Package: flexsurvPlus (via r-universe)

November 12, 2026

Title Provides Functions to Perform Survival Analysis for Economic models

Version 1.06

Description A group of functions designed to perform survival analyses.

License Apache License (>= 2)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

VignetteBuilder knitr

Imports assertthat (>= 0.2.1), dplyr (>= 0.8.4), magrittr (>= 1.5), survival (>= 3.1), tibble (>= 3.0.0), stats (>= 2.0)

Depends R (>= 2.10), flexsurv

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggplot2 (>= 3.3.0), boot (>= 1.3-24), pander, tidyr

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev

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

Date/Publication 2023-05-05 07:15:28 UTC

RemoteUrl https://github.com/Roche/flexsurvPlus

RemoteRef HEAD

RemoteSha e834c452017f3c891d761e27d9cb3f7b4c0b8f2d

Contents

bootPSM 2
convSTEM 3
flexsurvPlus 4
runPSM 4

sim_adtte 6
summaryKM 8
summaryPSM 10

Index **13**

`bootPSM` *Run a complete parametric survival analysis*

Description

This function is a wrapper for the `runPSM` function intended to be run with the `boot` package. It enables running a complete parametric survival analysis for use when performing bootstrapping to explore uncertainty. By re-using random seeds for each bootstrap sample it is possible to maintain correlations across multiple endpoints.

Usage

```
bootPSM(data, i, ...)
```

Arguments

`data`	A data frame containing individual patient data for the relevant time to event outcomes. This is passed to the `data` argument of the `runPSM` function
`i`	Index used to select a sample within `boot`.
`...`	Additional parameters as used by `runPSM`

Details

For more details and examples see the package vignettes:

- `vignette("Fitting_models_in_R", package = "flexsurvPlus")`
- `vignette("Bootstrap_models_in_R", package = "flexsurvPlus")`
- `vignette("Model_theory", package = "flexsurvPlus")`

This function is intended to be used in conjunction with the `boot` function to return the statistic to be bootstrapped. In this case by performing parametric survival modelling using flexsurv and returning the parameters of the survival distributions. This is used as the 'statistic' argument in the boot function.

Value

The 'parameters_vector' object from the `runPSM` function.

'parameters_vector' is a vector which contains the coefficients for all of the flexsurv models specified. The column names are in the format 'modeltype.distribution.parameter.TreatmentName', for example, comshp.weibull.shape.Int refers to the shape parameter of the common shape weibull distribution for the intervention treatment and 'indshp.gengamma.mu.ref' refers to the mu parameter of the independent shape generalised gamma distribution for the reference treatment. Columns with

'TE' at the end are the treatment effect coefficients (applicable to the scale and shape parameters for independent shape models, applicable to the scale parameter only for the common shape model and not applicable for the separate model).

`convSTEM` *Convert survival parameters to SAS/STEM parametric forms*

Description

Convert survival parameters to SAS/STEM parametric forms

Usage

```
convSTEM(x = NULL, samples = NULL, use = "everything")
```

Arguments

`x`	An object created by calling `runPSM`
`samples`	An object created by calling `boot` with `bootPSM`
`use`	an optional character string giving a method for computing covariances in the presence of missing values. See `cov` for details. Option "complete.obs" maybe needed when some bootstrap samples do not converge to estimate covariance only using those that do. This function primarily exists for backward compatibility with older excel models where parametric extrapolation was performed with SAS and alternative parametric forms were used for distributions. As such only a subset of models are supported. One or both of `x` and `samples` must be specified and affect what is returned. For more details please see the vignette `vignette("STEM_compatability", package = "flexsurvPlus")` Possible distributions include • Exponential ('exp') • Weibull ('weibull') • Gompertz ('gompertz') • Log-normal ('lnorm') • Log-logistic ('llogis') • Generalized gamma ('gengamma') • Gamma ('gamma')

Value

a list containing 4 data frames

- stem_param Converted parameter estimates
- stem_cov Converted covariance matrix (if `samples` provided)
- stem_modsum Converted model summary (if `x` provided)
- stem_boot Converted bootstrap samples (if `samples` provided)

`flexsurvPlus` *flexsurvPlus: A package for performing survival analyses.*

Description

The flexsurvPlus package provides functions to help perform and summarize results from basic survival analyses

`runPSM` *Run complete parametric survival analysis for multiple models with multiple distributions*

Description

Run complete parametric survival analysis for multiple models with multiple distributions

Usage

```
runPSM(
  data,
  time_var,
  event_var,
  weight_var = "",
  model.type = c("Separate", "Common shape", "Independent shape"),
 distr = c("exp", "weibull", "gompertz", "lnorm", "llogis", "gengamma", "gamma", "genf"),
  strata_var,
  int_name,
  ref_name
)
```

Arguments

`data`	A data frame containing individual patient data for the relevant time to event outcomes.
`time_var`	Name of time variable in 'data'. Variable must be numerical and >0.
`event_var`	Name of event variable in 'data'. Variable must be numerical and contain 1's to indicate an event and 0 to indicate a censor.
`weight_var`	Optional name of a variable in "data" containing case weights.
`model.type`	Character vector indicating the types of model formula provided. Permitted values are

- 'Common shape' a model with a single covariate for the effect of treatment on the scale parameter of the model. The model fit is in the form Surv(Time, Event==1) ~ ARM. The shape parameter is the same for each treatment, and derived directly from the model (no additional manipulation is required). The scale parameter is derived directly from the model for the reference category, however, for the intervention arm, this is calculated as reference shape + treatment effect (shape).
- 'Independent shape' a model with a single covariate for treatment that affects both the scale and shape parameters of the model. The model fit is in the form Surv(Time, Event==1) ~ ARM + shape(ARM). The scale parameter is derived directly from the model for the reference category, however, for the intervention arm, this is calculated as reference scale + treatment effect (scale). The shape parameter is derived directly from the model for the reference category, however, for the intervention arm, this is calculated as reference shape + treatment effect (shape).
- 'Separate' a model with no covariates fitted separately to data from each treatment group in a study. The model fit is in the form Surv(Time, Event==1) ~ 1 and is fit twice (one separate model for each of the two treatments). The parameters for each treatment, are derived directly from the model (no additional manipulation is required).
- 'One arm' a model with no covariates is fitted to the entire data set without a strata variable. The model fit is in the form Surv(Time, Event==1) ~ 1 and is fit to the entire data (no strata). The parameters for each treatment, are derived directly from the model (no additional manipulation is required).

Default is c("Separate", "Common shape", "Independent shape").

`distr` A vector string of distributions, see dist argument in `flexsurvreg`. Default is all available distributions (see below).

`strata_var` Name of stratification variable in "data". This is usually the treatment variable and must be categorical. Not required when model.type='One arm'.

`int_name` Character to indicate the name of the treatment of interest, must be a level of the "strata_var" column in "data", used for labelling the parameters.

`ref_name` Character to indicate the name of the reference treatment, must be a level of the "strata_var" column in "data", used for labelling the parameters. Not required when model.type='One arm'.

Details

Possible distributions include:

- Exponential ('exp')
- Weibull ('weibull')
- Gompertz ('gompertz')
- Log-normal ('lnorm')
- Log-logistic ('llogis')
- Generalized gamma ('gengamma')

- Gamma ('gamma')
- Generalised F ('genf')

For more details and examples see the package vignettes:

- `vignette("Fitting_models_in_R", package = "flexsurvPlus")`
- `vignette("Bootstrap_models_in_R", package = "flexsurvPlus")`
- `vignette("Model_theory", package = "flexsurvPlus")`

Value

A list containing 'models' (models fit using flexsurvreg), 'model_summary' (a tibble containing AIC, BIC and convergence information), 'parameters_vector' (a vector containing the coefficients of each flexsurv model), and 'config' (a list containing information on the function call).

- 'models' is a list of flexsurv objects for each distribution specified
- 'model_summary' is a tibble object containing the fitted model objects, the parameter estimates (`coef`), `AIC` and `BIC` from flexsurv objects.
- 'parameters_vector' is a vector which contains the coefficients for all of the flexsurv models specified. The column names are in the format 'modeltype.distribution.parameter.TreatmentName', for example, comshp.weibull.shape.Int refers to the shape parameter of the common shape weibull distribution for the intervention treatment and 'indshp.gengamma.mu.ref' refers to the mu parameter of the independent shape generalised gamma distribution for the reference treatment. Columns with 'TE' at the end are the treatment effect coefficients (applicable to the scale and shape parameters for independent shape models, applicable to the scale parameter only for the common shape model and not applicable for the separate or one-arm model).

`sim_adtte` *A function to simulate Survival data*

Description

This function simulates survival data with correlated time to progression and overall survival times. Optionally crossover from treatment arms can be simulated.

Usage

```
sim_adtte(
  rc_siminfo = FALSE,
  rc_origos = FALSE,
  id = 1,
  seed = 1234,
  rho = 0,
  pswitch = 0,
  proppd = 0,
  beta_1a = log(0.7),
  beta_1b = log(0.7),
```

```
  beta_2a = log(0.7),
  beta_2b = log(0.7),
  beta_pd = log(0.4),
  arm_n = 250,
  enroll_start = 0,
  enroll_end = 1,
  admin_censor = 2,
  os_gamma = 1.2,
  os_lambda = 0.3,
  ttp_gamma = 1.5,
  ttp_lambda = 2
)
```

Arguments

rc_siminfo	Should simulation params be included in simulated dataframe (logical). Defaults to FALSE.
rc_origos	Should OS without switching be included in simulated dataframe (logical). Defaults to FALSE.
id	Identifer added to simulated dataframe. Defaults to 1.
seed	Seed used for random number generator. Defaults to 1234.
rho	correlation coefficient between TTP and OS. Defaults to 0.
pswitch	proportion of patients who switch. Defaults to 0.
proppd	proportion of patients with PFS before switch allowed. Defaults to 0.
beta_1a	treatment effect (as log(Hazard Ratio)) for OS pre PFS. Defaults to log(0.7).
beta_1b	treatment effect (as log(Hazard Ratio)) for OS post PFS. Defaults to log(0.7).
beta_2a	treatment effect (as log(Hazard Ratio)) for OS pre PFS (switch). Defaults to log(0.7).
beta_2b	treatment effect (as log(Hazard Ratio)) for OS post PFS (switch). Defaults to log(0.7).
beta_pd	treatment effect on progression (as log(HR)). Defaults to log(0.4).
arm_n	patients per arm. Defaults to 250.
enroll_start	start of enrollment. Defaults to 0.
enroll_end	end of enrollment. Defaults to 1.
admin_censor	end of trial. Defaults to 2.
os_gamma	weibull shape - for OS. Defaults to 1.2.
os_lambda	weibull scale - for OS. Defaults to 0.3.
ttp_gamma	weibull shape - for TTP. Defaults to 1.5.
ttp_lambda	weibull scale - for TTP. Defaults to 2.

Details

The simulation times are derived from formulas in Austin 2012 adapted to enable correlations between endpoints. Austin, P.C. (2012), Generating survival times to simulate Cox proportional hazards models with time-varying covariates. Statist. Med., 31: 3946-3958. https://doi.org/10.1002/sim.5452

Examples

```
require(survival)
require(dplyr)

ADTTE <- sim_adtte()
survfit(Surv(AVAL, event = CNSR ==  0) ~ ARM, data = filter(ADTTE, PARAMCD == "PFS")) %>%
  plot()

survfit(Surv(AVAL, event = CNSR ==  0) ~ ARM, data = filter(ADTTE, PARAMCD == "OS")) %>%
  plot()
```

---

summaryKM *Extract information about non-parametric survival models*

Description

Extract information about non-parametric survival models

Usage

```
summaryKM(
  data,
  time_var,
  event_var,
  weight_var = "",
  strata_var,
  int_name,
  ref_name,
  types = c("survival", "cumhaz", "median", "rmst"),
  t = NULL,
  ci = FALSE,
  se = FALSE,
  ...
)
```

Arguments

data	A data frame containing individual patient data for the relevant time to event outcomes.
time_var	Name of time variable in 'data'. Variable must be numerical and >0.
event_var	Name of event variable in 'data'. Variable must be numerical and contain 1's to indicate an event and 0 to indicate a censor.
weight_var	Optional name of a variable in "data" containing case weights.
strata_var	Name of stratification variable in "data". This is usually the treatment variable and must be categorical. Not required if only one arm is being analyzed.

int_name	Character to indicate the name of the treatment of interest, must be a level of the "strata_var" column in "data", used for labelling the parameters.
ref_name	Character to indicate the name of the reference treatment, must be a level of the "strata_var" column in "data", used for labelling the parameters. Not required if only one arm is being analyzed.
types	A list of statistics to extract - options include "survival", "cumhaz", "median", and "rmst". For details see the vignette on descriptive analysis.
t	The time points to be used - this only controls the rmst statistic.
ci	Should a confidence interval be returned (TRUE or FALSE)
se	Should a standard error be returned (TRUE or FALSE)
...	Additional arguments passed to survfit

Value

A data frame containing the following values and similar to that returned by summaryPSM

- Model - returned as "Kaplan Meier"
- ModelF - an ordered factor of Model
- Dist - returned as "Kaplan Meier"
- DistF - an ordered factor of Dist
- distr - returned as "km"
- Strata - Either Intervention or Reference
- StrataName - As specified by int_name and ref_name respectively.
- type - as defined by the types parameter.
- variable - "est", "lcl", "ucl", "se" respectively
- time - either NA or the time the statistic is evaluated at
- value - estimated value

Examples

```
require(dplyr)
require(ggplot2)

PFS_data <- sim_adtte(seed = 2020, rho = 0.6) %>%
filter(PARAMCD=="PFS") %>%
transmute(USUBJID,
            ARMCD,
            PFS_days = AVAL,
            PFS_event = 1- CNSR,
            wt = runif(500,0,1)
)

pfs_info <- summaryKM(
  data = PFS_data,
  time_var = "PFS_days",
  event_var = "PFS_event",
```

```
  strata_var = "ARMCD",
  int_name = "A",
  ref_name = "B",
  ci = TRUE,
  t = c(500, 700))

ggplot(data = filter(pfs_info, type == "survival", variable == "est"),
       aes(x = time, y = value, color = StrataName)) +
  geom_step() +
  geom_step(data = filter(pfs_info, type == "survival", variable == "lcl"), linetype = 2) +
  geom_step(data = filter(pfs_info, type == "survival", variable == "ucl"), linetype = 2) +
  geom_point(data = filter(pfs_info, type == "survival", variable == "censored")) +
  xlab("Time") +
  ylab("Survival") +
  ggtitle("KM estimates and 95% CI")

filter(pfs_info, type == "cumhaz", variable == "est") %>%
  ggplot(aes(x = time, y = value, color = StrataName)) +
  geom_step() +
  xlab("Time") +
  ylab("Cumulative hazard")

filter(pfs_info, type == "median") %>%
  transmute(StrataName, variable, value)

filter(pfs_info, type == "rmst") %>%
  transmute(StrataName, variable, time, value)

# example with weights
 pfs_info_wt <- summaryKM(
   data = PFS_data,
   time_var = "PFS_days",
   event_var = "PFS_event",
   strata_var = "ARMCD",
   weight_var = "wt",
   int_name = "A",
   ref_name = "B",
   types = "survival"
   )

   ggplot(data = filter(pfs_info, type == "survival", variable == "est"),
          aes(x = time, y = value, color = StrataName)) +
     geom_step(aes(linetype = "Original")) +
     geom_step(data = filter(pfs_info_wt, type == "survival", variable == "est"),
               aes(linetype = "Weighted")) +
     xlab("Time") +
     ylab("Survival") +
     ggtitle("KM estimates and 95% CI")
```

---

summaryPSM *Extract information about fitted parametric survival models*

Description

Extract information about fitted parametric survival models

Usage

```
summaryPSM(
  x,
  types = c("mean", "survival", "hazard", "cumhaz", "median", "rmst"),
  t = NULL,
  ci = FALSE,
  se = FALSE
)
```

Arguments

x	An object created by calling runPSM
types	A list of statistics to extract - see summary.flexsurvreg for details
t	The time points to be used - see summary.flexsurvreg for details
ci	Should a confidence interval be returned - see summary.flexsurvreg for details
se	Should a standard error be returned - see summary.flexsurvreg for details

Value

A data frame containing the following values

- Model - The Model as specified in runPSM model.type
- ModelF - an ordered factor of Model
- Dist - The distribution
- DistF - an ordered factor of Dist
- distr - as specified in runPSM distr
- Strata - Either Intervention or Reference
- StrataName - As specified by int_name and ref_name respectively in runPSM
- type - as defined by the types parameter see summary.flexsurvreg for details
- variable - "est", "lcl", "ucl", "se" respectively
- time - either NA or the time the statistic is evaluated at
- value - estimated value

Examples

```
require(dplyr)
require(ggplot2)

PFS_data <- sim_adtte(seed = 2020, rho = 0.6) %>%
filter(PARAMCD=="PFS") %>%
transmute(USUBJID,
```

```
            ARMCD,
            PFS_days = AVAL,
            PFS_event = 1- CNSR
)

psm_pfs <- runPSM(
  data = PFS_data,
  time_var = "PFS_days",
  event_var = "PFS_event",
  strata_var = "ARMCD",
  int_name = "A",
  ref_name = "B")

summaryPSM(psm_pfs, types = c("mean","rmst"), t = c(100,2000)) %>%
   filter(Dist == "Generalized Gamma", Strata == "Intervention")

summaryPSM(psm_pfs, types = "survival", t = seq(0,2000,100)) %>%
  ggplot(aes(x=time, y = value, color = StrataName, linetype = Model)) +
  geom_line()+
  facet_grid(~Dist)

summaryPSM(psm_pfs, types = "hazard", t = seq(0,5000,100)) %>%
  ggplot(aes(x=time, y = value, color = StrataName, linetype = Model)) +
  geom_line()+
  facet_grid(~Dist) +
  coord_cartesian(ylim = c(0,0.02))

summaryPSM(psm_pfs, types = "cumhaz", t = seq(0,5000,100)) %>%
  ggplot(aes(x=time, y = value, color = StrataName, linetype = Model)) +
  geom_line()+
  facet_grid(~Dist) +
  coord_cartesian(ylim = c(0,100))
```

Index

∗ **Simulation**
sim_adtte, 6
∗ **Survival,**
sim_adtte, 6

AIC, *6*

BIC, *6*
boot, *2*, *3*
bootPSM, 2, *3*

coef, *6*
convSTEM, 3
cov, *3*

flexsurv, *2*
flexsurvPlus, 4
flexsurvreg, *5*

runPSM, *2*, *3*, 4, *11*

sim_adtte, 6
summary.flexsurvreg, *11*
summaryKM, 8
summaryPSM, *9*, 10
survfit, *9*