Package 'survHE'

February 9, 2021

Title Survival Analysis in Health Economic Evaluation

Version 1.1.2

Date 2021-02-08

URL https://github.com/giabaio/survHE,
  http://www.statistica.it/gianluca/

BugReports https://github.com/giabaio/survHE/issues

Description Contains a suite of functions for survival analysis in health economics.
  These can be used to run survival models under a frequentist (based on maximum likelihood)
  or a Bayesian approach (both based on Integrated Nested Laplace Approximation or Hamiltonian
  Monte Carlo). The user can specify a set of parametric models using a common notation and
  select the preferred mode of inference. The results can also be post-processed to produce
  probabilistic sensitivity analysis and can be used to export the output to an Excel
  file (e.g. for a Markov model, as often done by modellers and practition-
  ers). <doi:10.18637/jss.v095.i14>.

License GPL (>= 3)

Depends methods, R (>= 3.6.0), Rcpp (>= 0.12.19), flexsurv, dplyr,
  ggplot2

Imports rms, xlsx, tools, rstan (>= 2.18.1), tibble

Suggests shinystan, INLA

LinkingTo BH (>= 1.66.0-1), Rcpp (>= 0.12.19), RcppEigen (>=
  0.3.3.4.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Additional_repositories https://inla.r-inla-download.org/R/stable

Encoding UTF-8

LazyData true

NeedsCompilation yes

SystemRequirements GNU make

RoxygenNote 7.1.1

Author Gianluca Baio [aut, cre]

Maintainer Gianluca Baio <g.baio@ucl.ac.uk>

Repository CRAN

Date/Publication 2021-02-09 00:40:07 UTC

R topics documented:

| survHE-package | *Survival Analysis in Health Economic Evaluation* |

Description

Contains a suite of functions to perform survival analysis with the aim of aiding in health economic modelling (extrapolation, model checking and PSA)

Details

Package:	survHE
Type:	Package
Version:	1.1.2
Date:	2021-02-08
License:	GPL2
LazyLoad:	yes

Contains a suite of functions to perform survival analysis with the aim of aiding in health economic modelling (extrapolation, model checking and PSA)

Author(s)

Gianluca Baio

Maintainer: Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

Examples

```
## Not run:
# Loads some survival data
data(bc)
# Fits a parametric model
m <- fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
# Print output in tabular format
print(m)
# Visualise output in terms of survival curves
plot(m)

## End(Not run)
```

---

data   *A fictional survival trial.*

Description

A dataset containing fictional data from a trial, where the main outcome is in terms of time-to-event and censoring indicator and with additional covariates.

Usage

```
data
```

Format

A data frame with 367 rows and 8 variables:

ID_patient The individual level identifier

time The observed time at which the event happens

censored An indicator to describe whether the event is fully observed or censored

arm An indicator for the treatment arm, with 0 = control and 1 = active treatment

sex An indicator for the individual's sex, with 0 = male and 1 = female

age A numeric variable with the individual's age

imd A categorical variable representing a measure of area-level social deprivation

ethnic A categorical variable representing the individual's ethnic group, as measured from a Census

digitise   *Format digitised data for use in survival analysis*

Description

Produces txt files with Kaplan Meier and individual level survival data from digitised Kaplan Meier curves obtained by DigitizeIT

Usage

```
digitise(
  surv_inp,
  nrisk_inp,
  km_output = "KMdata.txt",
  ipd_output = "IPDdata.txt"
)
```

Arguments

surv_inp	a txt file obtained for example by DigitizeIT and containing the input survival times from graph reading
nrisk_inp	a txt file obtained by DigitizeIT and containing the reported number at risk
km_output	the name of the file to which the KM data will be written
ipd_output	the name of the file to which the individual level data data will be written

Author(s)

Patricia Guyot and Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

Examples

```
## Not run:
# Defines the txt files to be used as inputs
surv.inp <- system.file("extdata", "survival.txt", package = "survHE")
nrisk.inp <- system.file("extdata", "nrisk.txt", package = "survHE")
# Runs 'digitise' to create the relevant output files
digitise(surv.inp, nrisk.inp)

## End(Not run)
```

`fit.models`	*Fit parametric survival analysis for health economic evaluations*

Description

Runs the survival analysis with several useful options, using either MLE (via flexsurv) or a Bayesian approach (via R-INLA or rstan)

Usage

```
fit.models(formula = NULL, data, distr = NULL, method = "mle", ...)
```

Arguments

formula
: a formula specifying the model to be used, in the form `Surv(time,event)~treatment[+covariates]` for flexsurv, or `inla.surv(time,event)~treatment[+covariates]` for INLA

data
: A data frame containing the data to be used for the analysis. This must contain data for the 'event' variable. In case there is no censoring, then `event` is a column of 1s.

distr
: a (vector of) string(s) containing the name(s) of the model(s) to be fitted. Available options are:

: `flexsurv`: "exponential","gamma","genf","gengamma","gompertz","weibull", "weibullPH","loglogistic","lognormal" INLA: "exponential","weibull","lognormal","loglogistic" hmc: "exponential","gamma","genf","gengamma","gompertz","weibull","weibullPH", "loglogistic","lognormal"

method
: A string specifying the inferential method (`'mle'`, `'inla'` or `'hmc'`). If `method` is set to `'hmc'`, then `survHE` will write suitable model code in the Stan language (according to the specified distribution), prepare data and initial values and then run the model.

...
: Additional options (for INLA or HMC).

: **INLA** specific options `dz` = defines the step length for the grid search over the hyperparameters space (default = 0.1) `diff.logdens` = defines the difference in the log-density for the hyperparameters to stop integration (default = 5) `control.fixed` = defines the default for the priors, unless specified by the user. Default values are prior mean = 0 for *all* fixed effects prior var = 1000 for *all* fixed effects prior mean = 0 for the intercept prior prec -> 0 for the intercept `control.family` = a list of options. If distr is a vector, then can be provided as a named list of options, for example something like this: `control.family=list(weibull=list(param=c(.1,.1)),lognormal=list(initial=2))` the names of the elements of the list need to be the same as those given in the vector `distr`

: **HMC** specific options `chains` = number of chains to run in the HMC (default = 2) `iter` = total number of iterations (default = 2000) `warmup` = number of warmup iterations (default = iter/2) `thin` = number of thinning (default = 1) `control` = a list specifying Stan-related options, eg `control=list(adapt_delta=0.85)`

(default = NULL) seed = the random seed (to make things replicable) pars =
a vector of parameters (string, default = NA) include = a logical indicator (if
FALSE, then the pars are not saved; default = TRUE) priors = a list (of lists)
specifying the values for the parameters of the prior distributions in the models
save.stan = a logical indicator (default = FALSE). If TRUE, then saves the
data list for Stan and the model file(s)

Details

On object in the class survHE containing the following elements

Value

models   A list containing the fitted models. These contain the output from the original in-
                  ference engine (flexsurv, INLA or rstan). Can be processed using the methods
                  specific to the original packages, or via survHE-specific methods (such as plot,
                  print) or other specialised functions (eg to extrapolate the survival curves, etc).

model.fitting   A list containing the output of the model-fit statistics (AIC, BIC, DIC). The AIC
                  and BIC are estimated for all methods, while the DIC is only estimated when
                  using Bayesian inference.

method   A string indicating the method used to fit the model, ie 'mle', 'inla' or 'hmc'.

misc   A list containing the time needed to run the model(s) (in seconds), the formula
                  used, the results of the Kaplan-Meier analysis (which is automatically performed
                  using npsurv) and the original data frame.

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness
modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

make.surv

Examples

```
## Not run:
# Loads an example dataset from 'flexsurv'
data(bc)

# Fits the same model using the 3 inference methods
mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
inla = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="inla")
hmc = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
```

```
      distr="exp",method="hmc")

# Prints the results in comparable fashion using the survHE method
print(mle)
print(inla)
print(hmc)

# Or visualises the results using the original packages methods
print(mle,original=TRUE)
print(inla,original=TRUE)
print(hmc,original=TRUE)

# Plots the survival curves and estimates
plot(mle)
plot(mle,inla,hmc,labs=c("MLE","INLA","HMC"),colors=c("black","red","blue"))

## End(Not run)
```

---

make.ipd   *Create an individual level dataset from digitised data*

Description

Piles in the simulated IPD resulting from running digitise for more than one treatment arm

Usage

```
make.ipd(ipd_files, ctr = 1, var.labs = c("time", "event", "arm"))
```

Arguments

ipd_files	a list including the names of the IPD files created as output of digitise
ctr	the index of the file associated with the control arm (default, the first file). This will be coded as 0
var.labs	a vector of labels for the column of the resulting data matrix. NB these should match the arguments to the formula specified for fit.models. The user can specify values. These should be 4 elements (ID, TIME, EVENT, ARM)

Author(s)

Gianluca Baio

References

Something will go here

See Also

Something will go here

Examples

```
## Not run:
# Defines the txt files to be used as inputs
surv.inp <- system.file("extdata", "survival.txt", package = "survHE")
nrisk.inp <- system.file("extdata", "nrisk.txt", package = "survHE")
# Runs 'digitise' to create the relevant output files
digitise(surv.inp, nrisk.inp, ipd_output = "IPD.txt")
# Now uses 'make.ipd' to create the pseudo-data
make.ipd("IPD.txt", ctr = 1, var.labs = c("time", "event", "arm"))

## End(Not run)
```

---

make.surv   *Engine for Probabilistic Sensitivity Analysis on the survival curves*

Description

Creates the survival curves for the fitted model(s)

Usage

```
make.surv(fit, mod = 1, t = NULL, newdata = NULL, nsim = 1, ...)
```

Arguments

fit	the result of the call to the fit.models function, containing the model fitting (and other relevant information)
mod	the index of the model. Default value is 1, but the user can choose which model fit to visualise, if the call to fit.models has a vector argument for distr (so many models are fitted & stored in the same object)
t	the time vector to be used for the estimation of the survival curve
newdata	a list (of lists), specifiying the values of the covariates at which the computation is performed. For example list(list(arm=0),list(arm=1)) will create two survival curves, one obtained by setting the covariate arm to the value 0 and the other by setting it to the value 1. In line with flexsurv notation, the user needs to either specify the value for *all* the covariates or for none (in which case, newdata=NULL, which is the default). If some value is specified and at least one of the covariates is continuous, then a single survival curve will be computed in correspondence of the average values of all the covariates (including the factors, which in this case are expanded into indicators).
nsim	The number of simulations from the distribution of the survival curves. Default at nsim=1, in which case uses the point estimate for the relevant distributional parameters and computes the resulting survival curve
...	Additional options

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

`fit.models`, `psa.plot`, `write.surv`

Examples

```
## Not run:
# Loads an example dataset from 'flexsurv'
data(bc)

# Fits the same model using the 3 inference methods
mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
p.mle = make.surv(mle)
psa.plot(p.mle)
# Can also use the main 'plot' function to visualise the survival curves
# and include uncertainty by using a number 'nsim' of simulations
plot(mle,nsim=10)

## End(Not run)
```

---

make.transition.probs *make.transition.probs*

Description

Computes the transition probabilities (to be passed to a Markov model) from the survival curves obtained using `fit.models` and `make.surv`, using the formula $p(t)=1-S(t+k)/S(t)$, where k is the Markov model cycle length and t is a generic time

Usage

```
make.transition.probs(x, ...)
```

Arguments

x	an object obtained as output of the call to `make.surv`
...	additional arguments. Includes labs = a string vector of names for the elements of the list (strata for the survival analysis)

Details

Something will go here

Value

Something will go here

Note

Something will go here

Author(s)

Gianluca Baio

References

Something will go here

See Also

Something will go here

Examples

```
## Not run:
# Something will go here

## End(Not run)
```

---

model.fit.plot	*Graphical representation of the measures of model fitting based on Information Criteria*

Description

Plots a summary of the model fit for all the models fitted

Usage

```
model.fit.plot(..., type = "aic", scale = "absolute", stacked = FALSE)
```

Arguments

`...`	Optional inputs. Must include at least one `survHE` object.
`type`	should the AIC, the BIC or the DIC plotted? (values = `"aic"`, `"bic"` or `"dic"`)
`scale`	If `scale='absolute'` (default), then plot the absolute value of the *IC. If `scale='relative'` then plot a rescaled version taking the percentage increase in the *IC in comparison with the best-fitting model
`stacked`	Should the bars be stacked and grouped by survHE object? (default=F)

Details

Something will go here

Value

A plot with the relevant model fitting statistics

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

`fit.models`

Examples

```
## Not run:
data(bc)

mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr=c("exp","wei","lno"),method="mle")
model.fit.plot(mle)

## End(Not run)
```

| plot.survHE | *Plot survival curves for the models fitted using* fit.models |

Description

Plots the results of model fit.

Usage

```
## S3 method for class 'survHE'
plot(...)
```

Arguments

...   Must include at least one result object saved as the call to the fit.models function. Nay include other optional parameters. These include whether the KM curve should be added add.km and whether the user specifies a profile of covariates (in the list newdata). Other possibilities are additional (mainly graphical) options. These are: xlab = a string with the label for the x-axis (default = "time") ylab = a string with the label for the y-axis (default = "Survival") lab.profile = a (vector of) string(s) indicating the labels associated with the strata defining the different survival curves to plot. Default to the value used by the Kaplan Meier estimate given in fit.models. newdata = a list (of lists) providing the values for the relevant covariates If NULL, then will use the mean values for the covariates if at least one is a continuous variable, or the combination of the categorical covariates. xlim = a vector determining the limits for the x-axis colors = a vector of characters defining the colours in which to plot the different survival curves lab.profile = a vector of characters defining the names of the models fitted add.km = TRUE (whether to also add the Kaplan Meier estimates of the data) annotate = FALSE (whether to also add text to highlight the observed vs extrapolated data) legend.position = a vector of proportions to place the legend. Default to 'c(.75,.9)', which means 75 legend.title = suitable instructions to format the title of the legend; defaults to 'element_text(size=15,face="bold")' but there may be other arguments that can be added (using 'ggplot' facilities) legend.text = suitable instructions to format the text of the legend; defaults to 'element_text(colour="black", size=14, face="plain")' but there may be other arguments that can be added (using 'ggplot' facilities)

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

    `fit.models`, `write.surv`

Examples

```
## Not run:
data(bc)

mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
inla = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="inla")
plot(MLE=mle,INLA=inla)

## End(Not run)
```

---

poly.weibull   *Fit Poly-Weibull model for survival analysis of mixture hazards*

Description

    Runs the survival analysis using a Poly-Weibull model

Usage

```
poly.weibull(formula = NULL, data, ...)
```

Arguments

formula	a list of formulae (one for each components of the mixture. Can specify one single formula (in which case, the model is a simple Weibull regression). For example, a valid call is using `formula=list(Surv(time,event)~1,Surv(time,event)~arm)`
data	A data frame containing the data to be used for the analysis. This must contain data for the 'event' variable. In case there is no censoring, then `event` is a column of 1s.
...	Additional options (for INLA or HMC).
	HMC specific options `chains` = number of chains to run in the HMC (default = 2) `iter` = total number of iterations (default = 2000) `warmup` = number of warmup iterations (default = iter/2) `thin` = number of thinning (default = 1) `control` = a list specifying Stan-related options, eg `control=list(adapt_delta=0.85)` (default = NULL) `seed` = the random seed (to make things replicable) `pars` = a vector of parameters (string, default = NA) `include` = a logical indicator (if FALSE, then the pars are not saved; default = TRUE) `priors` = a list (of lists) specifying the values for the parameters of the prior distributions in the models `save.stan` = a logical indicator (default = FALSE). If TRUE, then saves the data list for Stan and the model file(s)

Details

On object in the class `survHE` containing the following elements

Value

`models`	A list containing the fitted models. These contain the output from the original inference engine (`flexsurv`, `INLA` or `rstan`). Can be processed using the methods specific to the original packages, or via `survHE`-specific methods (such as `plot`, `print`) or other specialised functions (eg to extrapolate the survival curves, etc).
`model.fitting`	A list containing the output of the model-fit statistics (AIC, BIC, DIC). The AIC and BIC are estimated for all methods, while the DIC is only estimated when using Bayesian inference.
`method`	A string indicating the method used to fit the model, ie `'mle'`, `'inla'` or `'hmc'`.
`misc`	A list containing the time needed to run the model(s) (in seconds), the formula used, the results of the Kaplan-Meier analysis (which is automatically performed using `npsurv`) and the original data frame.

Note

Something will go here

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

`fit.models`, `make.surv`

Examples

```
## Not run:


#See Baio (2019) for extended example
```

print.survHE   *Print a summary of the survival model(s) fitted by* fit.models

Description

Prints the summary table for the model(s) fitted, with the estimate of the parameters

Usage

```
## S3 method for class 'survHE'
print(x, mod = 1, ...)
```

Arguments

x	the survHE object (the output of the call to fit.models)
mod	is the index of the model. Default value is 1, but the user can choose which model fit to visualise, if the call to fit.models has a vector argument for distr (so many models are fitted & stored in the same object)
...	additional options, including: digits = number of significant digits to be shown in the summary table (default = 6) original = a flag to say whether the *original* table from either flexsurv or INLA or rstan should be printed

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

Examples

```
## Not run:
data(bc)

mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
print(mle)

## End(Not run)
```

---

psa.plot   *Graphical depiction of the probabilistic sensitivity analysis for the survival curves*

Description

Plots the survival curves for all the PSA simulations. The function is actually deprecated - similar graphs can be obtained directly using the `plot` method (with options), which allows a finer depiction of the results.

Usage

```
psa.plot(psa, ...)
```

Arguments

psa   the result of the call to the function `make.surv`

...   Optional graphical parameters, such as: `xlab` = label for the x-axis `ylab` = label for the y-axis `col` = (vector) of colors for the lines to be plotted `alpha` = the level of transparency for the curves (default = 0.2)

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

`make.surv`, `write.surv`

Examples

```
## Not run:
data(bc)

# Fits the same model using the 3 inference methods
mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
p.mle = make.surv(mle,nsim=100)
psa.plot(p.mle)

## End(Not run)
```

---

summary.survHE	*Prints a summary table for the distribution the mean survival time for a given model and data*

Description

Calculates the mean survival time as the area under the survival curve

Usage

```
## S3 method for class 'survHE'
summary(object, mod = 1, t = NULL, nsim = 1000, ...)
```

Arguments

object	a survHE object (resulting from the call to `fit.models`
mod	the model to be analysed (default = 1)
t	the vector of times to be used in the computation. Default = NULL, which means the observed times will be used. NB: the vector of times should be: i) long enough so that S(t) goes to 0; and ii) dense enough so that the approximation to the AUC is sufficiently precise
nsim	the number of simulations from the survival curve distributions to be used (to compute interval estimates)
...	Additional options

Details

A list comprising of the following elements

Value

mean.surv	A matrix with the simulated values for the mean survival times
tab	A summary table

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

`fit.models`, `make.surv`

Examples

```
## Not run:
data(bc)

mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
summary(mle,nsim=100)

## End(Not run)
```

---

write.surv   *write.surv*

Description

Writes the survival summary to an excel file (helpful to then call the values in the Markov model)

Usage

```
write.surv(object, file, sheet = NULL, what = "surv")
```

Arguments

object	a summary.flexsurvreg object containing the survival curves (with times, estimates and interval limits)
file	a string with the full path to the file name to be saved
sheet	a string with the name of the sheet to be created
what	a string to describe what to be exported. Can either be 'surv' (default), which outputs the simulation(s) for the survival curves or 'sim', which outputs the simulation(s) for the underlying model parameters. If there are several 'profiles', they get written in separate spreadsheets and a clear indication is given as the name of the spreadsheet

Details

Something will go here

Value

A spreadsheet file with the simulation(s) of the relevant quantity

Author(s)

Gianluca Baio

References

G Baio (2019). survHE: Survival analysis for health economic evaluation and cost-effectiveness modelling. Journal of Statistical Software (2020). vol 95, 14, 1-47. <doi:10.18637/jss.v095.i14>

See Also

```
make.surv
```

Examples

```
## Not run:
# Loads an example dataset from 'flexsurv'
data(bc)

# Fits the same model using the 3 inference methods
mle = fit.models(formula=Surv(recyrs,censrec)~group,data=bc,
    distr="exp",method="mle")
p.mle = make.surv(mle)
write.surv(p.mle,file="test.xlsx")

## End(Not run)
```

Index