gamma", "log_lik").
...	Not used.

Value

A posterior::draws_df object on the back-transformed scale.

augment.sbp	<i>Augment data with sbp model information</i>
-------------	--

Description

Add fitted values, residuals, and optional survival predictions to the training (or supplied) data.

Usage

```
## S3 method for class 'sbp'
augment(
  x,
  data = NULL,
  eval_time = NULL,
  type = c("martingale", "deviance", "cox-snell", "coxsnell"),
  ...
)
```

Arguments

x	A fitted "sbp" object.
data	Optional data frame; defaults to the training data when available.
eval_time	Optional numeric vector of times for nested survival predictions (same structure as predict(x, type = "survival")).
type	Residual type passed to residuals.sbp .
...	Not used.

Value

A data.frame with original rows plus .residual and, when eval_time is set, a list-column .pred.

bernstein	<i>Bernstein polynomial baseline specification</i>
-----------	--

Description

Helper for specifying a Bernstein polynomial baseline via the dist / baseline arguments accepted by [spbp](#).

Usage

```
bernstein(m = NULL)
```

Arguments

m Bernstein polynomial degree (number of basis coefficients).

Value

A list with components baseline and m.

Examples

```
bernstein(5)
```

bp.basis	<i>Bernstein basis polynomials calculations</i>
----------	---

Description

Bernstein basis polynomials calculations

Usage

```
bp.basis(time, degree, tau = max(time))
```

Arguments

time a vector of times.
degree Bernstein polynomial degree
tau must be greater than times maximum value observed.

Value

A list containing matrices g and G corresponding BP basis and corresponding tau value used to compute them.

bp_survival_reg	<i>Bernstein survival regression specification</i>
-----------------	--

Description

Convenience constructor mapping family to the appropriate parsnip model (proportional_hazards, proportional_odds, or survival_reg).

Usage

```
bp_survival_reg(  
  family = c("ph", "po", "aft"),  
  mode = "censored regression",  
  engine = "spsurv"  
)
```

Arguments

family	Model family: "ph", "po", or "aft".
mode	Model mode (default "censored regression").
engine	parsnip engine (default "spsurv" for MLE).

Value

A parsnip model specification.

bpaft	<i>Bernstein AFT Model</i>
-------	----------------------------

Description

Fits the BPAFT model to time-to-event data.

Usage

```
bpaft(  
  formula,  
  degree = NULL,  
  data,  
  approach = c("mle", "bayes"),  
  dist = NULL,  
  baseline = NULL,  
  ...  
)
```

Arguments

formula	a Surv object with time to event observations, right censoring status and explanatory terms.
degree	Bernstein polynomial degree. If omitted, defaults to <code>ceiling(sqrt(n))</code> for <code>n = nrow(data)</code> (see sbbp).
data	a data.frame object.
approach	Bayesian or maximum likelihood estimation methods, default is <code>approach = "mle"</code> .
dist	optional baseline specification; use bernstein(m) for the Bernstein polynomial degree.
baseline	optional alias for <code>dist</code> .
...	further arguments passed to or from other methods

Value

An object of class `sbbp`, including component `degree`.

See Also

[sbbp](#), [bpph](#) and [bppo](#) for other BP based models.

Examples

```
library("spsurv")
data("veteran", package = "survival")

fit <- bpaft(Surv(time, status) ~ karno + celltype,
  data = veteran
)

summary(fit)
```

bpph

Bernstein PH Model

Description

Fits the BPPH model to time-to-event data.

Usage

```
bpph(
  formula,
  degree = NULL,
  data,
  approach = c("mle", "bayes"),
```

```

    dist = NULL,
    baseline = NULL,
    ...
  )

```

Arguments

formula	a Surv object with time to event observations, right censoring status and explanatory terms.
degree	Bernstein polynomial degree. If omitted, defaults to <code>ceiling(sqrt(n))</code> for <code>n = nrow(data)</code> (see spbp).
data	a data.frame object.
approach	Bayesian or maximum likelihood estimation methods, default is <code>approach = "mle"</code> .
dist	optional baseline specification; use bernstein(m) for the Bernstein polynomial degree.
baseline	optional alias for <code>dist</code> .
...	further arguments passed to or from other methods

Value

An object of class `spbp`, including component `degree`.

See Also

[spbp](#), [bppo](#) and [bpaft](#) for other BP based models.

Examples

```

library("spsurv")
data("veteran", package = "survival")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
  data = veteran
)

summary(fit)

```

bppo

Bernstein PO Model

Description

Fits the BPPO model to time-to-event data.

Usage

```
bppo(
  formula,
  degree = NULL,
  data,
  approach = c("mle", "bayes"),
  dist = NULL,
  baseline = NULL,
  ...
)
```

Arguments

formula	a Surv object with time-to-event observations, right censoring status and explanatory terms.
degree	Bernstein polynomial degree. If omitted, defaults to <code>ceiling(sqrt(n))</code> for <code>n = nrow(data)</code> (see spbp).
data	a data.frame object.
approach	Bayesian or maximum likelihood estimation methods, default is <code>approach = "mle"</code> .
dist	optional baseline specification; use bernstein(m) for the Bernstein polynomial degree.
baseline	optional alias for <code>dist</code> .
...	further arguments passed to or from other methods

Value

An object of class `spbp`, including component `degree`.

See Also

[spbp](#), [bpqh](#) and [bpafit](#) for other BP based models.

Examples

```
library("spsurv")
data("veteran", package = "survival")

fit <- bppo(Surv(time, status) ~ karno + celltype,
  data = veteran
)

summary(fit)
```

coef.spbp	<i>Estimated regression coefficients</i>
-----------	--

Description

Estimated regression coefficients

Usage

```
## S3 method for class 'spbp'
coef(object, summary = c("mean", "median", "mode"), ...)
```

Arguments

object	an object of the class spbp
summary	posterior summary if method = "bayes" in x
...	further arguments passed to or from other methods

Value

the estimated regression coefficients

confint.spbp	<i>Confidence intervals for the regression coefficients</i>
--------------	---

Description

Confidence intervals for the regression coefficients

Usage

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

Arguments

object	a fitted model object.
parm	a specification of which parameters are to be given confidence intervals: regression coefficient names and/or Bernstein baseline names (e.g. "gamma[1]"). If missing, all regression coefficients are used.
level	the confidence level required.
...	further arguments passed to parent method

Value

100(1-alpha) confidence intervals for the requested parameters.

credint	<i>Generic S3 method credint</i>
---------	----------------------------------

Description

Generic S3 method credint

Usage

```
credint(x, ...)
```

Arguments

x	a fitted model object
...	further arguments passed to parent method

Value

Generic function dispatching to credint.spbp.

credint.spbp	<i>Credible intervals for the regression coefficients</i>
--------------	---

Description

Credible intervals for the regression coefficients

Usage

```
## S3 method for class 'spbp'
credint(x, prob = 0.95, type = c("HPD", "Equal-Tailed"), ...)
```

Arguments

x	an object of the class x.
prob	the probability level required.
type	interval type.
...	further arguments passed to or from other methods.

Value

100(1-alpha) credible intervals for the regression coefficients

estimates	<i>Parameter estimates for fitted spbp models</i>
-----------	---

Description

Parameter estimates for fitted spbp models

Usage

```
estimates(object, ...)
```

Arguments

object	A fitted "spbp" object.
...	Not used.

Value

Named numeric vector of regression and Bernstein-polynomial parameters.

extractAIC.spbp	<i>Extract AIC from a fitted spbp model</i>
-----------------	---

Description

Extract AIC from a fitted spbp model

Usage

```
## S3 method for class 'spbp'  
extractAIC(fit, scale = 0, k = 2, ...)
```

Arguments

fit	A fitted "spbp" object.
scale	Not used.
k	Penalty per parameter (default 2).
...	Not used.

ggresiduals

ggplot2 residual diagnostic plots for spbp models

Description

Convenience wrapper around [residuals.spbp](#) that returns a **ggplot2** object for martingale, deviance, or Cox-Snell residuals.

Usage

```
ggresiduals(
  object,
  type = c("martingale", "deviance", "cox-snell", "coxsnell"),
  against = c("fitted", "index"),
  ...
)
```

Arguments

<code>object</code>	A fitted "spbp" object.
<code>type</code>	Residual type: "martingale" (default), "deviance", "cox-snell", or "coxsnell".
<code>against</code>	What to plot on the x-axis: "fitted" (default, Cox-Snell cumulative hazard) or "index" (observation index).
<code>...</code>	Further arguments passed to residuals.spbp .

Value

A ggplot object (requires **ggplot2**).

Examples

```
## Not run:
library(spsurv)
library(ggplot2)
data(veteran, package = "survival")
fit <- bpph(Surv(time, status) ~ karno, data = veteran, degree = 4)
ggresiduals(fit, type = "martingale")

## End(Not run)
```

glance.spbp	<i>Glance at a fitted spbp model</i>
-------------	--------------------------------------

Description

Return a one-row broom/generics-style summary of model-level statistics.

Usage

```
## S3 method for class 'spbp'  
glance(x, ...)
```

Arguments

x	A fitted "spbp" object.
...	Currently unused.

Value

A one-row data.frame with model-level statistics. Columns that are entirely NA for the fitted object are omitted.

logLik.spbp	<i>Log-likelihood for fitted spbp models</i>
-------------	--

Description

Log-likelihood for fitted spbp models

Usage

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

Arguments

object	A fitted "spbp" object.
...	Not used.

model.matrix.spbp	<i>Model.matrix method for fitted spbp models</i>
-------------------	---

Description

Model.matrix of a fitted [spbp](#) model.

Usage

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

Arguments

object	an object of class ‘spbp’, see spbp .
...	arguments passed to parent method.

Value

The model matrix.

See Also

[spbp](#), [model.matrix](#)

Examples

```
library("spsurv")
data("veteran", package = "survival")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
  data = veteran
)

model.matrix(fit)
```

plot.spbp	<i>BP based models plot.</i>
-----------	------------------------------

Description

Plot for a fitted [spbp](#) model.

Usage

```
## S3 method for class 'spbp'
plot(
  x,
  main,
  graph = c("baseline", "basis"),
  cumulative = FALSE,
  frame = FALSE,
  lwd = 3,
  ...
)
```

Arguments

x	an object of class ‘spbp’ result of a spbp fit.
main	graph title
graph	type of polynomial graph, default is "basis"
cumulative	TRUE for odds and cumulative hazard
frame	graphical parameter; default is FALSE
lwd	graphical parameter; default is 3
...	further arguments passed to or from other methods

See Also

[spbp.](#)

Examples

```
library("spsurv")
data("veteran", package = "survival")

fit <- bph(Surv(time, status) ~ karno + factor(celltype),
  data = veteran
)
plot(fit)
```

predict.spbp

Predicted survival as a tidy data frame

Description

Survival (and optional CI) on a time grid, as a `data.frame` for `ggplot2::geom_line`.

Usage

```
## S3 method for class 'spbp'
predict(
  object,
  newdata = NULL,
  times = NULL,
  eval_time = NULL,
  type = NULL,
  conf_type = NULL,
  interval = 0.95,
  interval.type = c("hpd", "quantile"),
  monotone = NULL,
  ...
)
```

Arguments

object	Fitted "spbp" from spbp / bpph / etc.
newdata	Optional data.frame of covariate profiles (same convention as survfit.spbp).
times	Time grid. Default: a dense sequence from 0 to the maximum observed time (suitable for smooth <code>ggplot2::geom_line</code> / <code>geom_ribbon</code> plots). Pass a long <code>seq(...)</code> to override resolution; use observed event times only if stepwise curves are intended.
eval_time	Evaluation times for type = "survival" (required).
type	Prediction type. For <code>tidymodels</code> /censored compatibility use "survival", "time", or "linear_pred". For survival curves (default), use NULL, "curve", or a confidence transformation ("log", "log-log", "plain").
conf_type	Confidence interval transformation for curve predictions ("log", "log-log", "plain"). Used when type is NULL, "curve", or a confidence transformation name.
interval	Confidence level for curve predictions; passed to survfit.spbp .
interval.type, monotone	Passed to survfit.spbp (Bayesian fits).
...	Passed to survfit.spbp for curve predictions.

Value

For curve predictions, same structure as [as.data.frame.survfitbp](#). For type = "survival", a tibble with list-column `.pred` (elements contain `.eval_time` and `.pred_survival`). For type = "time", a tibble with `.pred_time`. For type = "linear_pred", a tibble with `.pred_linear_pred`.

See Also

[survfit.spbp](#), [augment.spbp](#)

Examples

```
data(veteran, package = "survival")
fit <- bpph(Surv(time, status) ~ karno, data = veteran, approach = "mle", init = 0)
pr <- predict(fit, times = seq(0, 400, by = 2))
predict(fit, veteran[1:2, ], type = "survival", eval_time = c(100, 200))
predict(fit, veteran[1:2, ], type = "time")
## Not run:
  ggplot2::ggplot(pr, ggplot2::aes(time, surv)) + ggplot2::geom_line()

## End(Not run)
```

print.spbp

*Bernstein Polynomial Based Regression Object Print***Description**

Bernstein Polynomial Based Regression Object Print

Usage

```
## S3 method for class 'spbp'
print(
  x,
  bp.param = FALSE,
  show_baseline = FALSE,
  digits = 2,
  signif.stars = getOption("show.signif.stars"),
  what = "summary",
  ...
)
```

Arguments

<code>x</code>	an object of class <code>spbp</code> .
<code>bp.param</code>	print BP parameters only (alias for baseline-only display).
<code>show_baseline</code>	logical; append Bernstein baseline Wald table after regression coefficients (MLE only). See also <code>print(fit, bp.param = TRUE)</code> for baseline-only output.
<code>digits</code>	number of digits to display; defaults to 2.
<code>signif.stars</code>	see getOption .
<code>what</code>	character vector; any of "summary", "tidy", and "glance" for broom-style coefficient and model-level tables.
<code>...</code>	further arguments passed to summary.spbp .

Value

none

```
print.summary.bpaft.bayes
```

*Bernstein Polynomial Based Regression Object Summary BPAFT
Bayes*

Description

Bernstein Polynomial Based Regression Object Summary BPAFT Bayes

Usage

```
## S3 method for class 'summary.bpaft.bayes'  
print(...)
```

Arguments

... further arguments passed to or from other methods

Value

none

```
print.summary.bpaft.mle
```

Bernstein Polynomial Based Regression Object Summary BPAFT MLE

Description

Bernstein Polynomial Based Regression Object Summary BPAFT MLE

Usage

```
## S3 method for class 'summary.bpaft.mle'  
print(...)
```

Arguments

... further arguments passed to or from other methods

Value

none

`print.summary.bpph.bayes`

Bernstein Polynomial Based Regression Object Summary BPPH Bayes

Description

Bernstein Polynomial Based Regression Object Summary BPPH Bayes

Usage

```
## S3 method for class 'summary.bpph.bayes'  
print(...)
```

Arguments

... further arguments passed to or from other methods

Value

none

`print.summary.bpph.mle`

Bernstein Polynomial Based Regression Object Summary BPPH MLE

Description

Bernstein Polynomial Based Regression Object Summary BPPH MLE

Usage

```
## S3 method for class 'summary.bpph.mle'  
print(...)
```

Arguments

... further arguments passed to or from other methods

Value

none

```
print.summary.bppo.bayes
```

Bernstein Polynomial Based Regression Object Summary BPPO Bayes

Description

Bernstein Polynomial Based Regression Object Summary BPPO Bayes

Usage

```
## S3 method for class 'summary.bppo.bayes'  
print(...)
```

Arguments

... further arguments passed to or from other methods

Value

none

```
print.summary.bppo.mle
```

Bernstein Polynomial Based Regression Object BPPO MLE

Description

Bernstein Polynomial Based Regression Object BPPO MLE

Usage

```
## S3 method for class 'summary.bppo.mle'  
print(...)
```

Arguments

... further arguments passed to or from other methods

Value

none

```
print.summary.spbp.bayes
```

Bernstein Polynomial Based Regression Object Summary Bayes

Description

Bernstein Polynomial Based Regression Object Summary Bayes

Usage

```
## S3 method for class 'summary.spbp.bayes'  
print(x, digits = 2, signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

x	a summary.spbp.bayes object
digits	number of digits to display.
signif.stars	see getOption
...	further arguments passed to or from other methods

Value

none

```
print.summary.spbp.mle
```

Bernstein Polynomial Based Regression Object Summary MLE

Description

Bernstein Polynomial Based Regression Object Summary MLE

Usage

```
## S3 method for class 'summary.spbp.mle'  
print(x, digits = 2, signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

x	a summary.spbp.mle object
digits	number of digits to display.
signif.stars	see getOption
...	further arguments passed to or from other methods

Value

none

proportional_odds	<i>Proportional odds regression model specification</i>
-------------------	---

Description

Proportional odds survival regression with a Bernstein polynomial baseline ([bppo](#)). This model type is registered by **spsurv** for use with tidymodels; it is not part of the **censored** package.

Usage

```
proportional_odds(mode = "censored regression", engine = "spsurv")
```

Arguments

mode	A single character string for the prediction outcome mode. The only possible value for this model is "censored regression".
engine	A single character string specifying what computational engine to use for fitting.

Value

A model specification.

pw.basis	<i>Power basis polynomials calculations</i>
----------	---

Description

Power basis polynomials calculations

Usage

```
pw.basis(degree)
```

Arguments

degree	Bernstein polynomial degree
--------	-----------------------------

Value

A list containing matrices g and G corresponding BP basis and corresponding tau value used to compute them.

rank_models	<i>Rank fitted spbp models by AIC</i>
-------------	---------------------------------------

Description

Convenience wrapper around [AIC](#) that returns models sorted from lowest to highest AIC.

Usage

```
rank_models(...)
```

Arguments

... One or more fitted "spbp" objects (MLE).

Value

A data.frame with columns fit, model, degree, aic, and npars.

Examples

```
## Not run:
library(spsurv)
data(veteran, package = "survival")
f1 <- bpph(Surv(time, status) ~ 1, data = veteran, degree = 4)
f2 <- bpph(Surv(time, status) ~ karno, data = veteran, degree = 4)
rank_models(f1, f2)

## End(Not run)
```

residuals.spbp	<i>BP based models residuals.</i>
----------------	-----------------------------------

Description

Residuals for a fitted [spbp](#) model.

Usage

```
## S3 method for class 'spbp'
residuals(
  object,
  type = c("martingale", "deviance", "cox-snell", "coxsnell"),
  ...
)
```

Arguments

- object an object of class 'spbp' result of a [spbp](#) fit.
- type type of residuals; default is "martingale". Also "deviance", "cox-snell", and "coxsnell" (alias).
- ... arguments passed to parent method.

See Also

[spbp](#), [spbp](#).

Examples

```
library("spsurv")
data("veteran", package = "survival")

fit <- bpph(Surv(time, status) ~ karno + factor(celltype),
  data = veteran
)

residuals(fit)
```

se	<i>Standard errors for fitted spbp model parameters</i>
----	---

Description

Standard errors for fitted spbp model parameters

Usage

```
se(object, ...)
```

Arguments

- object A fitted "spbp" object.
- ... Not used.

Value

Named numeric vector of standard errors aligned with [estimates](#).

spbp

spbp: The BP Based Survival Analysis Function

Description

Semiparametric Survival Analysis Using Bernstein Polynomial

Usage

```
spbp(formula, ...)
```

Arguments

`formula` a [Surv](#) response with event time, censoring status, and optional covariates.
`...` Arguments passed to [spbp.default](#), including data, model, approach, and degree. Further arguments in `...` are passed to `rstan::optimizing` (MLE) or `rstan::sampling` (Bayes), e.g. `iter`, `chains`, `init`.

Details

Fits Bernstein PH, PO, or AFT models to survival data via Stan (MLE or Bayesian).

The generic dispatches to [spbp.default](#) for formula objects. Convenience wrappers [bph](#), [bpo](#), and [bapt](#) fix the model family. See [vignette\("getting-started", package = "spsurv"\)](#) for a tutorial, [vignette\("model-families", package = "spsurv"\)](#) for PH / PO / AFT comparison, and [vignette\("bp-degree", package = "spsurv"\)](#) for choosing the Bernstein polynomial degree.

Value

An object of class "spbp". See [spbp.default](#) for the list of components (coefficients, `bp.param`, degree, etc.).

See Also

[spbp.default](#), [bph](#), [bpo](#), [bapt](#), [bernstein](#)

Examples

```
library("spsurv")
data("veteran", package = "survival")

fit_mle <- spbp(Surv(time, status) ~ karno + factor(celltype),
  data = veteran, model = "po"
)
summary(fit_mle)

fit_bayes <- spbp(Surv(time, status) ~ karno + factor(celltype),
  data = veteran, model = "po", approach = "bayes",
```

```

cores = 1, iter = 300, chains = 1,
priors = list(
  beta = c("normal(0,5)"),
  gamma = "halfnormal(0,5)"
)
)

summary(fit_bayes)

```

spbp.default

spbp: The BP Based Semiparametric Survival Analysis Function

Description

spbp: The BP Based Semiparametric Survival Analysis Function

Usage

```

## Default S3 method:
spbp(
  formula,
  degree,
  data,
  approach = c("mle", "bayes"),
  model = c("ph", "po", "aft"),
  priors = list(beta = c("normal(0,2)"), gamma = c("lognormal(0,4)"), frailty =
    c("gamma(0.01,0.01)")),
  cores = .spbp_default_cores(),
  scale = TRUE,
  dist = NULL,
  baseline = NULL,
  verbose = FALSE,
  chains = 4,
  ...
)

```

Arguments

formula	a Surv object with time to event, status and explanatory terms
degree	Bernstein polynomial degree (integer). If omitted and neither dist nor baseline supplies bernstein(m) , the default is <code>ceiling(sqrt(n))</code> where n is the number of rows in data.
data	a data.frame object
approach	Bayesian or Maximum Likelihood estimation methods; default is "mle"
model	Bernstein PH ("ph"), PO ("po"), or AFT ("aft") model; default is "ph"

priors	prior settings for the Bayesian approach; ‘normal’ or ‘cauchy’ for beta; ‘lognormal’ or ‘loglogistic’ for gamma (BP coefficients). Defaults are <code>normal(0,2)</code> for standardised regression coefficients and <code>lognormal(0,4)</code> for Bernstein coefficients.
cores	number of core threads to use (Bayes sampling)
scale	logical; indicates whether to center and scale the data
dist	optional baseline specification; use <code>bernstein(m)</code> for the Bernstein polynomial degree
baseline	optional alias for <code>dist</code>
verbose	passed to Stan
chains	number of MCMC chains (Bayes)
...	further arguments passed to <code>rstan::optimizing</code> (MLE) or <code>rstan::sampling</code> (Bayes), e.g. <code>iter</code> , <code>warmup</code> , <code>init</code> .

Details

Right-censored survival data are modeled with a Bernstein-polynomial baseline and regression on covariates. With `approach = "mle"`, parameters are estimated by Stan’s optimizer and approximate inference uses the Hessian when available. With `approach = "bayes"`, posterior samples are drawn with NUTS; use `summary`, [tidy.spbp](#), and [glance.spbp](#) for output. Covariates are centered and scaled when `scale = TRUE` (default).

The returned object includes:

`coefficients` Regression estimates on the original covariate scale.

`bp.param` Bernstein baseline coefficients (gamma).

`degree` Polynomial degree used (also in `call$degree`).

`loglik` MLE: intercept-only and full-model log-likelihoods; Bayes: posterior mean pointwise log-likelihoods.

`call` Matched call with `approach`, `model`, and `degree`.

Value

An object of class `spbp`. Component `degree` records the Bernstein polynomial degree used in the fit (also stored in `call$degree`).

See Also

[bpph](#), [bppp](#), [bpafit](#), [bernstein](#), [summary.spbp](#)

```
spread_surv_draws.spbp
```

Posterior survival curves in long tidy format

Description

Posterior survival curves in long tidy format

Usage

```
spread_surv_draws.spbp(object, times, newdata = NULL, ...)
```

Arguments

<code>object</code>	A Bayesian "spbp" fit.
<code>times</code>	Numeric vector of evaluation times.
<code>newdata</code>	Optional covariate profiles.
<code>...</code>	Not used.

Value

A data.frame with `.chain`, `.iteration`, `.draw`, `time`, `surv`, and optional covariate columns.

```
summary.spbp
```

Bernstein Polynomial Based Regression Object Summary

Description

Bernstein Polynomial Based Regression Object Summary

Usage

```
## S3 method for class 'spbp'
summary(
  object,
  interval = 0.95,
  compact = TRUE,
  show_call = TRUE,
  show_intervals = TRUE,
  show_baseline = FALSE,
  mle_test = c("lr", "wald"),
  bayes_criterion = c("waic", "dic", "lpml"),
  ...
)
```

Arguments

object	an object of class <code>spbp</code>
interval	interval coverage (confidence or credibility)
compact	logical; if TRUE, <code>print.summary</code> methods show essential output only.
show_call	logical; include the model call in printed summary.
show_intervals	logical; include interval table in printed summary.
show_baseline	logical; include Bernstein baseline (gamma) Wald table in printed summary (MLE only).
mle_test	global test to print for MLE summaries: "lr" or "wald".
bayes_criterion	global criterion to print for Bayesian summaries: "waic", "dic", or "lpml".
...	further arguments passed to or from other methods

Value

An object of class analogous to for e.g. `'summary.bppo.bayes'`.

<code>survfit.spbp</code>	<i>BP-based model survival curves</i>
---------------------------	---------------------------------------

Description

Compute survival curves for a fitted `spbp` model.

Usage

```
## S3 method for class 'spbp'
survfit(
  formula,
  newdata = NULL,
  times = NULL,
  se.fit = TRUE,
  interval = 0.95,
  type = c("log", "log-log", "plain"),
  interval.type = c("hpd", "quantile"),
  monotone = NULL,
  tidy = FALSE,
  baseline = FALSE,
  ...
)

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

```
## S3 method for class 'survfitbp'
plot(
  x,
  conf.int,
  col = 1,
  lty = 1,
  lwd = 1.5,
  xlab = "Time",
  ylab = "Survival probability",
  ylim = c(0, 1),
  mark.time = FALSE,
  ...
)
```

Arguments

formula	An object of class "spbp" returned by spbp .
newdata	Optional data frame used to obtain survival curves for specific covariate values.
times	Optional evaluation times: a Surv object (legacy), or a non-negative numeric vector (unique values are used; suitable for smooth <code>ggplot2::geom_line</code> plots via predict.spbp or as.data.frame.survfitbp).
se.fit	Logical; if TRUE, compute standard errors.
interval	Confidence level for intervals (e.g. 0.95).
type	Character; confidence interval transformation. One of "log", "log-log", or "plain".
interval.type	For Bayesian fits only: "hpd" (default) or "quantile" (equal-tailed).
monotone	Logical; for Bayesian fits, enforce non-increasing credible-band limits over time so ribbons plot smoothly with <code>ggplot2::geom_line</code> . Defaults to TRUE when <code>approach = "bayes"</code> .
tidy	Logical; if TRUE, return a <code>data.frame</code> for <code>ggplot2</code> . instead of a "survfit" object.
baseline	Logical; if TRUE, return the baseline survival curve $S_0(t)$ at observed event times (or times) with no covariate effect. When <code>tidy = TRUE</code> , returns a simple <code>data.frame</code> with columns <code>time</code> and <code>surv</code> .
...	Further arguments passed to survfit.coxph for the reference Cox object (e.g. <code>conf.type</code> is ignored; use <code>type</code> instead).
x	Object from survfit.spbp .
row.names, optional	Unused; included for S3 consistency.
conf.int	Logical; draw pointwise interval limits. Defaults to TRUE for a single curve when limits are present.
col, lty, lwd, xlab, ylab, ylim	Graphical parameters.
mark.time	Ignored (Bernstein curves are continuous).

Value

An object of class "survfit" (with classes survfitbp, survfitcox, survfit).

A data.frame with columns id (curve index), time, surv, lower, upper, cumhaz, std.err.

Functions

- `as.data.frame(survfitbp)`: Tidy survival curves for `ggplot2::geom_line/geom_ribbon`.
- `plot(survfitbp)`: Plot Bernstein-polynomial survival as a smooth line (`type = "l"`), never a Kaplan–Meier step function. Pass `conf.int = TRUE` for pointwise bands (also drawn with lines).

See Also

[spbp](#), [survfit](#), [predict.spbp](#), [survfit.spbp](#) (`as.data.frame` method).

Examples

```
library(spsurv)
data(veteran, package = "survival")
fit <- bpph(Surv(time, status) ~ karno + factor(celltype), data = veteran)
survfit(fit)
plot(survfit(fit, times = seq(0, max(veteran$time), length.out = 80)))

data(veteran, package = "survival")
fit <- bpph(Surv(time, status) ~ karno, data = veteran, approach = "mle", init = 0)
sf <- survfit(fit, times = seq(0, max(veteran$time), length.out = 80))
head(as.data.frame(sf))
```

tidy.spbp

Tidy summary for fitted spbp models

Description

Create a broom/generics-style tidy data frame with one row per model term.

Usage

```
## S3 method for class 'spbp'
tidy(
  x,
  conf.int = FALSE,
  conf.level = 0.95,
  exponentiate = FALSE,
  component = c("coef", "baseline", "all"),
  ...
)
```

Arguments

<code>x</code>	A fitted "spbp" object.
<code>conf.int</code>	Logical; include confidence/credible interval columns.
<code>conf.level</code>	Interval coverage level.
<code>exponentiate</code>	Logical; if TRUE, exponentiate the estimate and interval columns (regression coefficients only).
<code>component</code>	Which parameters to return: "coef" (default), "baseline" (Bernstein gamma weights), or "all".
<code>...</code>	Currently unused.

Value

A data.frame with columns `term`, `estimate`, `std.error`, and when available `statistic`, `p.value`, `conf.low`, `conf.high`. Columns that are entirely NA are omitted.

<code>vcov.spbp</code>	<i>Covariance of the regression coefficients</i>
------------------------	--

Description

Uses block-wise inversion of the negative Hessian, with a clear split between the regression coefficients (beta) and the Bernstein polynomial coefficients (gamma).

Usage

```
## S3 method for class 'spbp'
vcov(object, bp.param = FALSE, mask_unstable_gamma = TRUE, ...)
```

Arguments

<code>object</code>	an object of the class <code>spbp</code>
<code>bp.param</code>	return Bernstein Polynomial variance.
<code>mask_unstable_gamma</code>	when TRUE (default), set γ -related entries to NA if the Bernstein information block is ill-conditioned. Set FALSE internally when propagating survival uncertainty despite instability.
<code>...</code>	arguments passed to parent method.

Value

the variance-covariance matrix associated with the regression coefficients.

Index

AIC, [25](#)
AIC.spbp, [4](#)
anova.spbp, [4](#)
as.data.frame.survfitbp, [18](#), [32](#)
as.data.frame.survfitbp(survfit.spbp),
 [31](#)
as_draws_df.spbp, [5](#)
augment.spbp, [5](#), [18](#)

bernstein, [6](#), [8–10](#), [27–29](#)
bp.basis, [6](#)
bp_survival_reg, [7](#)
bpaft, [7](#), [9](#), [10](#), [27](#), [29](#)
bpph, [8](#), [8](#), [10](#), [18](#), [27](#), [29](#)
bpp, [8](#), [9](#), [9](#), [24](#), [27](#), [29](#)

coef.spbp, [11](#)
confint.spbp, [11](#)
credint, [12](#)
credint.spbp, [12](#)

estimates, [13](#), [26](#)
extractAIC.spbp, [13](#)

getOption, [19](#), [23](#)
ggresiduals, [14](#)
glance.spbp, [15](#), [29](#)

logLik.spbp, [15](#)

model.matrix, [16](#)
model.matrix.spbp, [16](#)

plot.spbp, [16](#)
plot.survfitbp(survfit.spbp), [31](#)
predict.spbp, [17](#), [32](#), [33](#)
print.spbp, [19](#)
print.summary.bpaft.bayes, [20](#)
print.summary.bpaft.mle, [20](#)
print.summary.bpph.bayes, [21](#)
print.summary.bpph.mle, [21](#)

print.summary.bppo.bayes, [22](#)
print.summary.bppo.mle, [22](#)
print.summary.spbp.bayes, [23](#)
print.summary.spbp.mle, [23](#)
proportional_odds, [24](#)
pw.basis, [24](#)

rank_models, [25](#)
residuals.spbp, [5](#), [14](#), [25](#)

se, [26](#)
spbp, [3](#), [6](#), [8–10](#), [16–18](#), [25](#), [26](#), [27](#), [31–33](#)
spbp.default, [27](#), [28](#)
spread_surv_draws.spbp, [30](#)
spsurv (spsurv-package), [3](#)
spsurv-package, [3](#)
summary.spbp, [19](#), [29](#), [30](#)
Surv, [27](#), [32](#)
survfit, [33](#)
survfit.coxph, [32](#)
survfit.spbp, [18](#), [31](#), [32](#), [33](#)
survreg, [4](#)

tidy.spbp, [29](#), [33](#)

vcov.spbp, [34](#)