

# Package ‘survstan’

April 12, 2024

**Title** Fitting Survival Regression Models via 'Stan'

**Version** 0.0.7.1

## Description

Parametric survival regression models under the maximum likelihood approach via 'Stan'. Implemented regression models include accelerated failure time models, proportional hazards models, proportional odds models, accelerated hazard models, Yang and Prentice models, and extended hazard models. Available baseline survival distributions include exponential, Weibull, log-normal, log-logistic, gamma, generalized gamma, rayleigh, Gompertz and fatigue (Birnbaum-Saunders) distributions. References: Lawless (2002) <ISBN:9780471372158>; Bennett (1982) <[doi:10.1002/sim.4780020223](https://doi.org/10.1002/sim.4780020223)>; Chen and Wang(2000) <[doi:10.1080/01621459.2000.10474236](https://doi.org/10.1080/01621459.2000.10474236)>; Demarqui and Mayrink (2021) <[doi:10.1214/20-BJPS471](https://doi.org/10.1214/20-BJPS471)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Biarch** true

**Depends** R (>= 3.4.0), survival

**Imports** actuar (>= 3.0.0), broom, doFuture, dplyr, extraDistr, foreach, future, generics, ggplot2, gridExtra, MASS, methods, purrr, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), Rdpack, rlang, rstan (>= 2.26.0), rstantools (>= 2.3.1), tibble, tidyr

**RdMacros** Rdpack

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

**SystemRequirements** GNU make

**URL** <https://github.com/fndemarqui/survstan>,  
<https://fndemarqui.github.io/survstan/>

**BugReports** <https://github.com/fndemarqui/survstan/issues>

**Suggests** emmeans (>= 1.4.2), estimability, GGally, knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Fabio Demarqui [aut, cre, cph]  
 (<<https://orcid.org/0000-0001-9236-1986>>),  
 Andrew Johnson [ctb]

**Maintainer** Fabio Demarqui <fndemarqui@est.ufmg.br>

**Repository** CRAN

**Date/Publication** 2024-04-12 16:50:02 UTC

## R topics documented:

survstan-package	3
aftreg	3
ahreg	4
AIC.survstan	5
anova.survstan	6
coef.survstan	7
confint.survstan	7
cross_time	8
cross_time.survstan	9
ehreg	10
emmeans-survstan-helpers	11
estimates	11
extractAIC.survstan	12
gastric	13
ggprentice	13
ggresiduals	15
ggresiduals.survstan	15
ggstacy	16
Gompertz	17
ipass	18
logLik.survstan	19
model.matrix.survstan	20
phreg	20
poreg	21
print.summary.survstan	22
rank_models	23
residuals.survstan	24
se	25
se.survstan	25
summary.survstan	26
survfit.survstan	26
tidy.survstan	27
vcov.survstan	28
ypreg	29

---

survstan-package      *The 'survstan' package.*

---

## Description

The aim of the R package `survstan` is to provide a toolkit for fitting survival models using Stan. The R package `survstan` can be used to fit right-censored survival data under independent censoring. The implemented models allow the fitting of survival data in the presence/absence of covariates. All inferential procedures are currently based on the maximum likelihood (ML) approach.

`_PACKAGE`

## References

Stan Development Team (2023). “RStan: the R interface to Stan.” R package version 2.21.8, <https://mc-stan.org/>.

Lawless JF (2002). *Statistical Models and Methods for Lifetime Data*, Wiley Series in Probability and Statistics, 2nd Edition edition. John Wiley and Sons. ISBN 9780471372158.

Bennett S (1983). “Analysis of survival data by the proportional odds model.” *Statistics in Medicine*, **2**(2), 273-277. doi:10.1002/sim.4780020223.

Chen YQ, Wang M (2000). “Analysis of Accelerated Hazards Models.” *Journal of the American Statistical Association*, **95**(450), 608-618. doi:10.1080/01621459.2000.10474236.

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.

---

`aftreg`      *Fitting Accelerated Failure Time Models*

---

## Description

Function to fit accelerated failure time (AFT) models.

## Usage

```
aftreg(formula, data, baseline = "weibull", dist = NULL, init = 0, ...)
```

**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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.
baseline	the chosen baseline distribution; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distribution (for compatibility with the <a href="#">survreg</a> function); default is NULL.
init	initial values specification (default value is 0); see the detailed documentation for <code>init</code> in <a href="#">optimizing</a> .
...	further arguments passed to other methods.

**Value**

ahreg returns an object of class "aftreg" containing the fitted model.

**Examples**

```
library(survstan)
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
summary(fit)
```

---

ahreg

*Fitting Accelerated Hazard Models*


---

**Description**

Function to fit accelerated hazard (AH) models.

**Usage**

```
ahreg(formula, data, baseline = "weibull", dist = NULL, init = 0, ...)
```

**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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.

baseline	the chosen baseline distribution; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distribution (for compatibility with the <a href="#">survreg</a> function); default is NULL.
init	initial values specification (default value is 0); see the detailed documentation for <code>init</code> in <a href="#">optimizing</a> .
...	further arguments passed to other methods.

**Value**

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

**Examples**

```
library(survstan)
fit <- ahreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
summary(fit)
```

---

AIC.survstan	<i>Akaike information criterion</i>
--------------	-------------------------------------

---

**Description**

Akaike information criterion

**Usage**

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

**Arguments**

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

**Value**

the Akaike information criterion value when a single model is passed to the function; otherwise, a `data.frame` with the Akaike information criterion values and the number of parameters is returned.

## Examples

```
library(survstan)
fit1 <- aftreg(Surv(futime, fustat) ~ 1, data = ovarian, baseline = "weibull", init = 0)
fit2 <- aftreg(Surv(futime, fustat) ~ rx, data = ovarian, baseline = "weibull", init = 0)
fit3 <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
AIC(fit1, fit2, fit3)
```

---

anova.survstan

*anova method for survstan models*

---

## Description

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

## Usage

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

## Arguments

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

## Value

the ANOVA table.

## Examples

```
library(survstan)
fit1 <- aftreg(Surv(futime, fustat) ~ 1, data = ovarian, baseline = "weibull", init = 0)
fit2 <- aftreg(Surv(futime, fustat) ~ rx, data = ovarian, baseline = "weibull", init = 0)
fit3 <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
anova(fit1, fit2, fit3)
```

---

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

---

**Description**

Estimated regression coefficients

**Usage**

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

**Arguments**

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

**Value**

the estimated regression coefficients

**Examples**

```
library(survstan)  
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)  
coef(fit)
```

---

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

---

**Description**

Confidence intervals for the regression coefficients

**Usage**

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

**Arguments**

object	an object of the class survstan.
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.

**Examples**

```
library(survstan)
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
confint(fit)
```

---

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

---

**Description**

Generic S3 method cross\_time

**Usage**

```
cross_time(object, ...)
```

**Arguments**

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

**Value**

the crossing survival time

---

cross\_time.survstan    *Computes the crossing survival times*

---

### Description

Computes the crossing survival times

### Usage

```
## S3 method for class 'survstan'  
cross_time(  
  object,  
  newdata1,  
  newdata2,  
  conf.level = 0.95,  
  nboot = 1000,  
  cores = 1,  
  ...  
)
```

### Arguments

object	an object of class survstan
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).
cores	number of cores to be used in the bootstrap sampling; default is 1 core;
...	further arguments passed to or from other methods.

### Value

the crossing survival time

### Examples

```
library(survstan)  
data(ipass)  
fit <- ypreg(Surv(time, status)~arm, data=ipass, baseline = "weibull")  
summary(fit)  
newdata1 <- data.frame(arm=0)  
newdata2 <- data.frame(arm=1)  
tcross <- cross_time(fit, newdata1, newdata2, nboot = 10)  
tcross
```

---

ehreg

*Fitting Extended Hazard Models*

---

## Description

Function to fit Extended Hazard (EH) models.

## Usage

```
ehreg(formula, data, baseline = "weibull", dist = NULL, init = 0, ...)
```

## 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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.
baseline	the chosen baseline distribution; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distribution (for compatibility with the <a href="#">survreg</a> function); default is NULL.
init	initial values specification (default value is 0); see the detailed documentation for <code>init</code> in <a href="#">optimizing</a> .
...	further arguments passed to other methods.

## Value

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

## Examples

```
library(survstan)
fit <- ehreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
summary(fit)
```

---

 emmeans-survstan-helpers

*Support Functions for **emmeans***


---

### Description

Functions required for compatibility of **survstan** with **emmeans**. Users are not required to call these functions themselves. Instead, they will be called automatically by the **emmeans** function of the **emmeans** package.

### Usage

```
recover_data.survstan(object, ...)
```

```
recover_data.ypreg(object, term = c("short", "long"), ...)
```

```
recover_data.ehreg(object, term = c("AF", "RH"), ...)
```

### Arguments

object	An object of the same class as is supported by a new method.
...	Additional parameters that may be supported by the method.
term	character specifying whether AF or RH term regression coefficients are to be used.

---

 estimates

*Parameters estimates of a survstan model*


---

### Description

Parameters estimates of a survstan model

### Usage

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

### Arguments

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

### Value

the parameters estimates of a given survstan model.

## Examples

```
library(survstan)
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
estimates(fit)
```

---

extractAIC.survstan     *Extract AIC from a Fitted Model*

---

## Description

Computes the (generalized) Akaike An Information Criterion for a fitted parametric model.

## Usage

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

## Arguments

fit	a fitted model of the class survstan
scale	optional numeric specifying the scale parameter of the model. Currently only used in the "lm" method, where scale specifies the estimate of the error variance, and scale = 0 indicates that it is to be estimated by maximum likelihood.
k	numeric specifying the 'weight' of the equivalent degrees of freedom part in the AIC formula.
...	further arguments passed to or from other methods.

## Value

the ANOVA table.

## Examples

```
library(survstan)
fit1 <- aftreg(Surv(futime, fustat) ~ 1, data = ovarian, baseline = "weibull", init = 0)
fit2 <- aftreg(Surv(futime, fustat) ~ rx, data = ovarian, baseline = "weibull", init = 0)
fit3 <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
extractAIC(fit1)
extractAIC(fit2)
extractAIC(fit3)
```

---

gastric	<i>Gastric cancer data set</i>
---------	--------------------------------

---

**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.

---

ggprentice	<i>The Generalized Gamma Distribution (Prentice's alternative parametrization)</i>
------------	--

---

**Description**

Probability function, distribution function, quantile function and random generation for the distribution with parameters mu, sigma and varphi.

**Usage**

```
dggprentice(x, mu, sigma, varphi, log = FALSE)
pggprentice(q, mu = 0, sigma = 1, varphi, lower.tail = TRUE, log.p = FALSE)
qggprentice(p, mu = 0, sigma = 1, varphi, lower.tail = TRUE, log.p = FALSE)
rggprentice(n, mu = 0, sigma = 1, varphi, ...)
```

**Arguments**

<code>x</code>	vector of (non-negative integer) quantiles.
<code>mu</code>	location parameter of the distribution.
<code>sigma</code>	scale parameter of the distribution ( $\sigma > 0$ ).
<code>varphi</code>	shape parameter of the distribution.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$ .
<code>q</code>	vector of quantiles.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ ; otherwise, $P[X > x]$ .
<code>p</code>	vector of probabilities.
<code>n</code>	number of random values to return.
<code>...</code>	further arguments passed to other methods.

**Details**

Probability density function:

$$f(x|\mu, \sigma, \varphi) = \begin{cases} \frac{|\varphi|(\varphi^{-2})^{\varphi-2}}{\sigma x \Gamma(\varphi^{-2})} \exp\{\varphi^{-2}[\varphi w - \exp(\varphi w)]\} I_{[0, \infty)}(x), & \varphi \neq 0 \\ \frac{1}{\sqrt{2\pi x \sigma}} \exp\left\{-\frac{1}{2} \left(\frac{\log(x) - \mu}{\sigma}\right)^2\right\} I_{[0, \infty)}(x), & \varphi = 0 \end{cases}$$

where  $w = \frac{\log(x) - \mu}{\sigma}$ , for  $-\infty < \mu < \infty$ ,  $\sigma > 0$  and  $-\infty < \varphi < \infty$ .

Distribution function:

$$F(x|\mu, \sigma, \varphi) = \begin{cases} F_G(y|1/\varphi^2, 1), & \varphi > 0 \\ 1 - F_G(y|1/\varphi^2, 1), & \varphi < 0 \\ F_{LN}(x|\mu, \sigma), & \varphi = 0 \end{cases}$$

where  $y = \left(\frac{x}{\sigma}\right)^\varphi$ ,  $F_G(\cdot|\nu, 1)$  is the distribution function of a gamma distribution with shape parameter  $1/\varphi^2$  and scale parameter equals to 1, and  $F_{LN}(x|\mu, \sigma)$  corresponds to the distribution function of a lognormal distribution with location parameter  $\mu$  and scale parameter  $\sigma$ .

**Value**

`dggprentice` gives the (log) probability function, `pggprentice` gives the (log) distribution function, `qggprentice` gives the quantile function, and `rggprentice` generates random deviates.

---

ggresiduals                      *Generic S3 method ggresiduals*

---

**Description**

Generic S3 method ggresiduals

**Usage**

```
ggresiduals(object, ...)
```

**Arguments**

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

**Details**

Generic method to plot residuals of survival models.

**Value**

the desired residual plot.

---

ggresiduals.survstan    *ggresiduals method for survstan models*

---

**Description**

ggresiduals method for survstan models

**Usage**

```
## S3 method for class 'survstan'  
ggresiduals(object, type = c("coxsnell", "martingale", "deviance"), ...)
```

**Arguments**

object                      a fitted model object of the class survstan.  
type                          type of residuals used in the plot: coxsnell (default), martingale and deviance.  
...                          further arguments passed to or from other methods.

**Details**

This function produces residuals plots of Cox-Snell residuals, martingale residuals and deviance residuals.

**Value**

the desired residual plot.

**Examples**

```
library(survstan)
ovarian$rx <- as.factor(ovarian$rx)
fit <- aftreg(Surv(futime, fustat) ~ age + rx, data = ovarian, baseline = "weibull", init = 0)
ggresiduals(fit, type = "coxsnell")
ggresiduals(fit, type = "martingale")
ggresiduals(fit, type = "deviance")
```

---

ggstacy

*The Generalized Gamma Distribution (Stacy's original parametrization)*


---

**Description**

Probability function, distribution function, quantile function and random generation for the distribution with parameters alpha, gamma and kappa.

**Usage**

```
dggstacy(x, alpha, gamma, kappa, log = FALSE)
```

```
pggstacy(q, alpha, gamma, kappa, log.p = FALSE, lower.tail = TRUE)
```

```
qggstacy(
  p,
  alpha = 1,
  gamma = 1,
  kappa = 1,
  log.p = FALSE,
  lower.tail = TRUE,
  ...
)
```

```
rggstacy(n, alpha = 1, gamma = 1, kappa = 1, ...)
```

**Arguments**

x	vector of (non-negative integer) quantiles.
alpha	shape parameter of the distribution (alpha > 0).
gamma	scale parameter of the distribution (gamma > 0).

<code>kappa</code>	shape parameter of the distribution ( $\kappa > 0$ ).
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$ .
<code>q</code>	vector of quantiles.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ ; otherwise, $P[X > x]$ .
<code>p</code>	vector of probabilities.
<code>...</code>	further arguments passed to other methods.
<code>n</code>	number of random values to return.

### Details

Probability density function:

$$f(x|\alpha, \gamma, \kappa) = \frac{\kappa}{\gamma^\alpha \Gamma(\alpha/\kappa)} x^{\alpha-1} \exp \left\{ - \left( \frac{x}{\gamma} \right)^\kappa \right\} I_{[0, \infty)}(x),$$

for  $\alpha > 0$ ,  $\gamma > 0$  and  $\kappa > 0$ .

Distribution function:

$$F(t|\alpha, \gamma, \kappa) = F_G(x|\nu, 1),$$

where  $x = \left( \frac{t}{\gamma} \right)^\kappa$ , and  $F_G(\cdot|\nu, 1)$  corresponds to the distribution function of a gamma distribution with shape parameter  $\nu = \alpha/\gamma$  and scale parameter equals to 1.

### Value

`dggstacy` gives the (log) probability function, `pggstacy` gives the (log) distribution function, `qgstacy` gives the quantile function, and `rggstacy` generates random deviates.

---

Gompertz

*The Gompertz Distribution*

---

### Description

Probability function, distribution function, quantile function and random generation for the distribution with parameters alpha and gamma.

### Usage

```
dggompertz(x, alpha = 1, gamma = 1, log = FALSE, ...)
```

```
pgompertz(q, alpha = 1, gamma = 1, lower.tail = TRUE, log.p = FALSE, ...)
```

```
qggompertz(p, alpha = 1, gamma = 1, lower.tail = FALSE, log.p = FALSE, ...)
```

```
rgompertz(n, alpha = 1, gamma = 1, ...)
```

**Arguments**

<code>x</code>	vector of (non-negative integer) quantiles.
<code>alpha</code>	shape parameter of the distribution ( $\alpha > 0$ ).
<code>gamma</code>	scale parameter of the distribution ( $\gamma > 0$ ).
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$ .
<code>...</code>	further arguments passed to other methods.
<code>q</code>	vector of quantiles.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ ; otherwise, $P[X > x]$ .
<code>p</code>	vector of probabilities.
<code>n</code>	number of random values to return.

**Details**

Probability density function:

$$f(x|\alpha, \gamma) = \alpha\gamma \exp\{\gamma x - \alpha(e^{\gamma x} - 1)\}I_{[0, \infty)}(x),$$

for  $\alpha > 0$  and  $\gamma > 0$ .

Distribution function:

$$F(x|\alpha, \gamma) = 1 - \exp\{-\alpha(e^{\gamma x} - 1)\},$$

for  $x > 0$ ,  $\alpha > 0$  and  $\gamma > 0$ .

**Value**

`dgompertz` gives the (log) probability function, `pgompertz` gives the (log) distribution function, `qgompertz` gives the quantile function, and `rgompertz` generates random deviates.

---

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.survstan

*Extract Log-Likelihood from a Fitted Model*

---

**Description**

Extracts the log-likelihood function for a fitted parametric model.

**Usage**

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

**Arguments**

object	a fitted model of the class survstan
...	further arguments passed to or from other methods.

**Value**

the log-likelihood value when a single model is passed to the function; otherwise, a data.frame with the log-likelihood values and the number of parameters is returned.

**Examples**

```
library(survstan)  
fit1 <- aftreg(Surv(futime, fustat) ~ 1, data = ovarian, baseline = "weibull", init = 0)  
fit2 <- aftreg(Surv(futime, fustat) ~ rx, data = ovarian, baseline = "weibull", init = 0)  
fit3 <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)  
logLik(fit1, fit2, fit3)
```

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

---

**Description**

Reconstruct the model matrix for a survstan model.

**Usage**

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

**Arguments**

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

**Value**

The model matrix (or matrices) for the fit.

**Examples**

```
library(survstan)  
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)  
model.matrix(fit)
```

---

`phreg` *Fitting Proportional Hazards Models*

---

**Description**

Function to fit proportional hazards (PH) models.

**Usage**

```
phreg(formula, data, baseline = "weibull", dist = NULL, init = 0, ...)
```

**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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.
baseline	the chosen baseline distribution; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distribution (for compatibility with the <code>survreg</code> function); default is NULL.
init	initial values specification (default value is 0); see the detailed documentation for <code>init</code> in <code>optimizing</code> .
...	further arguments passed to other methods.

**Value**

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

**Examples**

```
library(survstan)
fit <- phreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
summary(fit)
```

---

poreg

*Fitting Proportional Odds Models*

---

**Description**

Function to fit proportional odds (PO) models.

**Usage**

```
poreg(formula, data, baseline = "weibull", dist = NULL, init = 0, ...)
```

**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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.
baseline	the chosen baseline distribution; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distribution (for compatibility with the <code>survreg</code> function); default is NULL.
init	initial values specification (default value is 0); see the detailed documentation for <code>init</code> in <code>optimizing</code> .
...	further arguments passed to other methods.

**Value**

`poreg` returns an object of class "poreg" containing the fitted model.

**Examples**

```
library(survstan)
fit <- poreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
summary(fit)
```

---

```
print.summary.survstan
```

*Print the summary.survstan output*

---

**Description**

Produces a printed summary of a fitted `survstan` model.

**Usage**

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

**Arguments**

x	an object of the class <code>summary.survstan</code> .
...	further arguments passed to or from other methods.

**Value**

No return value, called for side effects.

---

rank_models	<i>Rank a collection of survstan models</i>
-------------	---

---

**Description**

Rank a collection of survstan models

**Usage**

```
rank_models(formula, data, survreg, baseline, dist = NULL, ...)
```

**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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.
survreg	survival regression models to be fitted (AFT, AH, PH, PO, YP and EH).
baseline	baseline distributions to be fitted; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distributions (for compatibility with the <a href="#">survreg</a> function); default is NULL.
...	further arguments passed to other methods.

**Value**

a tibble containing the fitted models ranked according to their AICs.

**Examples**

```
library(survstan)
library(dplyr)

veteran <- veteran %>%
  mutate(across(c(trt, prior, celltype), as.factor))
fits <- rank_models(
  formula = Surv(time, status) ~ celltype+karno,
  data = veteran,
  survreg = c("aftreg", "ahreg", "phreg", "poreg", "ypreg", "ehreg"),
  baseline = c("exponential", "weibull", "lognormal", "loglogistic", "fatigue", "gamma", "rayleigh")
)
```

)

---

residuals.survstan      *residuals method for survstan models*

---

**Description**

residuals method for survstan models

**Usage**

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

**Arguments**

object	a fitted model object of the class survstan.
type	type of residuals desired: coxsnell (default), martingale and deviance.
...	further arguments passed to or from other methods.

**Details**

This function extracts the residuals, martingale residuals and deviance residuals of a survstan object.

**Value**

a vector containing the desired residuals.

**Examples**

```
library(survstan)  
ovarian$rx <- as.factor(ovarian$rx)  
fit <- aftreg(Surv(futime, fustat) ~ age + rx, data = ovarian, baseline = "weibull", init = 0)  
residuals(fit, type = "coxsnell")  
residuals(fit, type = "martingale")  
residuals(fit, type = "deviance")
```

---

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

---

**Description**

Generic S3 method se

**Usage**

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

**Arguments**

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

**Value**

the standard errors associated with a set of parameter estimators for a given model.

---

se.survstan	<i>Estimated standard errors</i>
-------------	----------------------------------

---

**Description**

Estimated standard errors

**Usage**

```
## S3 method for class 'survstan'
se(object, ...)
```

**Arguments**

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

**Value**

a vector with the standard errors.

**Examples**

```
library(survstan)
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
se(fit)
```

---

summary.survstan	<i>Summary for a survstan object</i>
------------------	--------------------------------------

---

**Description**

Summary for a survstan object

**Usage**

```
## S3 method for class 'survstan'
summary(object, conf.level = 0.95, ...)
```

**Arguments**

object	the result of a call to summary.survstan
conf.level	the confidence level required.
...	further arguments passed to or from other methods.

**Value**

an object of the class summary.survstan containing a summary of the fitted model.

---

survfit.survstan	<i>survfit method for survstan models</i>
------------------	---

---

**Description**

Computes the predicted survivor function for a phpe model.

**Usage**

```
## S3 method for class 'survstan'
survfit(formula, newdata = NULL, ...)
```

**Arguments**

formula	an object of the class survstan
newdata	a data frame containing the set of explanatory variables; if NULL, a data.frame with the observed failure times and their corresponding estimated baseline survivals is returned.
...	further arguments passed to or from other methods.

**Value**

a data.frame containing the estimated survival probabilities.

**Examples**

```

library(survstan)
library(ggplot2)
data(ipass)
ipass$arm <- as.factor(ipass$arm)
fit <- ypreg(Surv(time, status)~arm, data=ipass, baseline = "weibull")
summary(fit)
newdata <- data.frame(arm=as.factor(0:1))
surv <- survfit(fit, newdata)
ggplot(surv, aes(x=time, y=surv, color = arm)) +
  geom_line()

```

---

tidy.survstan	<i>Tidy a survstan object</i>
---------------	-------------------------------

---

**Description**

Tidy a survstan object

**Usage**

```

## S3 method for class 'survstan'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)

```

**Arguments**

x	a fitted model object.
conf.int	Logical indicating whether or not to include a confidence interval in the tidied output. Defaults to FALSE.
conf.level	the confidence level required.
...	further arguments passed to or from other methods.

**Details**

Convert a fitted model into a tibble.

**Value**

a tibble with a summary of the fit.

## Examples

```
library(survstan)
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
tidy(fit)
```

---

vcov.survstan	<i>Variance-covariance matrix</i>
---------------	-----------------------------------

---

## 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 'survstan'
vcov(object, all = FALSE, ...)
```

## Arguments

object	an object of the class survstan.
all	logical; if FALSE (default), only covariance matrix associated with regression coefficients is returned; if TRUE, the full covariance matrix is returned.
...	further arguments passed to or from other methods.

## Value

the variance-covariance matrix associated with the parameters estimators.

## Examples

```
library(survstan)
fit <- aftreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull", init = 0)
vcov(fit)
```

---

ypreg

*Fitting Yang and Prentice Models*

---

### Description

Function to fit Yang and Prentice (YP) models.

### Usage

```
ypreg(formula, data, baseline = "weibull", dist = NULL, init = 0, ...)
```

### 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	data an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which function is called.
baseline	the chosen baseline distribution; options currently available are: exponential, weibull, lognormal, loglogistic and Birnbaum-Saunders (fatigue) distributions.
dist	alternative way to specify the baseline distribution (for compatibility with the <a href="#">survreg</a> function); default is NULL.
init	initial values specification (default value is 0); see the detailed documentation for <code>init</code> in <a href="#">optimizing</a> .
...	further arguments passed to other methods.

### Value

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

### Examples

```
library(survstan)
fit <- ypreg(Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian, baseline = "weibull")
summary(fit)
```

# Index

- \* **Continuous**
  - ggprentice, 13
  - ggstacy, 16
  - Gompertz, 17
- \* **Univariate**
  - ggprentice, 13
  - ggstacy, 16
  - Gompertz, 17
- \* **datasets**
  - gastric, 13
  - ipass, 18
- \* **distribution**
  - ggprentice, 13
  - ggstacy, 16
  - Gompertz, 17
- aftreg, 3
- ahreg, 4
- AIC.survstan, 5
- anova.survstan, 6
  
- coef.survstan, 7
- confint.survstan, 7
- cross\_time, 8
- cross\_time.survstan, 9
  
- dggprentice (ggprentice), 13
- dggstacy (ggstacy), 16
- dgompertz (Gompertz), 17
  
- ehreg, 10
- emmeans-survstan-helpers, 11
- estimates, 11
- extractAIC.survstan, 12
  
- gastric, 13
- ggprentice, 13
- ggresiduals, 15
- ggresiduals.survstan, 15
- ggstacy, 16
- Gompertz, 17
  
- gompertz (Gompertz), 17
  
- Hgompertz (Gompertz), 17
- hgompertz (Gompertz), 17
  
- ipass, 18
  
- logLik.survstan, 19
  
- model.matrix.survstan, 20
  
- optimizing, 4, 5, 10, 21, 22, 29
  
- pggprentice (ggprentice), 13
- pggstacy (ggstacy), 16
- pgompertz (Gompertz), 17
- phreg, 20
- poreg, 21
- print.summary.survstan, 22
  
- qggprentice (ggprentice), 13
- qggstacy (ggstacy), 16
- qgompertz (Gompertz), 17
  
- rank\_models, 23
- recover\_data.ehreg
  - (emmeans-survstan-helpers), 11
- recover\_data.survstan
  - (emmeans-survstan-helpers), 11
- recover\_data.ypreg
  - (emmeans-survstan-helpers), 11
- residuals.survstan, 24
- rggprentice (ggprentice), 13
- rggstacy (ggstacy), 16
- rgompertz (Gompertz), 17
  
- se, 25
- se.survstan, 25
- summary.survstan, 26
- survfit.survstan, 26
- survreg, 4, 5, 10, 21–23, 29

survstan (survstan-package), [3](#)  
survstan-package, [3](#)

tidy.survstan, [27](#)

vcov.survstan, [28](#)

ypreg, [29](#)