

Package ‘psc’

May 9, 2026

Type Package

Title Personalised Synthetic Controls

Version 2.0.1

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 7.3.3

URL <https://github.com/richjjackson/psc/>,
<https://github.com/richJJackson/psc>,
<https://richjjackson.github.io/psc/>

BugReports <https://github.com/richJJackson/psc/issues>

Depends R (>= 4.0.0), survival, ggplot2

Imports mvtnorm, enrichwith, stats, flexsurv, survminer, gtsummary,
RColorBrewer, parallel, ggpubr, posterior, lme4, utils

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

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

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

Repository CRAN

Date/Publication 2026-01-12 15:30:02 UTC

Contents

acc	3
bin.mod	4
boot_lp	4
boot_sest	5
cfmDataSumm	6
cfmDataVis	6
cfmDataVis_fac	7
cfmDataVis_num	7
cfmSumm.flexsurvreg	8
cfmSumm.glm	8
coef.psc	9
cont.mod	9
count.mod	10
data	10
e4_data	11
facVisComp	12
gemCFM	12
init	13
lik.flexsurvreg	13
lik.glm	14
modelExtract	15
modelExtract.flexsurvreg	15
modelExtract.glm	16
modelExtract.lmerMod	16
modp	17
numVisComp	17
plot.psc	18
plot.psc.binary	18
plot.psc.cont	19
plot.psc.count	20
plot.psc.flexsurvreg	20
plotCFM	21
postSummary	22
print.psc	23
print.quiet_gglist	23
print.quiet_gtsumm	24
print.quiet_list	24
psc.object	25
pscCFM	25
pscData	26
pscData_addLik	26
pscData_addtrt	27
pscData_error	27
pscData_match	28
pscData_miss	28
pscData_structure	29

<i>acc</i>	3
pscEst	29
pscEst_run	30
pscEst_samp	31
pscEst_start	32
pscEst_update	33
pscfit	33
spline_surv_est	35
summary.psc	36
surv.mod	36
visComp	37
Index	39

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

Description

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

Usage

`acc(old, new)`

Arguments

<code>old</code>	a numeric value
<code>new</code>	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

*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 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

boot_lp

*Counter Factual Model - summary***Description**

A function to estimate the linear predictor - used in bootstrapping CFM for CIs

Usage

boot_lp(i, pscOb, resp = resp, rest = NULL)

Arguments

i indicator object

pscOb an object of class 'psc'

resp A boolean object to determine if results should be presented on the response scale

rest A matrix of sample covariate estimates

Value

A simulated set of responses

 boot_sest

Counter Factual Model - summary

Description

A function to estimate the survival function based on parameter estimates - used in bootstrapping CFM for CIs

Usage

```
boot_sest(
  i,
  pscOb = pscOb,
  lam = lam,
  kn = kn,
  k = k,
  cov = cov,
  tm = tm,
  rest = rest,
  beta = beta
)
```

Arguments

i	indicator object
pscOb	a pscOb object
lam	parameters of the flexible spline model
kn	knots included in the flexible spline model
k	number of knots in the flexible spline model
cov	a matrix of covariates
tm	time at which to assess the survival function
rest	a set of parameter covariate draws
beta	parameter with which to adjust the baseline function

Value

A set of survival estimates

cfmDataSumm	<i>Summarising data within a Counter Factual Model (CFM)</i>
-------------	--

Description

The pscCFM creates a model object which is stripped of identifiable information. The cfmDataSumm function supplies a tabulated form of the dataset used in the CFM for summary information. Information returned in the form of a table

Usage

```
cfmDataSumm(cfm)
```

Arguments

cfm a 'glm' or 'flexsurvreg' model object

Value

a summary table

cfmDataVis	<i>Visualising data within a CFM</i>
------------	--------------------------------------

Description

The pscCFM creates a model object which is stripped of identifiable information. The cfmDataVis function supplies a visualised form of the dataset for summary information

Usage

```
cfmDataVis(cfm)
```

Arguments

cfm a 'glm' or 'flexsurvreg' model object

Value

a list of grobs for each model covariate

cfmDataVis_fac	<i>Visualising Categorical Data</i>
----------------	-------------------------------------

Description

A function which summarises categorical data using a bar plot. A sub-function of cfmDataVis

Usage

```
cfmDataVis_fac(x, nm)
```

Arguments

x	a covariate to be summarised
nm	a covariate name

Value

a ggplot object

cfmDataVis_num	<i>Visualising Numerical Data</i>
----------------	-----------------------------------

Description

A function which summarises categorical data using density plots. A sub-function of cfmDataVis

Usage

```
cfmDataVis_num(x, nm)
```

Arguments

x	a covariate to be summarised
nm	a covariate name

Value

a ggplot object

cfmSumm.flexsurvreg *Counter Factual Model - summary*

Description

A generic function to provide a summary of a Counter factual model of class 'glm'

Usage

```
cfmSumm.flexsurvreg(pscOb, bootCI = TRUE, nboot = 1000)
```

Arguments

pscOb	an object of class 'psc'
bootCI	a boolean to determine if bootstrapping CIs are required
nboot	Number of bootstraps

Value

A summary of a cfm object

cfmSumm.glm *Counter Factual Model - summary*

Description

A generic function to provide a summary of a Counter factual model of class 'glm'

Usage

```
cfmSumm.glm(pscOb, bootCI = TRUE, nboot = 1000, resp = TRUE)
```

Arguments

pscOb	an object of class 'psc'
bootCI	a boolean to determine if bootstrapping CIs are required
nboot	Number of bootstraps
resp	Should results be on the response scale?

Value

A summary of a cfm object

coef.psc *Returns the coefficient estimate of a psc object.*

Description

Returns basic measures of the posterior distribution obtained from the psc object

Usage

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

Arguments

object	a 'psc' object
...	not used

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 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 'flezsurgreg':

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

e4_data	<i>Example Dataset of patients treated with GemCap in the ESPAC-4 trial</i>
---------	---

Description

A dataset containing 346 simulated patients. Data are based on the patients randomised to receive GemCap in the ESPAC-4 trial

Usage

e4_data

Format

A model of class 'flezsurgreg':

time survival time
cen censoring indicator
nodes negative (n=1) or positive (n=2) lymph nodes
grade tumour grade (1,2 or 3)
lca199 log transformed ca19.9
t T-stage (1,2 or 3)

Source

simulated

facVisComp

Visualising Categorical Data

Description

A function which compares visually a new categorical covariate against equivalent data from a CFM

Usage

```
facVisComp(p, x)
```

Arguments

p a ggplot objects
x a categorical covariate

Value

a ggplot object

gemCFM

Model for a survival outcome based on Gemcitabine patients from ESPAC-3

Description

A generated model with a survival endpoint and a cumulative hazard function estimated using flexible parametric splines. Data for the model were obtained from the ESPAC-3 trials

Usage

```
gemCFM
```

Format

A model of class 'pscCFM' containing a 'flexsurvreg' model:

gamma cumulative baseline hazard parameters
nodes negative (n=1) or positive (n=2) lymph nodes
grade tumour grade (1,2 or 3)
lca199 log transformed ca19.9
ResecM Resection Margins)

Source

simulated

init	<i>Function for estimating initial parameter values</i>
------	---

Description

Function for estimating initial parameter values

Usage

```
init(pscOb)
```

Arguments

pscOb a psc object

Details

This function takes the likelihood and data structures provided by the pscData() structures and fits the likelihood to provide starting values for MCMC estimation

Value

Parameter Estimates and standard error for the efficacy parameter

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 ?pscfits and ?pscEst

Usage

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

Arguments

beta a parameter to be estimate
pscOb A pscOb object containing a cleaned dataset including covariates to match the CFM

Details

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

Value

the results of a likelihood functions

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, pscOb)
```

Arguments

beta	a parameter to be estimate
pscOb	a pscOb object containing a cleaned dataset including covariates to match the CFM

Details

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

Value

the results of a likelihood functions

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

Description

A generic function for extracting model information

Usage

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

Arguments

CFM a model of class either 'flexsurvreg'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

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 either 'glm'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

modelExtract.lmerMod *A generic function for extracting model information*

Description

A generic function for extracting model information

Usage

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

Arguments

CFM a model of class either 'lmer'

Details

A function for extracting the model information required for using pscfit

Value

a list of extracted model components

modp

modp

Description

A function which returns 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

numVisComp

Visualising Numerical Data

Description

A function which summarises numerical data using a density plot

Usage

numVisComp(p, x)

Arguments

p a ggplot object
x a numeric vector

Value

a ggplot object

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

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
psc <- pscfit(gemCFM, e4_data, nsim=1500, nchain=1)
plot(psc)
```

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

Description

A function which illustrates 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(pscOb, ...)
```

Arguments

pscOb	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

plot.psc.cont	<i>Function for Plotting PSC objects</i>
---------------	--

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

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

Description

Function for Plotting PSC objects #' 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.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 observed response rates of the experimental arms

Value

a survival plot corresponding to the psc fit

plot.psc.flexsurvreg	<i>Function for Plotting PSC objects</i>
----------------------	--

Description

Function for Plotting PSC objects

Usage

```
## S3 method for class 'psc.flexsurvreg'
plot(pscOb, addFit = T, ...)
```

Arguments

pscOb	an object of class 'psc'
addFit	should a curve for the model fit be added?
...	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

plotCFM	<i>Function for Plotting PSC objects</i>
---------	--

Description

A function which visualises the data of a CFM or the combined CFM and DC data for a 'psc' object

Usage

```
plotCFM(x, ...)
```

Arguments

x	an object of class 'CFM' or 'psc'
...	not used

Details

This function returns either density plots (continuous data) or bar plots (categorical data) to describe the data in the CFM. If an object is supplied which has combined the CFM and DC (e.g. a psc object or an object which has been passed through pscData()) then a comparison of the CFM and DC will be supplied

Value

a plot to describe the data included in the models

Examples

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
plotCFM(gemCFM)
psc <- pscfit(gemCFM, e4_data, nsim=2000, nchain=1)
plotCFM(psc)
```

postSummary	<i>Posterior Summary</i>
-------------	--------------------------

Description

A function that provides a summary of the posterior distributions obtained from a pscEst() procedure

Usage

```
postSummary(pscOb, thin = 2, burn = 1000, par = "beta")
```

Arguments

pscOb	a pscOb function which has passed through pscEst()
thin	a thin to be applied to the posterior distributions
burn	a burnin to be applied to the posterior distribution
par	the parameter to be summarised - defaults to 'beta' to summarise all 'beta' parameters in the posterior distribution

Details

This function makes use of the 'posterior' package to pull together each of the 'draw' matrices included in the psc object and produce posterior summaries

Value

Returns a summary of a 'psc' object including details on the original Counter Factual Model, a summary of the Data Cohort, the predicted responses from the CFM and details on the model fit.

Examples

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
pscOb <- pscData(gemCFM, e4_data)
pscOb <- init(pscOb)
pscOb <- pscEst(pscOb)
pscOb <- postSummary(pscOb)
```

print.psc	<i>Personalised Synthetic Controls - print</i>
-----------	--

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

print.quiet_gglist	<i>quiet_gglist</i>
--------------------	---------------------

Description

Ensuring a quiet list of the grobs data are supplied to cfmDataVis

Usage

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

Arguments

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

Value

A quiet list

`print.quiet_gtsumm` *quiet_gtsumm*

Description

Ensuring a quiet list of the grobs data are supplied to `cfmDataVis`

Usage

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

Arguments

<code>x</code>	an object of class 'psc'
<code>...</code>	not used

Value

A quiet list

`print.quiet_list` *quiet_gtsumm*

Description

Ensuring a quiet list of the grobs data are supplied to `cfmDataVis`

Usage

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

Arguments

<code>x</code>	an object of class 'psc'
<code>...</code>	not used

Value

A quiet list

psc.object	<i>Fitted psc object</i>
------------	--------------------------

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

pscCFM	<i>Creating a CFM model which can be shared</i>
--------	---

Description

Standard R model objects contain within them the datasets used to create the model and as such care is needed when sharing these objects for research. The `psc.cfm` function creates an object with all identifiable information retracted and includes only the information required to use the models within the `psc` package

Usage

```
pscCFM(CFM, dataSumm = T, dataVis = T)
```

Arguments

CFM	a 'glm' or 'flexsurvreg' model object
dataSumm	a logical indicator specifying whether a summary of the data should be provided, defaults to TRUE.
dataVis	a logical indicator specifying whether a visualisations of the data should be provided, defaults to TRUE.

Value

a list containing objects which specify the required exported components of the model.

pscData	<i>A function which structures the Data Cohort in a format for model estimation</i>
---------	---

Description

This function ensures the data are supplied in a structure which allows for estimation. This is performed by re-fitting the original CFM with the DC and extracting the appropriate structures. Data are returned in terms of "Y" for model outcomes, "X" for data and "Z" for random effects where mixed models are supplied.

Usage

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

Arguments

CFM	a Counter Factual Model
DC	a Data Cohort object
id	to be specified for subgroup analysis. Defaults to NULL
trt	to be specified for multiple treatment comparisons. Defaults to NULL

Value

A set of structures for use with estimation procedures

Examples

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
pscOb <- pscData(gemCFM, e4_data)
```

pscData_addLik	<i>A function that add a likelihood for estimation to the pscObject</i>
----------------	---

Description

The purpose of this function is to include the appropriate likelihood to the psc object for estimation procedures

Usage

```
pscData_addLik(CFM)
```

Arguments

CFM A counter factual model

Value

a likelihood function

pscData_addtrt *A function that includes a treatment indicator when multiple treatment comparisons are required*

Description

The purpose of this function is to organise a treatment indicator where multiple treatment comparisons are being evaluated. This acts as a sub-function to the pscData.R function.

Usage

```
pscData_addtrt(DC, trt)
```

Arguments

DC a data cohort to be 'cleaned'
trt a treatment indicator

Value

a dataset which is checked and compatible with the CFM

pscData_error *A function which performs error checks between the DC and CFM*

Description

The purpose of this function is check that terms included in the Data Cohort match those used within the Counter Factual Model. This acts as a sub-function to the pscData.R function.

Usage

```
pscData_error(term.nm, DC)
```

Arguments

term.nm Term names from the CFM
DC a data cohort to be 'cleaned'

Value

a 'stop' command when errors are detected

pscData_match	<i>A function to ensure that data from the cfm and data cohort are compatible</i>
---------------	---

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. This matches the data classes and checks the levels in the DC match those used in the CFM. This acts as a sub-function to the pscData.R function.

Usage

```
pscData_match(cls, lev, DC)
```

Arguments

cls	a list of extracted data classes
lev	a list of factor levels
DC	a data cohort to be 'cleaned'

Value

a dataset which is checked and compatible with the CFM

pscData_miss	<i>A function which removes missing data from the DC</i>
--------------	--

Description

Currently the psc package works only on complete-case datasets. This function removes rows with missing data and returns a warning to inform the user. This acts as a sub-function to the pscData.R function.

Usage

```
pscData_miss(DC)
```

Arguments

DC	a data cohort to be 'cleaned'
----	-------------------------------

Value

a dataset with missing data removed

pscData_structure	<i>A function which structures the Data Cohort in a format for model estimation</i>
-------------------	---

Description

This function ensures the data are supplied in a structure which allows for estimation. This is performed by re-fitting the original CFM with the DC and extracting the appropriate structures. Data are returned in terms of "Y" for model outcomes, "X" for data and "Z" for random effects where mixed models are supplied.

Usage

```
pscData_structure(CFM, DC)
```

Arguments

CFM	a Counter Factual Model
DC	a Data Cohort object

Value

A set of structures for use with estimation procedures re-export Surv from survival

pscEst	<i>Function for performing Bayesian MCMC estimation procedures in 'pscfitt'</i>
--------	---

Description

Function for performing Bayesian MCMC estimation procedures in 'pscfitt'

Usage

```
pscEst(pscOb, nsim = 1000, nchain = 1)
```

Arguments

pscOb	an pscOb object which has been passed through pscData() and init() functions
nsim	the number of MCMC simulations to run
nchain	Number of chains to use for analysis

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 $\pi(\beta)$. @examples `e4_data <- psc::e4_data gemCFM <- psc::gemCFM pscOb <- pscData(gemCFM,e4_data) pscOb <- init(pscOb) pscOb <- pscEst(pscOb,nsim=1500,nchain=1) importFrom survival Surv survfit`

Value

A matrix containing the draws from the posterior distribution

pscEst_run	<i>Running the Bayesian MCMC routine A procedure which runs the MCMC estimation routine</i>
------------	---

Description

Running the Bayesian MCMC routine A procedure which runs the MCMC estimation routine

Usage

```
pscEst_run(pscOb, nsim, nchain)
```

Arguments

pscOb an pscOb object which has been passed through pscData() and init() functions
 nsim the number of MCMC simulations to run
 nchain Number of chains to use for analysis

Value

An updated set of attributes for the pscOb which includes

Examples

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
pscOb <- pscData(gemCFM,e4_data)
pscOb <- init(pscOb)
pscOb <- pscEst_start(pscOb,nsim=1000,nchain=2)
pscOb <- pscEst_run(pscOb,nsim=1000,nchain=2)
```

pscEst_samp	<i>Starting conditions for Bayesian MCMC estimation procedures in 'pscfir' A procedure which runs the sampling process for MCMC estimation</i>
-------------	--

Description

Starting conditions for Bayesian MCMC estimation procedures in 'pscfir' A procedure which runs the sampling process for MCMC estimation

Usage

```
pscEst_samp(pscOb, nsim)
```

Arguments

pscOb an pscOb object which has been passed through pscData() and init() functions
 nsim the number of MCMC simulations to run

Value

An updated set of attributes for the pscOb which includes

pscEst_start	<i>Starting conditions for Bayesian MCMC estimation procedures in 'pscfitt' A procedure which sets the starting conditions for MCMC estimation</i>
--------------	--

Description

Starting conditions for Bayesian MCMC estimation procedures in 'pscfitt' A procedure which sets the starting conditions for MCMC estimation

Usage

```
pscEst_start(pscOb, nsim, nchain)
```

Arguments

pscOb	an pscOb object which has been passed through pscData() and init() functions
nsim	the number of MCMC simulations to run
nchain	Number of chains to use for analysis

Details

A procedure which sets the starting conditions for MCMC estimation including defining starting estimates, setting a matrix for draws to be save in and defining, target and prior distributions and deifnign the posterior desitribution from the CFM. This also sets the number of cores to be used for estimation where parallel computing is applied.

Value

An updated set of attributes for the pscOb which includes

Examples

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
pscOb <- pscData(gemCFM,e4_data)
pscOb <- init(pscOb)
pscOb <- pscEst_start(pscOb,nsim=1000,nchain=2)
```

pscEst_update	<i>Updating the posterior distribution as part of the MCMC estimation process A procedure which performs a single update of the posterior distribution</i>
---------------	--

Description

Updating the posterior distribution as part of the MCMC estimation process A procedure which performs a single update of the posterior distribution

Usage

```
pscEst_update(i, draws, pscOb)
```

Arguments

i	index of the draw number (i>1)
draws	a matrix containing the posterior draws to update
pscOb	an pscOb object which has been passed through pscData() and init() functions

Value

An updated set of posterior draws

pscfit	<i>Personalised Synthetic Controls model fit</i>
--------	--

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

```
pscfit(
  CFM,
  DC,
  nsim = 2000,
  id = NULL,
  trt = NULL,
  nchain = 2,
  thin = 2,
  burn = 500
)
```

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.
nchain	Number of chains used in posterior MCMC estimation. Defaults to nchain=3.
thin	Thin applied to posterior draws. Defaults to thin=2.
burn	Number of posterior samples to use as burn-in. Defaults to burn=500

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 `pscffit` 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 required, 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

```
e4_data <- psc::e4_data
gemCFM <- psc::gemCFM
psc <- pscfit(gemCFM,e4_data,nsim=1500,nchain=1)
print(psc)
```

spline_surv_est

Counter Factual Model - summary

Description

A function to estimate the survival function based on parameter estimates - used in ootstrapping CFM for CIs

Usage

```
spline_surv_est(lam, kn, k, haz_co, cov_co, cov = cov, tm = tm, beta = 0)
```

Arguments

lam	parameters of the flexible spline model
kn	knots included in the flexible spline model
k	number of knots in the flexible spline model
haz_co	parameters for the baseline hazard function in the flexible spline model
cov_co	covariate parameters of the flexible spline model
cov	a matrix of covaraites from the Data Cohort
tm	time at which to assess the survival function
beta	parameter with which to adjust the baseline function (defaults to beta=0)

Value

A data frame containing survival estimates for a give time

`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

<code>object</code>	an object of class 'psc'
<code>...</code>	not used

Value

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

Examples

```
e4_data <- psc::e4_data  
gemCFM <- psc::gemCFM  
psc <- pscfit(gemCFM, e4_data, nsim=1500, nchain=1)  
summary(psc)
```

`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 'flezsurgreg':

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

visComp

Visualising Comparisons between a CFM and a DC

Description

The visComp function takes the data visualisations supplied as part of the CFM model and appends summaries of the equivalent datapoints from the Data Cohort.

Usage

```
visComp(CFM, DC, id = NULL)
```

Arguments

CFM	an object of class pscCFM
DC	A dataset including columns to match to covariates in the model
id	Numeric vector stating which patient(s) from the dataset should be included in the analysis. Defaults to all patients

Value

a list of grobs for each model covariate

Index

* datasets

- bin.mod, 4
 - cont.mod, 9
 - count.mod, 10
 - data, 10
 - e4_data, 11
 - gemCFM, 12
 - psc.object, 25
 - surv.mod, 36
- acc, 3
- bin.mod, 4
- boot_lp, 4
- boot_sest, 5
- cfmDataSumm, 6
- cfmDataVis, 6
- cfmDataVis_fac, 7
- cfmDataVis_num, 7
- cfmSumm.flexsurvreg, 8
- cfmSumm.glm, 8
- coef.psc, 9
- cont.mod, 9
- count.mod, 10
- data, 10
- e4_data, 11
- facVisComp, 12
- gemCFM, 12
- init, 13
- lik.flexsurvreg, 13
- lik.glm, 14
- modelExtract, 15
- modelExtract.flexsurvreg, 15
- modelExtract.glm, 16
- modelExtract.lmerMod, 16
- modp, 17
- numVisComp, 17
- plot.psc, 18
- plot.psc.binary, 18
- plot.psc.cont, 19
- plot.psc.count, 20
- plot.psc.flexsurvreg, 20
- plotCFM, 21
- postSummary, 22
- print.psc, 23
- print.quiet_gglist, 23
- print.quiet_gtsumm, 24
- print.quiet_list, 24
- psc.object, 25
- pscCFM, 25
- pscData, 26
- pscData_addLik, 26
- pscData_addtrt, 27
- pscData_error, 27
- pscData_match, 28
- pscData_miss, 28
- pscData_structure, 29
- pscEst, 29
- pscEst_run, 30
- pscEst_samp, 31
- pscEst_start, 32
- pscEst_update, 33
- pscfits, 33
- spline_surv_est, 35
- summary.psc, 36
- surv.mod, 36
- visComp, 37