

Package: psc (via r-universe)

February 27, 2025

Type Package

Title Personalised Synthetic Controls

Version 1.0.0

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 (>= 3.3.0)

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

Suggests knitr, rmarkdown, devtools

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository <https://richjjackson.r-universe.dev>

RemoteUrl <https://github.com/richjjackson/psc>

RemoteRef HEAD

RemoteSha ac1e72039967dff5cd694b10ca0ed12b272b3cc5

Contents

acc	3
bin.mod	3
cfmSett	4
cfmValid	4
cfmValid.flexsurvreg	5
coef.psc	6
cont.mod	6
count.mod	7
data	8
dataComb	9
dataComb.flexsurvreg	9
dataComb.glm	10
dataSumm	11
initParm	11
initParm.flexsurvreg	12
initParm.glm	13
lik.flexsurvreg	13
lik.flexsurvreg.mtc	14
lik.glm	14
lik.glm.mtc	15
linPred	15
modelExtract	16
modelExtract.flexsurvreg	17
modelExtract.glm	18
modp	18
plot.psc	19
plot.psc.binary	20
plot.psc.cont	20
plot.psc.count	21
plot.psc.flexsurvreg	22
plot_ite	22
plot_ite.flexsurvreg	23
plot_ite.glm	24
print.psc	25
psc.object	25
pscCFM	26
pscEst	26
pscEst.flexsurvreg	27
pscEst.glm	28
pscfit	29
pscSumm	30
summary.psc	31
surv.mod	32
surv_fpm	33

Index

34

acc	<i>acc</i>
-----	------------

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 $\text{old/new} > U$ where U is a draw from the uniform distribution

bin.mod	<i>Example model for a survival outcome</i>
---------	---

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 evaluate 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 metastasis

Source

simulated

cfmSett	<i>A function to store setting information for a cfm model</i>
---------	--

Description

This function collects information on the setting of a model for the definition of a Counter Factual Model (CFM) object. Setting information is provided in the PICO format. Intended to be used in conjunction with the pscCRM.R function

Usage

```
cfmSett(P = NULL, I = NULL, C = NULL, O = NULL)
```

Arguments

P	Character; patient group
I	Character; intervention group - perhaps left unspecified
C	Character; comparator group
O	Character; outcome definition

Details

A function for containing basic information pertaining to the setting of a model in PICO format

cfmValid	<i>Validation of a flexsurvreg model This function intends to perform basic validation of a survival model of class 'flexsurvreg'. As a default internal validation will be performed by estimating basic measures of discrimination and calibration. If provided, measures will be estimated on external data. This is intended to be used in conjunction with the pscCRM.R function#' A generic function for extracting model information</i>
----------	---

Description

Validation of a flexsurvreg model This function intends to perform basic validation of a survival model of class 'flexsurvreg'. As a default internal validation will be performed by estimating basic measures of discrimination and calibration. If provided, measures will be estimated on external data. This is intended to be used in conjunction with the pscCRM.R function#' A generic function for extracting model information

Usage

```
cfmValid(cfm, exData)
```

Arguments

cfm	a model of class 'flexsurvreg'
exData	an external dataset (for external validation)

Value

a list of validation procedures for measures of discrimination and calibration. Discrimination performed by creating a variable with risk categories based on the 15th, 50th and 85th percentile of the linear predictor. Discrimination estimated provided in terms of kaplan merie estimates and hazard ratios.

Calibration performed by extracting the concordance statistic and by regression the linear predictor against the outcome.

Validation is internal unless an external dataset is provided.

cfmValid.flexsurvreg	<i>Validation of a flexsurvreg model This function intends to perform basic validation of a survival model of class 'flexsurvreg'. As a default internal validation will be performed by estimating basic measures of discrimination and calibration. If provided, measures will be estimated on external data. This is intended to be used in conjunction with the pscCRM.R function#' A generic function for extracting model information</i>
----------------------	---

Description

Validation of a flexsurvreg model This function intends to perform basic validation of a survival model of class 'flexsurvreg'. As a default internal validation will be performed by estimating basic measures of discrimination and calibration. If provided, measures will be estimated on external data. This is intended to be used in conjunction with the pscCRM.R function#' A generic function for extracting model information

Usage

```
## S3 method for class 'flexsurvreg'
cfmValid(cfm, exData = NULL)
```

Arguments

cfm	a model of class 'flexsurvreg'
exData	an external dataset (for external validation)

Value

a list of validation procedures for measures of discrimination and calibration. Discrimination performed by creating a variable with risk categories based on the 15th, 50th and 85th percentile of the linear predictor. Discrimination estimated provided in terms of kaplan merie estimates and hazard ratios.

Calibration performed by extracting the concordance statistic and by regression the linear predictor against the outcome.

Validation is internal unless an external dataset is provided.

coef.psc	<i>Returns the coefficient estimate of a psc object.</i>
----------	--

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	<i>Example model for a survival outcome</i>
----------	---

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 evaluate 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 evaluate 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 details.

Usage

data

Format

A model of class 'flezs survreg':

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 metastasis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count example outcome for count data

trt example identifier for multiple treatment comparisons

aet Aetiology

Source

simulated

dataComb	<i>A generic function for cleaning data ready for analysis</i>
----------	--

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. Defaults to 'NULL' i.e all observations
trt	used to specify multiple treatment effects. Defaults to NULL

Value

datComb returns a list containing objects which detail 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

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

dataComb.flexsurvreg	<i>Function for cleaning the data of a model with class 'flexsurvreg'</i>
----------------------	---

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 specifying 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 specify 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' specifying the model components 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 of 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. Defaults to 'NULL' i.e all observations
trt	used to specify multiple treatment effects. Defaults to NULL

Value

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

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

dataSumm	<i>A function which provides basic summary information of a matrix x</i>
----------	--

Description

A function which provides basic summaries of data provided within a data frame of covariates. 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)

initParm	<i>Fucntion for estimating initial parameter values 'flexsurvreg'</i>
----------	---

Description

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

Usage

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

Arguments

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

Details

This function takes the likelihood 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 *Function for estimating initial parameter values 'flexsurvreg'*

Description

Function 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 object obtained using dataComb.flexsurvreg
trt	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the likelihood 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	<i>Fucntion for estimating initial parameter values 'flexsurvreg'</i>
--------------	---

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 object obtained using dataComb.flexsurvreg
trt	An optional additional vector denoting treatment allocations for multiple treatment comparisons. Defaults to 'NULL'

Details

This function takes the likelihood 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	<i>Likelihood function for a psc model of class 'flexsurvreg'</i>
-----------------	---

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 ?pscfitt 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	<i>Likelihood function for a psc model of class 'flexsurv' with multiple treatment comparisons</i>
---------------------	--

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 multiple 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	<i>Likelihood function for a psc model of class 'glm'</i>
---------	---

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	<i>Likelihood function for a psc model of class 'glm' with multiple treatment comparisons</i>
-------------	---

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 multiple efficacy parameters (β) is being estimated. For more details on fitting please see ?pscfitt 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	<i>Estimates the linear predictor of a psc object</i>
---------	---

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 object created using dataComb()
resp	detailing whether the linear predictor should 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() function.

Examples

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

modelExtract	<i>A generic function for extracting model information</i>
--------------	--

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	<i>A generic function for extracting model information</i>
------------------	--

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	<i>modp</i>
------	-------------

Description

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

Usage

```
modp(x)
```

Arguments

x a numeric vector

Details

A function which returns a version of x with negative values replaced with 0

Value

a numeric vector with negative values replaced with 0

plot.psc

Function for Plotting PSC objects

Description

A function which illustrates 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 observed 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)  
plot(bin.psc)
```

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

Description

A function which illustrates the predicted response under the counterfactual 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 observed 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)  
plot(bin.psc)
```

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

Description

A function which illustrates the predicted response under the counterfactual 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)
plot(cont.psc)
```

plot.psc.count	<i>Function for Plotting PSC objects #' A function which illstrates the predicted response under the counter factual model and the observed response under the experimental treatment(s).</i>
----------------	---

Description

Function for Plotting PSC objects #' A function which illstrates 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)
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 function 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

plot_ite *Function for Plotting individual treatment effects for a PSC object*

Description

A function which plots a patients response against their predicted response from a CFM. Exact form of the output will depend on the form of the model supplied

Usage

```
plot_ite(x)
```

Arguments

x an object of class 'psc'

Details

This function plots the expected response of the control treatment along with the observe response rates for each patient in the dataset

Value

A plot showing the individual treatment effects

Examples

```
library(psc)
library(survival)
bin.mod <- psc::bin.mod
data <- psc::data
bin.psc <- pscfit(bin.mod,data)
plot_ite(bin.psc)
```

plot_ite.flexsurvreg *Function for Plotting PSC objects*

Description

A function which plots a patients response against their predicted response from a CFM

Usage

```
plot_ite.flexsurvreg(x)
```

Arguments

x an object of class 'psc'

Details

This function plots the expected response of the control treatment along with the observe response rates for each patient in the dataset

Value

A plot showing the individual treatment effects

Examples

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

plot_ite.glm

Function for Plotting PSC objects

Description

A function which plots a patients response against their predicted response from a CFM. Exact form of the output will depend on the form of the glm

Usage

```
plot_ite.glm(x)
```

Arguments

x an object of class 'psc'

Details

This function plots the expected response of the control treatment along with the observe response rates for each patient in the dataset

Value

A plot showing the individual treatment effects

Examples

```
library(psc)
library(survival)
bin.mod <- psc::bin.mod
data <- psc::data
bin.psc <- pscfit(bin.mod,data)
plot_ite(bin.psc)
```

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 `pscf` function, inheriting from class `psc` and representing a fitted personalised synthetic control model.

Usage

```
psc.object
```

Format

An object of class `NULL` of length 0.

Author(s)

Richard Jaskson (<richj23@liverpool.ac.uk>)

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

Description

This function creates an object containing all information relating to a model which may be used for counter-factual inference. This includes the model parameters, the intended setting for it's, details of validation and references for further details.

Usage

```
pscCFM(cfm, covnm = NULL, setting = NULL, valid = NULL, citation = NULL)
```

Arguments

cfm	a model object
covnm	an optional character vector supplying covariate names to the terms included in the model
setting	an object from 'cfmSett' detailing the setting of the model.
valid	an object from cfmValid which details validation of the model
citation	a character giving a reference (if applicable), defaults to NULL (e.g no citation)

Details

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

pscEst *Function for performing estimation procedures in 'pscfitt'*

Description

Function for performing estimation procedures in 'pscfitt'

Usage

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

Arguments

CFM	a model object supplied to pscfit
DC_clean	a cleaned dataset obtained using dataComb().
nsim	the number of MCMC simulations to run
start	the starting value for
trt	an optional vector denoting treatment allocations where multiple treatment comparisons are being 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 μ 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 distribution for the efficacy parameter ($\pi(\beta)$) is set as an uninformative $N(0, 1000)$.

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 'pscfitt' for more details on likelihood formation.

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

1. Set an 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 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 from 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 synthetic controls methodology.

Usage

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

Arguments

CFM	a model object supplied to pscfit
DC_clean	a cleaned dataset obtained using dataComb().
nsim	the number of MCMC simulations to run
start	the starting value for
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 from 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 synthetic controls methodology.

Usage

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

Arguments

CFM	a model object supplied to pscfit
DC_clean	a cleaned dataset obtained using dataComb().
nsim	the number of MCMC simulations to run
start	the starting value for
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 from the posterior distribution

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 Bayesian MCMC procedure whereby the parameters of the CFM acts as informative priors.

Usage

```
pscfitt(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 estimation restricted to sub-groups of the data cohort (using the 'id' option).

the `pscfitt` 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 parametric survival model of type 'flexsurvreg' and a generalised linear model of type 'glm'.

Where the CFM is of type 'flexsurvreg' the likelihood 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 $b(\cdot)$, $t(\cdot)$ and $c(\cdot)$ 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 details.

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 treatment allocations must be specified separately (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 obtained using dataComb().

Value

psc summary results including an estimate of the linear predictor combining the data and the model, an estimate of patient level response and summary statistics of the average responses for the sythetic 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 cumulative hazard function estimated using flexible parametric splines. Data for the model were synthetically generated and are based on a dataset to evaluate the use of Sorafenib in HCC akin to the PROSASH model

Usage`surv.mod`**Format**

A model of class 'flezsurreg':

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 metastasis

ageVasInv centred age nested within vascular invasion

time survival time

cen censoring indicator

os survival time

count example outcome for count data

trt example identifier for multiple 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	<i>A function to obtain survival estimates from a flexible parametric model</i>
----------	---

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 obtained 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 function which extracts survival probabilities from a flexsurvreg object

Value

a list of times and associated survival probabilities

Examples

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

Index

* datasets

- bin.mod, 3
 - cont.mod, 6
 - count.mod, 7
 - data, 8
 - psc.object, 25
 - surv.mod, 32
- acc, 3
- bin.mod, 3
- cfmSett, 4
- cfmValid, 4
- cfmValid.flexsurvreg, 5
- coef.psc, 6
- cont.mod, 6
- count.mod, 7
- data, 8
- dataComb, 9
- dataComb.flexsurvreg, 9
- dataComb.glm, 10
- dataSumm, 11
- initParm, 11
- initParm.flexsurvreg, 12
- initParm.glm, 13
- lik.flexsurvreg, 13
- lik.flexsurvreg.mtc, 14
- lik.glm, 14
- lik.glm.mtc, 15
- linPred, 15
- modelExtract, 16
- modelExtract.flexsurvreg, 17
- modelExtract.glm, 18
- modp, 18
- plot.psc, 19
- plot.psc.binary, 20
- plot.psc.cont, 20
- plot.psc.count, 21
- plot.psc.flexsurvreg, 22
- plot_ite, 22
- plot_ite.flexsurvreg, 23
- plot_ite.glm, 24
- print.psc, 25
- psc.object, 25
- pscCFM, 26
- pscEst, 26
- pscEst.flexsurvreg, 27
- pscEst.glm, 28
- pscfits, 29
- pscSumm, 30
- summary.psc, 31
- surv.mod, 32
- surv_fpm, 33