Package ‘psc’

April 10, 2025

Type Package

Title Personalised Synthetic Controls

Version 1.2.0

Maintainer Richard Jackson <`richJ23@liverpool.ac.uk`>

Description Allows the comparison of data cohorts (DC) against a Counter Factual Model (CFM) and measures the difference in terms of an efficacy parameter. Allows the application of Personalised Synthetic Controls.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

URL `https://github.com/richjjackson/psc/`

BugReports `https://github.com/richJJackson/psc/issues`

Depends R (>= 3.5.0)

Imports mvtnorm, survival, enrichwith, stats, flexsurv, ggplot2, survminer

Suggests knitr, rmarkdown, devtools, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Richard Jackson [cre, aut, cph] (<`https://orcid.org/0000-0002-7814-5088`>)

Repository CRAN

Date/Publication 2025-04-10 13:40:02 UTC

Contents

`acc` *acc*

Description

Function to accept (or not) a proposed solution used as part of the MCMC procedure

Usage

```
acc(old, new)
```

Arguments

`old`	a numeric value
`new`	a numeric value

Details

A function for the evaluation of two likelihoods as part of the MCMC procedure

Value

returns the an evaluation of old/new > U where U is a draw from the uniform distribution

---

`bin.mod` *Example model for a survival outcome*

Description

A generated model with a binary endpoint and a logistic link function. Data for the model were synthetically generated and are based on a dataset to evaulate the use of Sorafenib in HCC akin to the PROSASH model (see ?psc::surv.mod for more details)

Usage

```
bin.mod
```

Format

A model of class 'glm':

vi vascular invasion

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastesis

Source

simulated

`coef.psc` *Returns the coefficeint estimate of a psc object.*

Description

Returns basic measures of the posterior distribution obtained from the pscfit.R function

Usage

```
## S3 method for class 'psc'
coef(object, ..., level = 0.05)
```

Arguments

`object`	a 'psc' object
`...`	not used
`level`	the level at which credibility intervals are assessed, defaults to 0.05

Value

The summary of the posterior distribution for the efficacy parameter in terms of the median and 95

`cont.mod` *Example model for a survival outcome*

Description

A generated model with a continuous data endpoint and a identity link function. Data for the model were synthetically generated and are based on a dataset to evaulate the use of Sorafenib in HCC akin to the PROSASH model (see ?psc::surv.mod for more details)

Usage

```
cont.mod
```

Format

A model of class 'glm':

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

Source

simulated

`count.mod` *Example model for a survival outcome*

Description

A generated model with a count data endpoint and a log link function. Data for the model were synthetically generated and are based on a dataset to evaulate the use of Sorafenib in HCC akin to the PROSASH model (see ?psc::surv.mod for more details)

Usage

```
count.mod
```

Format

A model of class 'glm':

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

Source

simulated

`data` *Example Dataset of patients with aHCC receiving Lenvetanib*

Description

A dataset containing 100 simulated patients. Data are based on the data used to generate PROSASH survival model -see ?psc::surv.mod for more detials.

Usage

```
data
```

Format

A model of class ’flezsurvreg’:

gamma cumulative baseline hazard parameters

vi vascular invasion

age60 patient age (centred at 60)

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastesis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count exapmple outcome for count data

trt exapmple identifier for mulitple treatment comparisons

aet Aetiology

Source

simulated

`dataComb` *A generic function for cleaning data ready for analysis*

Description

A generic function for cleaning data ready for analysis

Usage

```
dataComb(CFM, DC, id = NULL, trt = NULL)
```

Arguments

`CFM`	a model object supplied to pscfit
`DC`	a dataset including covariates to match the CFM
`id`	to specify which observations in the data cohort should be evaluated. Defualts to ’NULL’ i.e all observations
`trt`	used to specify multiple treatment effects. Defaults to NULL

Value

datComb returns a list containing objects which detial the components of both the Counter Factual Model (CFM) and the Data Cohort (DC) the required exported components of the model and a cleaned data cohort.

Examples

```
bin.mod <- psc::bin.mod
data <- psc::data
dc <- dataComb(bin.mod,data)
```

`dataComb.flexsurvreg` *Function for cleaning the data of a model with class 'flexsurvreg'*

Description

The purpose of this function is to prepare the dataset and the counter-factual model for estimation and is the first step pf the pscfit.R process. The output is a complete-case dataset where the data names match the variables used in the CFM.

Usage

```
## S3 method for class 'flexsurvreg'
dataComb(CFM, DC, id = NULL, trt = NULL)
```

Arguments

`CFM`	a model object supplied to pscfit
`DC`	a dataset including covariates to match the CFM
`id`	a vector specifiying whether a subset of the dataset should be selected. Defaults to 'NULL' e.g. all data points included
`trt`	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Value

a list containing objects which specifiy the required exported components of the model and a cleaned data cohort. Attributes include

- 'model.type' specifying the class of model to be used as the CFM
- 'model_extract' sepcifying the model componets required for estimation
- 'cov' a cleaned dataset of covariates
- 'outcome' a cleaned dataset containing the outcomes

`dataComb.glm` *Function for cleaning the data of a model with class 'flexsurvreg'*

Description

The purpose of this function is to prepare the dataset and the counter-factual model for estimation and is the first step pf the pscfit.R process. The output is a complete-case dataset where the data names match the variables used in the CFM.

Usage

```
## S3 method for class 'glm'
dataComb(CFM, DC, id = NULL, trt = NULL)
```

Arguments

`CFM`	a model object supplied to pscfit
`DC`	a dataset including covariates to match the CFM
`id`	to specify which observations in the data cohort should be evaluated. Defualts to 'NULL' i.e all observations
`trt`	used to specify multiple treatment effects. Defaults to NULL

Value

a list containing objects which specifiy the required exported components of the model and a cleaned data cohort. Attirbutes of the 'cleaned' object include: Attributes include

- 'model.type' specifying the class of model to be used as the CFM
- 'model_extract' sepcifying the model componets required for estimation
- 'cov' a cleaned dataset of covariates
- 'outcome' a cleaned dataset containing the outcomes

`dataSumm` *A function which provides basic summary information of a matrix x*

Description

A function which provides basic summaries of data provided within a data frame of covariets. Summaries are in the form of frequencies of counts and associated percentages for categorical data and median (IQR) for continuous data. Intended for use with the pscCFM.R function to define the setting in which a model has been generated

Usage

```
dataSumm(x)
```

Arguments

`x`	a matrix of covariates

Details

Categorical/Character data are summarised by a table and Continuous data are summarised as median (IQR)

`data_match` *A function to ensure that data from the cfm and data cohort are compatible*

Description

The purpose of this function is to run a series of checks to ensure that the data included in the data cohort is comparable to the counter-factual model

Usage

```
data_match(cfm.data, dc.data)
```

Arguments

`cfm.data`	a model object supplied to pscfit
`dc.data`	a dataset including covariates to match the CFM

Value

a dataset which is checked and compatible with the CFM

`initParm` *Fucntion for estimating initial parameter values 'flexsurvreg'*

Description

A generic function for estimating the initial parameters for estimation as part of the pscfti.R function. Parameter estimates are obtained using standard optomisation methods provided by the 'optim' set of functions. For a single parameter a Brent method is applied. For mutliple treatment comparisons the 'BFGS' method is applied.

Usage

```
initParm(CFM, DC_clean, trt)
```

Arguments

`CFM`	A counter-factual model
`DC_clean`	a cleaned dataset obsect obtained using dataComb.flexsurvreg
`trt`	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the liklihood for a 'flexsurvreg' model and uses 'optim' to fit the likelihood.

Value

an 'optim' output giving the parameter values to be supplied as a starting value for the mcmc routine.

`initParm.flexsurvreg` *Fucntion for estimating initial parameter values 'flexsurvreg'*

Description

Fucntion for estimating initial parameter values 'flexsurvreg'

Usage

```
## S3 method for class 'flexsurvreg'
initParm(CFM, DC_clean, trt = NULL)
```

Arguments

`CFM`	A counter-factual model
`DC_clean`	a cleaned dataset obsect obtained using dataComb.flexsurvreg
`trt`	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the liklihood for a 'flexsurvreg' model and uses 'optim' to fit the likelihood.

Value

an 'optim' output giving the parameter values to be supplied as a starting value for the mcmc routine.

`initParm.glm` *Fucntion for estimating initial parameter values 'flexsurvreg'*

Description

Fucntion for estimating initial parameter values 'flexsurvreg'

Usage

```
## S3 method for class 'glm'
initParm(CFM, DC_clean, trt = trt)
```

Arguments

`CFM`	A counter-factual model
`DC_clean`	a cleaned dataset obsect obtained using dataComb.flexsurvreg
`trt`	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the liklihood for a 'flexsurvreg' model and uses 'optim' to fit the likelihood.

Value

an 'optim' output giving the parameter values to be supplied as a starting value for the mcmc routine.

`lik.flexsurvreg` *Likelihood function for a psc model of class 'flexsurvreg'*

Description

A function which defines the likelihood for a PSC model where the Counter Factual Model (CFM) takes the form of a 'flexsurvreg' object and an efficacy parameter (β) is being estimated. For more details on fitting please see ?pscfit and ?pscEst

Usage

```
lik.flexsurvreg(beta, DC_clean)
```

Arguments

`beta`	a parameter to be estimate
`DC_clean`	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg'

`lik.flexsurvreg.mtc` *Likelihood function for a psc model of class 'flexsurv' with multiple treatment comparisons*

Description

A function which defines the likelihood for a PSC model where the Counter Factual Model (CFM) takes the form of a 'flexsurvreg' object and a mulitple efficacy parameters (β) is being estimated. For more details on fitting please see ?pscfit and ?pscEst

Usage

```
lik.flexsurvreg.mtc(beta, DC_clean)
```

Arguments

`beta`	a parameter to be estimate
`DC_clean`	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg' where multiple treatment comparisons are required

---

`lik.glm` *Likelihood function for a psc model of class 'glm'*

Description

A function which defines the likelihood for a PSC model where the Counter Factual Model (CFM) takes the form of a 'glm' object and an efficacy parameter (β) is being estimated. For more details on fitting please see ?pscfit and ?pscEst

Usage

```
lik.glm(beta, DC_clean)
```

Arguments

`beta`	a parameter to be estimate
`DC_clean`	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'glm'

`lik.glm.mtc` *Likelihood function for a psc model of class 'glm' with multiple treatment comparisons*

Description

A function which defines the likelihood for a PSC model where the Counter Factual Model (CFM) takes the form of a 'glm' object and a mulitple efficacy parameters (β) is being estimated. For more details on fitting please see ?pscfit and ?pscEst

Usage

```
lik.glm.mtc(beta, DC_clean)
```

Arguments

`beta`	a parameter to be estimate
`DC_clean`	a cleaned dataset including covariates to match the CFM

Details

A likelihood function for use by pscfit for a model of class 'flexsurvreg' where multiple treatment comparisons are required

---

`linPred` *Estimates the linear predictor of a psc object*

Description

A function which created a linear predictor from a model and an external dataset. If required, linear predictors will be provided on the response scale. For a survival outcome, this will take the form of median survival estimates.

Usage

```
linPred(DC_clean, resp = FALSE)
```

Arguments

`DC_clean`	a cleaned data obhject created using dataComb()
`resp`	detailing whether the linear predictor shoudl be returned on the natural or response level. Defaults to the natural scale (resp=F)

Details

A function which combines the data from the data cohort against the model parameters of the PSC

Value

Extracts the linear predictor from a object containing both a counter factual model and a data cohort which is created using the dataComb() fucntion.

Examples

```
library(psc)
library(survival)
surv.mod <- psc::surv.mod
data <- psc::data
dc <- dataComb(surv.mod,data)
lp <- linPred(dc)
```

`modelExtract` *A generic function for extracting model information*

Description

A generic function for extracting model information

Usage

```
modelExtract(CFM)
```

Arguments

`CFM`	a model of class either 'glm' or 'flexsurvreg'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

`modelExtract.flexsurvreg`

A generic function for extracting model information

Description

This function extracts model information for use with the pscfit.R function.

Usage

```
## S3 method for class 'flexsurvreg'
modelExtract(CFM)
```

Arguments

`CFM`	a model of class ’flexsurvreg’

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components. Included are

- model.frame a dataset of the covareits used to create the model
- cov_co: covariate coefficients
- sig: variance-covariance matrix
- haz_co: hazard parameter coefficients
- k: number of knots
- knots: knot position
- lam: lambda parameter
- form: model formula

`modelExtract.glm` *A generic function for extracting model information*

Description

A generic function for extracting model information

Usage

```
## S3 method for class 'glm'
modelExtract(CFM)
```

Arguments

`CFM`	a model of class 'glm'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components. Included are

- model.frame a dataset of the covareits used to create the model
- cov_co: covariate coefficients
- sig: variance-covariance matrix
- form: model formula
- family: model family
- out.nm: outcome covariates names

`modp` *modp*

Description

A function which rrturns either the input value (if positive) or zero (if negative)

Usage

```
modp(x)
```

Arguments

`x`	a numberic vector

Details

A fucntion which returns a version of x with negative values replacd with 0

Value

a numeric vector with negative values replaced with 0

`plot.psc` *Function for Plotting PSC objects*

Description

A function which illsutrates the predicted response under the Counter Factual Model (CFM) and the observed response under the experimental treatment(s). Form of the output will depend on the form of the CFM used

Usage

```
## S3 method for class 'psc'
plot(x, ...)
```

Arguments

`x`	an object of class 'psc'
`...`	not used

Details

This function plots the expected response of the control treatment along with the observe response rates of the experimental arms

Value

a survival plot corresponding to the psc fit

Examples

```
bin.mod <- psc::bin.mod
data <- psc::data
bin.psc <- pscfit(bin.mod,data,nsim=3000)
plot(bin.psc)
```

`plot.psc.binary` *Function for Plotting PSC objects*

Description

A function which illsutrates the predicted response under the counter factual model and the observed response under the experimental treatment(s).

Usage

```
## S3 method for class 'psc.binary'
plot(x, ...)
```

Arguments

`x`	an object of class 'psc'
`...`	not used

Details

This function plots the expected response of the control treatment along with the observe response rates of the experimental arms

Value

a survival plot corresponding to the psc fit

Examples

```
bin.mod <- psc::bin.mod
data <- psc::data
bin.psc <- pscfit(bin.mod,data,nsim=3000)
plot(bin.psc)
```

`plot.psc.cont` *Function for Plotting PSC objects*

Description

A function which illsutrates the predicted response under the counter factual model and the observed response under the experimental treatment(s).

Usage

```
## S3 method for class 'psc.cont'
plot(x, ...)
```

Arguments

`x`	an object of class 'psc'
`...`	not used

Details

This function plots the expected response of the control treatment along with the observe response rates of the experimental arms

Value

a survival plot corresponding to the psc fit

Examples

```
cont.mod <- psc::cont.mod
data <- psc::data
cont.psc <- pscfit(cont.mod,data,nsim=3000)
plot(cont.psc)
```

`plot.psc.count` *Function for Plotting PSC objects #' A function which illsutrates the predicted response under the counter factual model and the observed response under the experimental treatment(s).*

Description

Function for Plotting PSC objects #' A function which illsutrates the predicted response under the counter factual model and the observed response under the experimental treatment(s).

Usage

```
## S3 method for class 'psc.count'
plot(x, ...)
```

Arguments

`x`	an object of class 'psc'
`...`	not used

Details

This function plots the expected response of the control treatment along with the observe response rates of the experimental arms

Value

a survival plot corresponding to the psc fit

Examples

```
count.mod <- psc::count.mod
data <- psc::data
count.psc <- pscfit(count.mod,data,nsim=3000)
plot(count.psc)
```

`plot.psc.flexsurvreg` *Function for Plotting PSC objects*

Description

Function for Plotting PSC objects

Usage

```
## S3 method for class 'psc.flexsurvreg'
plot(x, ...)
```

Arguments

`x`	an object of class 'psc'
`...`	not used

Details

making use of 'ggsurvplot' in the survminer package, this function plots the expected survival funtion for the 'control' treatment estimated from the CFM along with the Kaplan Meier estimates of the observed events

Value

a survival plot corresponding to the psc fit

`print.psc` *Personalised Synthetic Controls - print*

Description

Personalised Synthetic Controls - print

Usage

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

Arguments

`x`	an object of class 'psc'
`...`	not used

Value

printing psc results

`psc.object` *Fitted* `psc` *object*

Description

An object returned by the `pscfit` function, inheriting from class `psc` and representing a fitted personlised synthetic control model.

Usage

```
psc.object
```

Format

An object of class `NULL` of length 0.

Author(s)

Richard Jasckson (<`richj23@liverpool.ac.uk`>)

`pscEst` *Function for performing estimation procedures in 'pscfit'*

Description

Function for performing estimation procedures in 'pscfit'

Usage

```
pscEst(CFM, DC_clean, nsim, start, start.se, trt)
```

Arguments

`CFM`	a model object supplied to pscfit
`DC_clean`	a cleaned dataset ontained using dataComb().
`nsim`	the number of MCMC simulations to run
`start`	the stating value for
`start.se`	the stating value for
`trt`	an optional vector denoting treatment allocations where mulitple treatment comparisons are bieng made

Details

Define the set of model parameters B to contain Γ which summarize the parameters of the CFM. Prior distributions are defined for B using a multivariate normal distribution $\pi(B) \sim MVN(\mu, \Sigma)$ where $\mu|$ is the vector of coefficient estimates from the validated model and Σ is the variance-covariance matrix. This information is taken directly from the outputs of the parametric model and no further elicitation is required. The prior distirbution for the efficacy parameter ($\pi(\beta)$) is set as an uniformative $N(0, 1000)$.

Ultimately the aim is to estimate the posterior distribution for β conditional on the distribution of B and the observed data. A full form for the posterior distribution is then given as

$$P(\beta|B, D) \propto L(D|B, \beta)\pi(B)\pi(\beta)$$

Please see 'pscfit' for more details on liklihood formation.

For each iteration of the MCMC procedure, the following algorithm is performed

1. Set and indicator s=1, and define an initial state based on prior hyperparameters for $\pi(B)$ and $\pi(\beta)$ such that $b_s = \mu and \tau_s = 0$
2. Update $s = s + 1$ and draw model parameters b_s from $\pi(B)$ and an draw a proposal estimate of β from some target distribution
3. Estimate $\Gamma_(i, S) = \nu^T x_i$ where ν is the subset of parameters from b_s which relate to the model covariates and define 2 new likelihood functions $\Theta_(s, 1) = L(D|B = b_s, \beta = \tau_(s-1))$ & $\Theta_(s, 2) = L(D|B = b_s, \beta = \tau_s)$
4. Draw a single value ψ from a Uniform (0,1) distribution and estimate the condition $\omega = \Theta_(s, 1)/\Theta_(s, 2)$. If $\omega > \psi$ then accept τ_s as belonging to the posterior distribution $P(\beta|B, D)$ otherwise retain $\tau_(s-1)$
5. Repeat steps 2 – 4 for the required number of iterations

The result of the algorithm is a posterior distribution for the log hazard ratio, β, captures the variability in B through the defined priors $\pi(\beta)$.

Value

A matrix containing the draws form the posterior distribution

---

`pscEst.flexsurvreg` *Function for estimating initial parameter values 'flexsurvreg'*

Description

A function which performs the Bayesian MCMC estimation procedure for estimating the efficacy parameter (β) using personalised sunthetic controls methodology.

Usage

```
## S3 method for class 'flexsurvreg'
pscEst(CFM, DC_clean, nsim, start, start.se, trt = trt)
```

Arguments

`CFM`	a model object supplied to pscfit
`DC_clean`	a cleaned dataset ontained using dataComb().
`nsim`	the number of MCMC simulations to run
`start`	the stating value for
`start.se`	the stating value fo
`trt`	an optional vector denoting treatment allocations where multiple treatment comparisons are being made

Details

An MCMC routine for fitting a psc model

Value

A matrix containing the draws form the posterior distribution

---

`pscEst.glm` *Function for estimating initial parameter values 'flexsurvreg'*

Description

A function which performs the Bayesian MCMC estimation procedure for estimating the efficacy parameter (β) using personalised sunthetic controls methodology.

Usage

```
## S3 method for class 'glm'
pscEst(CFM, DC_clean, nsim, start, start.se, trt = trt)
```

Arguments

`CFM`	a model object supplied to pscfit
`DC_clean`	a cleaned dataset ontained using dataComb().
`nsim`	the number of MCMC simulations to run
`start`	the stating value for
`start.se`	the stating value fo
`trt`	an optional vector denoting treatment allocations where multiple treatment comparisons are being made

Details

An MCMC routine for fitting a psc model

Value

a matrix containing the draws form the posterior distribution

---

`pscfit` *Personalised Synthetic Controls model fit*

Description

Function which allows comparison of a data cohort against a parametric Counter Factual Model (CFM). The function allows models of the type 'flexsurvreg' and 'glm' to be supplied. The function performs by calculating the linear predictor as a combination of the CFM and the dataset supplied and then selects a likelihood based on the type of model specified. Likelihood is estimated using a Baysian MCMC procedure wherebey the parameters of the CFM acts as informative priors.

Usage

```
pscfit(CFM, DC, nsim = 5000, id = NULL, trt = NULL)
```

Arguments

`CFM`	An R model object of class 'glm' or 'flexsurvspline'
`DC`	A dataset including columns to match to covariates in the model
`nsim`	The number of simulations for the MCMC routine
`id`	Numeric vector stating which patient(s) from the dataset should be included in the analysis. Defaults to all patients
`trt`	An optional vector denoting treatment allocations for multiple treatment comparisons. Defaults to NULL.

Details

Model currently supports estimation of more than one treatment (using the 'trt') option and esitmation restricted to sub-groups of the data cohort (using the 'id' option.

the `pscfit` function compares a dataset ('DC') against a parametric model. This is done by selecting a likelihood which is identified by the type of CFM that is supplied. At present, two types of model are supported, a flexible parmaeteric survival model of type 'flexsurvreg' and a geleneralised linear model of type 'glm'.

Where the CFM is of type 'flexsurvreg' the likeihood supplied is of the form:

$$L(D|\Lambda,\Gamma_i) = \prod_{i=1}^{n} f(t_i|\Lambda,\Gamma_i)^{c_i} S(t_i|\Lambda,\Gamma_i)^{(1-c_i)}$$

Where Λ defines the cumulative baseline hazard function, Γ is the linear predictor and t and c are the event time and indicator variables.

Where the CFM is of the type 'glm' the likelihood supplied is of the form:

$$L(x|\Gamma_i) = \prod_{i=1}^{n} b(x|\Gamma_i) \exp\{\Gamma_i^T t(x) - c(\Gamma_i)\}$$

Where $b(.)$, $t(.)$ and $c(.)$ represent the functions of the exponential family. In both cases, Γ is defined as:

$$\Gamma = \gamma x + \beta$$

Where γ are the model coefficients supplied by the CFM and β is the parameter set to measure the difference between the CFM and the DC.

Estimation is performed using a Bayesian MCMC procedure. Prior distributions for Γ (& Λ) are derived directly from the model coefficients (mean and variance covariance matrix) or the CFM. A bespoke MCMC routine is performed to estimate β. Please see '?mcmc' for more detials.

For the standard example where the DC contains information from only a single treatment, trt need not be specified. Where comparisons between the CFM and multiple treatments are require, a covariate of treamtne allocations must be specified sperately (using the 'trt' option).

Value

a object of class 'psc' with attributes model.type, the cleaned Dataset and the posterior distribution of the fitted model

Attributes include

- A 'cleaned' dataset including extracted components of the CFM and the cleaned DC included in the procedure
- An object defining the class of model (and therefore the procedure applied - see above)
- A matrix containing the draws of the posterior distributions

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
surv.psc <- pscfit(surv.mod,data)
```

`pscSumm` *Personalised Synthetic Controls - summary*

Description

A function which provides a summary of a DC_clean object. To be used either in conjunction with dataComb.R or summary.psc.R

Usage

```
pscSumm(DC_clean)
```

Arguments

`DC_clean`	a cleaned dataset ontained using dataComb().

Value

psc summary results including an estimate of the linear predictor combing the data and the model, an estimate of patient level response and summary statistics of the average responses for the sythenthic and observed populations

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
dc <- dataComb(surv.mod,data)
pscSumm(dc)
```

`summary.psc` *Personalised Synthetic Controls - summary*

Description

A generic function to provide a summary of a 'psc' object obtained from pscfit.R

Usage

```
## S3 method for class 'psc'
summary(object, ...)
```

Arguments

`object`	an object of class 'psc'
`...`	not used

Value

A summary of a psc object obtained using pscSumm and a copy of the pscfit object

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
psc.ob <- pscfit(surv.mod,data)
summary(psc.ob)
```

`surv.mod` *Example model for a survival outcome*

Description

A generated model with a survival endpoint and a cuymulative hazard function estimated using flexible parametric splines. Data for the model were synthetically generated and are based on a dataset to evaulate the use of Sorafenib in HCC akin to the PROSASH model

Usage

```
surv.mod
```

Format

A model of class 'flezsurvreg':

gamma cumulative baseline hazard parameters

vi vascular invasion

age60 patient age (centred at 60)

ecog ECOG performance Status

logafp AFP - log scale

alb albumin

logcreat Creatinine - log scale

allmets metastesis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count exapmple outcome for count data

trt exapmple identifier for mulitple treatment comparisons

aet Aetiology

Source

simulated

References

Using prognostic and predictive clinical features to make personalised survival prediction in advanced hepatocellular carcinoma patients undergoing sorafenib treatment. Berhane S, et al., Br J Cancer. 2019 Jul;121(2):117-124

`surv_fpm` *A function to obtain survival estimates from a flexible parametric model*

Description

This function provides basic survival estimates from a flexible parametric survival model

Usage

```
surv_fpm(DC_clean, beta = 0, s = NULL)
```

Arguments

`DC_clean`	a cleaned dataset ontained using dataComb().
`beta`	a parameter to determine if the survival probabilities should be adjusted by some (log) hazard ratio. Defaults to beta=0, i.e. no adjustment.
`s`	if specified will return the time at which some threshold is passed (e.g. s=0.5 for median survival time)

Details

A fucntion which extracts survival probabilities from a flexsurvreg object

Value

a list of times and assoicated survival probabilities

Examples

```
library(psc)
library(survival)
data("surv.mod")
data("data")
dc <- dataComb(surv.mod,data)
s_est <- surv_fpm(dc)
```

Index