

Package ‘splitpopsurv’

October 5, 2026

Type Package

Title Split-Population (Cure / Mover-Stayer) Survival Models

Version 0.1.0

Description Maximum-likelihood estimation of split-population (cure / mover-stayer) survival models: an accelerated failure-time regression for event timing among “movers”, combined with a logistic regression on the probability of belonging to the immune “stayer” population. Five baseline timing distributions are provided -- log-logistic, Weibull, log-normal, gamma, and the generalized gamma that nests the other four -- following Schmidt & Witte (1989, Journal of Econometrics) and Yamaguchi (1992, 1998, Sociological Methodology). This is an R translation of a set of 'Stata' ml programs, with the log-likelihood corrected to match the published model and verified by simulation against known parameters.

License MIT + file LICENSE

Encoding UTF-8

Imports maxLik, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/nobifukuda/splitpopsurv>

BugReports <https://github.com/nobifukuda/splitpopsurv/issues>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Nobutaka Fukuda [aut, cre]

Maintainer Nobutaka Fukuda <nobutaka.fukuda@tohoku.ac.jp>

Repository CRAN

Date/Publication 2026-10-05 15:30:02 UTC

Contents

fit_splitpop_gamma	2
fit_splitpop_ggamma	3
fit_splitpop_loglogistic	5
fit_splitpop_lognormal	6
fit_splitpop_weibull	8
loglik_splitpop_gamma	9
loglik_splitpop_ggamma	10
loglik_splitpop_loglogistic	10
loglik_splitpop_lognormal	11
loglik_splitpop_weibull	12

Index	13
-------	----

fit_splitpop_gamma	<i>Fit a split-population gamma survival model</i>
--------------------	--

Description

Maximum-likelihood estimation of a split-population (cure) survival model with a gamma baseline for the "mover" population. R translation of the Stata program ‘SphGam’; see ‘docs/manual.html’ for the model and a note on a likelihood correction relative to the original Stata code.

Usage

```
fit_splitpop_gamma(  
  hform,  
  pform,  
  data,  
  time,  
  event,  
  group,  
  start = NULL,  
  method = "NR"  
)
```

Arguments

hform	One-sided formula (e.g. ‘~ x1 + x2’) for the H_regression equation: covariates for the log-logistic timing distribution.
pform	One-sided formula for the P_regression equation: covariates for the logit on the stayer ("cure") probability.
data	A ‘data.frame’ containing ‘time’, ‘event’, ‘group’, and every variable referenced in ‘hform’/‘pform’.
time	Character: name of the duration column (\$ML_y1 in Stata).

event	Character: name of the 0/1 event column, 1 = failure observed, 0 = censored (\$ML_y2 in Stata).
group	Character: name of a 0/1 column that flips which side of the cure-probability logit is used (\$ML_y3 in Stata) – see the manual. If there is no such distinction in your data, pass a column of all 1s.
start	Optional numeric starting vector; if 'NULL', a default is constructed automatically.
method	Optimizer passed to [maxLik::maxLik()]; default "NR" (Newton-Raphson). Try "BFGS" if that fails to converge.

Details

Note: unlike the other four models, the H_regression linear index here is used directly as a rate (no 'exp()' transform), so it must stay positive; the default 'start' sets a positive intercept and zero slopes to keep it safe regardless of covariate values.

Value

A 'maxLik' object; use 'summary()', 'coef()', 'logLik()', 'vcov()'.

Examples

```
set.seed(1)
n <- 150
x1 <- rnorm(n); x2 <- rbinom(n, 1, 0.5)
cured <- rbinom(n, 1, plogis(0.2 + 0.5 * x2))
t_latent <- (-log(runif(n)) / exp(0.5 - 0.4 * x1))^(1 / 1.3)
censor_time <- rexp(n, rate = 0.2)
time <- pmax(ifelse(cured == 1, censor_time, pmin(t_latent, censor_time)), 1e-3)
event <- ifelse(cured == 1, 0, as.numeric(t_latent <= censor_time))
mydata <- data.frame(time = time, event = event, x1 = x1, x2 = x2, group = 1)

fit <- fit_splitpop_gamma(~x1, ~x2, mydata,
                        "time", "event", "group", method = "BFGS")

summary(fit)
```

fit_splitpop_ggamma	<i>Fit a split-population generalized gamma survival model</i>
---------------------	--

Description

Maximum-likelihood estimation of a split-population (cure) survival model with a generalized gamma baseline (Prentice 1974 / Yamaguchi & Ferguson 1995, note 10) for the "mover" population – the family that nests the Weibull ($\kappa=1$), log-normal ($\kappa=0$), and gamma ($\sigma=1$) models above. R translation of the Stata program 'SphGGam' ('d0' method); see 'docs/manual.html' for the model and a note on a likelihood correction relative to the original Stata code.

Usage

```
fit_splitpop_ggamma(
  hform,
  pform,
  data,
  time,
  event,
  group,
  start = NULL,
  method = "NR"
)
```

Arguments

hform	One-sided formula (e.g. <code>'~ x1 + x2'</code>) for the H_regression equation: covariates for the log-logistic timing distribution.
pform	One-sided formula for the P_regression equation: covariates for the logit on the stayer ("cure") probability.
data	A 'data.frame' containing 'time', 'event', 'group', and every variable referenced in 'hform'/'pform'.
time	Character: name of the duration column (\$ML_y1 in Stata).
event	Character: name of the 0/1 event column, 1 = failure observed, 0 = censored (\$ML_y2 in Stata).
group	Character: name of a 0/1 column that flips which side of the cure-probability logit is used (\$ML_y3 in Stata) – see the manual. If there is no such distinction in your data, pass a column of all 1s.
start	Optional numeric starting vector; if 'NULL', a default is constructed automatically.
method	Optimizer passed to [maxLik::maxLik()]; default "NR" (Newton-Raphson). Try "BFGS" if that fails to converge.

Details

Note: `'lkappal < 0.01'` switches to a log-normal-equivalent branch, a genuine kink in the likelihood surface (preserved from the Stata guard). If `'summary()'` reports a non-finite standard error, the optimizer likely landed near that threshold – try a different starting 'kappa'.

Value

A 'maxLik' object; use `'summary()'`, `'coef()'`, `'logLik()'`, `'vcov()'`.

Examples

```
set.seed(1)
n <- 150
x1 <- rnorm(n); x2 <- rbinom(n, 1, 0.5)
cured <- rbinom(n, 1, plogis(0.2 + 0.5 * x2))
```

```

t_latent <- (-log(runif(n)) / exp(0.5 - 0.4 * x1))^(1 / 1.3)
censor_time <- rexp(n, rate = 0.2)
time <- pmax(iffelse(cured == 1, censor_time, pmin(t_latent, censor_time)), 1e-3)
event <- iffelse(cured == 1, 0, as.numeric(t_latent <= censor_time))
mydata <- data.frame(time = time, event = event, x1 = x1, x2 = x2, group = 1)

fit <- fit_splitpop_ggamma(~x1, ~x2, mydata,
                           "time", "event", "group", method = "BFGS")
summary(fit)

```

fit_splitpop_loglogistic

Fit a split-population log-logistic survival model

Description

Maximum-likelihood estimation of a split-population (cure) survival model with a log-logistic baseline for the "mover" (susceptible) population and a logistic regression on the probability of being a "stayer" (immune to the event). R translation of the Stata program 'SphLog'; see 'docs/manual.html' for the model, formulas, and a note on a likelihood correction relative to the original Stata code.

Usage

```

fit_splitpop_loglogistic(
  hform,
  pform,
  data,
  time,
  event,
  group,
  start = NULL,
  method = "NR"
)

```

Arguments

hform	One-sided formula (e.g. '~ x1 + x2') for the H_regression equation: covariates for the log-logistic timing distribution.
pform	One-sided formula for the P_regression equation: covariates for the logit on the stayer ("cure") probability.
data	A 'data.frame' containing 'time', 'event', 'group', and every variable referenced in 'hform'/'pform'.
time	Character: name of the duration column (\$ML_y1 in Stata).
event	Character: name of the 0/1 event column, 1 = failure observed, 0 = censored (\$ML_y2 in Stata).

group	Character: name of a 0/1 column that flips which side of the cure-probability logit is used (\$ML_y3 in Stata) – see the manual. If there is no such distinction in your data, pass a column of all 1s.
start	Optional numeric starting vector; if 'NULL', a default is constructed automatically.
method	Optimizer passed to [maxLik::maxLik()]; default "NR" (Newton-Raphson). Try "BFGS" if that fails to converge.

Value

A 'maxLik' object; use 'summary()', 'coef()', 'logLik()', 'vcov()'.

Examples

```
set.seed(1)
n <- 150
x1 <- rnorm(n); x2 <- rbinom(n, 1, 0.5)
cured <- rbinom(n, 1, plogis(0.2 + 0.5 * x2))
t_latent <- (-log(runif(n)) / exp(0.5 - 0.4 * x1))^(1 / 1.3)
censor_time <- rexp(n, rate = 0.2)
time <- pmax(ifelse(cured == 1, censor_time, pmin(t_latent, censor_time)), 1e-3)
event <- ifelse(cured == 1, 0, as.numeric(t_latent <= censor_time))
mydata <- data.frame(time = time, event = event, x1 = x1, x2 = x2, group = 1)

fit <- fit_splitpop_loglogistic(~x1, ~x2, mydata,
                               "time", "event", "group", method = "BFGS")
summary(fit)
```

fit_splitpop_lognormal

Fit a split-population log-normal survival model

Description

Maximum-likelihood estimation of a split-population (cure) survival model with a log-normal baseline for the "mover" population. R translation of the Stata program 'SphNom'; see 'docs/manual.html' for the model and a note on a likelihood correction relative to the original Stata code.

Usage

```
fit_splitpop_lognormal(
  hform,
  pform,
  data,
  time,
  event,
  group,
  start = NULL,
```

```

    method = "NR"
  )

```

Arguments

hform	One-sided formula (e.g. <code>'~ x1 + x2'</code>) for the H_regression equation: covariates for the log-logistic timing distribution.
pform	One-sided formula for the P_regression equation: covariates for the logit on the stayer ("cure") probability.
data	A 'data.frame' containing 'time', 'event', 'group', and every variable referenced in 'hform'/'pform'.
time	Character: name of the duration column (\$ML_y1 in Stata).
event	Character: name of the 0/1 event column, 1 = failure observed, 0 = censored (\$ML_y2 in Stata).
group	Character: name of a 0/1 column that flips which side of the cure-probability logit is used (\$ML_y3 in Stata) – see the manual. If there is no such distinction in your data, pass a column of all 1s.
start	Optional numeric starting vector; if 'NULL', a default is constructed automatically.
method	Optimizer passed to [maxLik::maxLik()]; default "NR" (Newton-Raphson). Try "BFGS" if that fails to converge.

Value

A 'maxLik' object; use 'summary()', 'coef()', 'logLik()', 'vcov()'.

Examples

```

set.seed(1)
n <- 150
x1 <- rnorm(n); x2 <- rbinom(n, 1, 0.5)
cured <- rbinom(n, 1, plogis(0.2 + 0.5 * x2))
t_latent <- (-log(runif(n)) / exp(0.5 - 0.4 * x1))^(1 / 1.3)
censor_time <- rexp(n, rate = 0.2)
time <- pmax(iffelse(cured == 1, censor_time, pmin(t_latent, censor_time)), 1e-3)
event <- iffelse(cured == 1, 0, as.numeric(t_latent <= censor_time))
mydata <- data.frame(time = time, event = event, x1 = x1, x2 = x2, group = 1)

fit <- fit_splitpop_lognormal(~x1, ~x2, mydata,
                             "time", "event", "group", method = "BFGS")
summary(fit)

```

fit_splitpop_weibull *Fit a split-population Weibull survival model*

Description

Maximum-likelihood estimation of a split-population (cure) survival model with a Weibull baseline for the "mover" population. R translation of the Stata program 'SphWieb'; see 'docs/manual.html' for the model and a note on a likelihood correction relative to the original Stata code.

Usage

```
fit_splitpop_weibull(
  hform,
  pform,
  data,
  time,
  event,
  group,
  start = NULL,
  method = "NR"
)
```

Arguments

hform	One-sided formula (e.g. '~ x1 + x2') for the H_regression equation: covariates for the log-logistic timing distribution.
pform	One-sided formula for the P_regression equation: covariates for the logit on the stayer ("cure") probability.
data	A 'data.frame' containing 'time', 'event', 'group', and every variable referenced in 'hform'/'pform'.
time	Character: name of the duration column (\$ML_y1 in Stata).
event	Character: name of the 0/1 event column, 1 = failure observed, 0 = censored (\$ML_y2 in Stata).
group	Character: name of a 0/1 column that flips which side of the cure-probability logit is used (\$ML_y3 in Stata) – see the manual. If there is no such distinction in your data, pass a column of all 1s.
start	Optional numeric starting vector; if 'NULL', a default is constructed automatically.
method	Optimizer passed to [maxLik::maxLik()]; default "NR" (Newton-Raphson). Try "BFGS" if that fails to converge.

Value

A 'maxLik' object; use 'summary()', 'coef()', 'logLik()', 'vcov()'.

Examples

```

set.seed(1)
n <- 150
x1 <- rnorm(n); x2 <- rbinom(n, 1, 0.5)
cured <- rbinom(n, 1, plogis(0.2 + 0.5 * x2))
t_latent <- (-log(runif(n)) / exp(0.5 - 0.4 * x1))^(1 / 1.3)
censor_time <- rexp(n, rate = 0.2)
time <- pmax(ifelse(cured == 1, censor_time, pmin(t_latent, censor_time)), 1e-3)
event <- ifelse(cured == 1, 0, as.numeric(t_latent <= censor_time))
mydata <- data.frame(time = time, event = event, x1 = x1, x2 = x2, group = 1)

fit <- fit_splitpop_weibull(~x1, ~x2, mydata,
                           "time", "event", "group", method = "BFGS")
summary(fit)

```

loglik_splitpop_gamma *Log-likelihood: split-population gamma model*

Description

Log-likelihood: split-population gamma model

Usage

```
loglik_splitpop_gamma(par, time, event, group, Xh, Xp)
```

Arguments

par	Numeric parameter vector: ‘H_regression’ coefficients, the ancillary scalar ‘theta2’, then ‘P_regression’ coefficients.
time, event, group	Numeric vectors: duration, 0/1 event indicator, 0/1 group indicator (see [fit_splitpop_loglogistic()]).
Xh, Xp	Design matrices for the H_regression and P_regression parts.

Value

A numeric vector of per-observation log-likelihood contributions.

Examples

```

Xh <- cbind(1, rnorm(20)); Xp <- cbind(1, rbinom(20, 1, 0.5))
par0 <- c(1, 0, 1, 0, 0) # positive H_regression intercept: theta1 must stay > 0
loglik_splitpop_gamma(par0, time = runif(20, 0.1, 5),
                      event = rbinom(20, 1, 0.7),
                      group = rep(1, 20), Xh = Xh, Xp = Xp)

```

loglik_splitpop_ggamma

Log-likelihood: split-population generalized gamma model

Description

Log-likelihood: split-population generalized gamma model

Usage

```
loglik_splitpop_ggamma(par, time, event, group, Xh, Xp)
```

Arguments

par	Numeric parameter vector: 'H_regression' coefficients, 'ln_sigma', 'kappa', then 'P_regression' coefficients.
time, event, group	Numeric vectors: duration, 0/1 event indicator, 0/1 group indicator (see [fit_splitpop_loglogistic()]).
Xh, Xp	Design matrices for the H_regression and P_regression parts.

Value

A numeric vector of per-observation log-likelihood contributions.

Examples

```
Xh <- cbind(1, rnorm(20)); Xp <- cbind(1, rbinom(20, 1, 0.5))
par0 <- c(0, 0, 0, 1, 0, 0)
loglik_splitpop_ggamma(par0, time = runif(20, 0.1, 5),
  event = rbinom(20, 1, 0.7),
  group = rep(1, 20), Xh = Xh, Xp = Xp)
```

loglik_splitpop_loglogistic

Log-likelihood: split-population log-logistic model

Description

Internal log-likelihood used by [fit_splitpop_loglogistic()]. Exposed so it can be evaluated by hand at candidate starting values (the R analogue of Stata's 'ml check').

Usage

```
loglik_splitpop_loglogistic(par, time, event, group, Xh, Xp)
```

Arguments

par	Numeric parameter vector: ‘H_regression’ coefficients, the ancillary scalar ‘theta2’, then ‘P_regression’ coefficients.
time, event, group	Numeric vectors: duration, 0/1 event indicator, 0/1 group indicator (see [fit_splitpop_loglogistic()]).
Xh, Xp	Design matrices for the H_regression and P_regression parts.

Value

A numeric vector of per-observation log-likelihood contributions.

Examples

```
Xh <- cbind(1, rnorm(20)); Xp <- cbind(1, rbinom(20, 1, 0.5))
par0 <- c(0, 0, 1, 0, 0)
loglik_splitpop_loglogistic(par0, time = runif(20, 0.1, 5),
                             event = rbinom(20, 1, 0.7),
                             group = rep(1, 20), Xh = Xh, Xp = Xp)
```

loglik_splitpop_lognormal

Log-likelihood: split-population log-normal model

Description

Log-likelihood: split-population log-normal model

Usage

```
loglik_splitpop_lognormal(par, time, event, group, Xh, Xp)
```

Arguments

par	Numeric parameter vector: ‘H_regression’ coefficients, the ancillary scalar ‘theta2’, then ‘P_regression’ coefficients.
time, event, group	Numeric vectors: duration, 0/1 event indicator, 0/1 group indicator (see [fit_splitpop_loglogistic()]).
Xh, Xp	Design matrices for the H_regression and P_regression parts.

Value

A numeric vector of per-observation log-likelihood contributions.

[illegible]

Log-likelihood: split-population Weibull model

[illegible]

Index

`fit_splitpop_gamma`, [2](#)
`fit_splitpop_ggamma`, [3](#)
`fit_splitpop_loglogistic`, [5](#)
`fit_splitpop_lognormal`, [6](#)
`fit_splitpop_weibull`, [8](#)

`loglik_splitpop_gamma`, [9](#)
`loglik_splitpop_ggamma`, [10](#)
`loglik_splitpop_loglogistic`, [10](#)
`loglik_splitpop_lognormal`, [11](#)
`loglik_splitpop_weibull`, [12](#)