Package 'beastt'

February 24, 2025

Title Bayesian Evaluation, Analysis, and Simulation Software Tools for Trials

Version 0.0.2

Description Bayesian dynamic borrowing with covariate adjustment via inverse probability weighting for simulations and data analyses in clinical trials. This makes it easy to use propensity score methods to balance covariate distributions between external and internal data.

License GPL (>= 3)

URL https://gsk-biostatistics.github.io/beastt/, https://github.com/GSK-Biostatistics/beastt

BugReports https://github.com/GSK-Biostatistics/beastt/issues

Suggests knitr, mvtnorm, rmarkdown, spelling, testthat (>= 3.0.0), tibble, vdiffr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

Imports cli, cobalt, distributional, dplyr, generics, ggdist, ggplot2, methods, mixtools, purrr, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rlang, rstan (>= 2.18.1), rstantools (>= 2.4.0), stringr, tidyr

VignetteBuilder knitr

Depends R (>= 4.1.0)

LazyData true

Language en-US

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

NeedsCompilation yes

Author Christina Fillmore [aut, cre] (<https://orcid.org/0000-0003-0595-2302>),
Ben Arancibia [aut],
Nate Bean [aut] (<https://orcid.org/0000-0001-9946-0119>),
Abi Terry [aut],
GlaxoSmithKline Research & Development Limited [cph, fnd],
Trustees of Columbia University [cph] (R/stanmodels.R, configure,
configure.win)

Maintainer Christina Fillmore <christina.e.fillmore@gsk.com>

Repository CRAN

Date/Publication 2025-02-24 15:30:02 UTC

Contents

beastt-package *The 'beastt' package.*

Description

Inverse Probability Weighted Bayesian Dynamic Borrowing

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.3. https://mc-stan.org

calc_post_beta *Calculate Posterior Beta*

Description

Calculate a posterior distribution that is beta (or a mixture of beta components). Only the relevant treatment arms from the internal dataset should be read in (e.g., only the control arm if constructing a posterior distribution for the control response rate).

Usage

```
calc_post_beta(internal_data, response, prior)
```

Arguments

`internal_data`	A tibble of the internal data.
`response`	Name of response variable
`prior`	A distributional object corresponding to a beta distribution or a mixture distribution of beta components

Details

For a given arm of an internal trial (e.g., the control arm or an active treatment arm) of size N_I, suppose the response data are binary such that $Y_i \sim \text{Bernoulli}(\theta)$, $i = 1, \ldots, N_I$. The posterior distribution for θ is written as

$$\pi(\theta \mid \boldsymbol{y}) \propto \mathcal{L}(\theta \mid \boldsymbol{y})\, \pi(\theta),$$

where $\mathcal{L}(\theta \mid \boldsymbol{y})$ is the likelihood of the response data from the internal arm and $\pi(\theta)$ is a prior distribution on θ (either a beta distribution or a mixture distribution with an arbitrary number of beta components). The posterior distribution for θ is either a beta distribution or a mixture of beta components depending on whether the prior is a single beta distribution or a mixture distribution.

Value

distributional object

Examples

```
library(dplyr)
library(distributional)
calc_post_beta(internal_data = filter(int_binary_df, trt == 1),
               response = y,
               prior = dist_beta(0.5, 0.5))
```

calc_post_norm *Calculate Posterior Normal*

Description

Calculate a posterior distribution that is normal (or a mixture of normal components). Only the relevant treatment arms from the internal dataset should be read in (e.g., only the control arm if constructing a posterior distribution for the control mean).

Usage

```
calc_post_norm(internal_data, response, prior, internal_sd = NULL)
```

Arguments

internal_data	A tibble of the internal data.
response	Name of response variable
prior	A distributional object corresponding to a normal distribution, a t distribution, or a mixture distribution of normal and/or t components
internal_sd	Standard deviation of internal response data if assumed known. It can be left as NULL if assumed unknown

Details

For a given arm of an internal trial (e.g., the control arm or an active treatment arm) of size N_I, suppose the response data are normally distributed such that $Y_i \sim N(\theta, \sigma_I^2)$, $i = 1, \ldots, N_I$. If σ_I^2 is assumed known, the posterior distribution for θ is written as

$$\pi(\theta \mid \boldsymbol{y}, \sigma_I^2) \propto \mathcal{L}(\theta \mid \boldsymbol{y}, \sigma_I^2)\ \pi(\theta),$$

where $\mathcal{L}(\theta \mid \boldsymbol{y}, \sigma_I^2)$ is the likelihood of the response data from the internal arm and $\pi(\theta)$ is a prior distribution on θ (either a normal distribution, a t distribution, or a mixture distribution with an arbitrary number of normal and/or t components). Any t components of the prior for θ are approximated with a mixture of two normal distributions.

If σ_I^2 is unknown, the marginal posterior distribution for θ is instead written as

$$\pi(\theta \mid \boldsymbol{y}) \propto \left\{ \int_0^\infty \mathcal{L}(\theta, \sigma_I^2 \mid \boldsymbol{y})\; \pi(\sigma_I^2)\; d\sigma_I^2 \right\} \times \pi(\theta).$$

In this case, the prior for σ_I^2 is chosen to be $\pi(\sigma_I^2) = (\sigma_I^2)^{-1}$ such that $\left\{ \int_0^\infty \mathcal{L}(\theta, \sigma_I^2 \mid \boldsymbol{y})\; \pi(\sigma_I^2)\; d\sigma_I^2 \right\}$ becomes a non-standardized t distribution. This integrated likelihood is then approximated with a mixture of two normal distributions.

If `internal_sd` is supplied a positive value and `prior` corresponds to a single normal distribution, then the posterior distribution for θ is a normal distribution. If `internal_sd = NULL` or if other types of prior distributions are specified (e.g., mixture or t distribution), then the posterior distribution is a mixture of normal distributions.

Value

distributional object

Examples

```
library(distributional)
library(dplyr)
post_treated <- calc_post_norm(internal_data = filter(int_norm_df, trt == 1),
                               response = y,
                               prior = dist_normal(50, 10),
                               internal_sd = 0.15)
```

`calc_post_weibull` *Calculate Posterior Weibull*

Description

Calculate a posterior distribution for the probability of surviving past a given analysis time(s) for time-to-event data with a Weibull likelihood. Only the relevant treatment arms from the internal dataset should be read in (e.g., only the control arm if constructing a posterior distribution for the control survival probability).

Usage

```
calc_post_weibull(internal_data, response, event, prior, analysis_time, ...)
```

Arguments

`internal_data` This can either be a propensity score object or a tibble of the internal data.

`response` Name of response variable

`event` Name of event indicator variable (1: event; 0: censored) `prior` A distributional object corresponding to a multivariate normal distribution or a mixture of 2 multivariate normals. The first element of the mean vector and the first row/column of covariance matrix correspond to the log-shape parameter, and the second element corresponds to the intercept (i.e., log-inverse-scale) parameter.

`analysis_time` Vector of time(s) when survival probabilities will be calculated

`...` rstan sampling option. This overrides any default options. For more information, see `rstan::sampling()`

Details

For a given arm of an internal trial (e.g., the control arm or an active treatment arm) of size N_I, suppose the response data are time to event such that $Y_i \sim \text{Weibull}(\alpha, \sigma)$, where

$$f(y_i \mid \alpha, \sigma) = \left(\frac{\alpha}{\sigma}\right)\left(\frac{y_i}{\sigma}\right)^{\alpha-1} \exp\left(-\left(\frac{y_i}{\sigma}\right)^{\alpha}\right),$$

$i = 1, \ldots, N_I$. Define $\boldsymbol{\theta} = \{\log(\alpha), \beta\}$ where $\beta = -\log(\sigma)$ is the intercept parameter of a Weibull proportional hazards regression model. The posterior distribution for $\boldsymbol{\theta}$ is written as

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{\nu}) \propto \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{\nu})\; \pi(\boldsymbol{\theta}),$$

where $\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{\nu}) = \prod_{i=1}^{N_I} f(y_i \mid \boldsymbol{\theta})^{\nu_i} S(y_i \mid \boldsymbol{\theta})^{1-\nu_i}$ is the likelihood of the response data from the internal arm with event indicator $\boldsymbol{\nu}$ and survival function $S(y_i \mid \boldsymbol{\theta}) = 1 - F(y_i \mid \boldsymbol{\theta})$, and $\pi(\boldsymbol{\theta})$ is a prior distribution on $\boldsymbol{\theta}$ (either a multivariate normal distribution or a mixture of two multivariate normal distributions). Note that the posterior distribution for $\boldsymbol{\theta}$ does not have a closed form, and thus MCMC samples for $\log(\alpha)$ and β are drawn from the posterior. These MCMC samples are used to construct samples from the posterior distribution for the probability of surviving past the specified analysis time(s) for the given arm.

To construct a posterior distribution for the treatment difference (i.e., the difference in survival probabilities at the specified analysis time), obtain MCMC samples from the posterior distributions for the survival probabilities under each arm and then subtract the control-arm samples from the treatment-arm samples.

Value

stan posterior object

Examples

```
library(distributional)
library(dplyr)
library(rstan)
mvn_prior <- dist_multivariate_normal(
   mu = list(c(0.3, -2.6)),
   sigma = list(matrix(c(1.5, 0.3, 0.3, 1.1), nrow = 2)))
post_treated <- calc_post_weibull(filter(int_tte_df, trt == 1),
                                  response = y,
                                  event = event,
                                  prior = mvn_prior,
                                  analysis_time = 12,
                                  warmup = 5000,
                                  iter = 15000)
# Extract MCMC samples of survival probabilities at time t=12
surv_prob_treated <- as.data.frame(extract(post_treated,
                                   pars = c("survProb")))[,1]
```

`calc_power_prior_beta` *Calculate Power Prior Beta*

Description

Calculate a (potentially inverse probability weighted) beta power prior for the control response rate using external control data.

Usage

```
calc_power_prior_beta(external_data, response, prior)
```

Arguments

`external_data`	This can either be a `prop_scr_obj` created by calling `create_prop_scr()` or a tibble of the external data. If it is just a tibble the weights will be assumed to be 1.
`response`	Name of response variable
`prior`	A beta distributional object that is the initial prior for the control response rate before the external control data are observed

Details

Weighted participant-level response data from the control arm of an external study are incorporated into an inverse probability weighted (IPW) power prior for the control response rate θ_C. When borrowing information from an external control arm of size N_{EC}, the components of the IPW power prior for θ_C are defined as follows:

Initial prior:

$$\theta_C \sim \text{Beta}(\nu_0, \phi_0)$$

IPW likelihood of the external response data $\boldsymbol{y}_E$ with weights $\hat{\boldsymbol{a}}_0$:

$$\mathcal{L}_E(\theta_C \mid \boldsymbol{y}_E, \hat{\boldsymbol{a}}_0) \propto \exp\left(\sum_{i=1}^{N_{EC}} \hat{a}_{0i}\left[y_i \log(\theta_C) + (1 - y_i)\log(1 - \theta_C)\right]\right)$$

IPW power prior:

$$\theta_C \mid \boldsymbol{y}_E, \hat{\boldsymbol{a}}_0 \sim \text{Beta}\left(\sum_{i=1}^{N_{EC}} \hat{a}_{0i} y_i + \nu_0, \sum_{i=1}^{N_{EC}} \hat{a}_{0i}(1 - y_i) + \phi_0\right)$$

Defining the weights $\hat{\boldsymbol{a}}_0$ to equal 1 results in a conventional beta power prior.

Value

Beta power prior object

See Also

Other power prior: calc_power_prior_norm(), calc_power_prior_weibull()

Examples

```
library(distributional)
library(dplyr)
# This function can be used directly on the data
calc_power_prior_beta(external_data = ex_binary_df,
                      response = y,
                      prior = dist_beta(0.5, 0.5))

# Or this function can be used with a propensity score object
ps_obj <- calc_prop_scr(internal_df = filter(int_binary_df, trt == 0),
                        external_df = ex_binary_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)

calc_power_prior_beta(ps_obj,
                      response = y,
                      prior = dist_beta(0.5, 0.5))
```

calc_power_prior_norm *Calculate Power Prior Normal*

Description

Calculate a (potentially inverse probability weighted) normal power prior using external data.

Usage

```
calc_power_prior_norm(
  external_data,
  response,
  prior = NULL,
  external_sd = NULL
)
```

Arguments

external_data This can either be a prop_scr_obj created by calling create_prop_scr() or a tibble of the external data. If it is just a tibble the weights will be assumed to be 1. Only the external data for the arm(s) of interest should be included in this object (e.g., external control data if creating a power prior for the control mean) `response` Name of response variable

`prior` Either `NULL` or a normal distributional object that is the initial prior for the parameter of interest (e.g., control mean) before the external data are observed

`external_sd` Standard deviation of external response data if assumed known. It can be left as `NULL` if assumed unknown

Details

Weighted participant-level response data from an external study are incorporated into an inverse probability weighted (IPW) power prior for the parameter of interest θ (e.g., the control mean if borrowing from an external control arm). When borrowing information from an external dataset of size N_E, the IPW likelihood of the external response data $\boldsymbol{y}_E$ with weights $\hat{\boldsymbol{a}}_0$ is defined as

$$\mathcal{L}_E(\theta \mid \boldsymbol{y}_E, \hat{\boldsymbol{a}}_0, \sigma_E^2) \propto \exp\left(-\frac{1}{2\sigma_E^2}\sum_{i=1}^{N_E}\hat{a}_{0i}(y_i-\theta)^2\right).$$

The `prior` argument should be either a distributional object with a family type of `normal` or `NULL`, corresponding to the use of a normal initial prior or an improper uniform initial prior (i.e., $\pi(\theta) \propto 1$), respectively.

The `external_sd` argument can be a positive value if the external standard deviation is assumed known or left as `NULL` otherwise. If `external_sd` = `NULL`, then `prior` must be `NULL` to indicate the use of an improper uniform initial prior for θ, and an improper prior is defined for the unknown external standard deviation such that $\pi(\sigma_E^2) \propto (\sigma_E^2)^{-1}$. The details of the IPW power prior for each case are as follows:

`external_sd = positive value` **(σ_E^2 known):** With either a proper normal or an improper uniform initial prior, the IPW weighted power prior for θ is a normal distribution.

`external_sd` = `NULL` **(σ_E^2 unknown):** With improper priors for both θ and σ_E^2, the marginal IPW weighted power prior for θ after integrating over σ_E^2 is a non-standardized t distribution.

Defining the weights $\hat{\boldsymbol{a}}_0$ to equal 1 results in a conventional normal (or t) power prior if the external standard deviation is known (unknown).

Value

Normal power prior object

See Also

Other power prior: `calc_power_prior_beta()`, `calc_power_prior_weibull()`

Examples

```
library(distributional)
library(dplyr)
# This function can be used directly on the data
calc_power_prior_norm(ex_norm_df,
                      response = y,
                      prior = dist_normal(50, 10),
                      external_sd = 0.15)

# Or this function can be used with a propensity score object
ps_obj <- calc_prop_scr(internal_df = filter(int_norm_df, trt == 0),
                        external_df = ex_norm_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)
calc_power_prior_norm(ps_obj,
                      response = y,
                      prior = dist_normal(50, 10),
                      external_sd = 0.15)
```

calc_power_prior_weibull

Calculate Power Prior Weibull

Description

Calculate an approximate (potentially inverse probability weighted) multivariate normal power prior for the log-shape and log-inverse-scale parameters of a Weibull likelihood for external time-to-event control data.

Usage

```
calc_power_prior_weibull(
  external_data,
  response,
  event,
  intercept,
  shape,
  approximation = c("Laplace", "MCMC"),
  ...
)
```

Arguments

`external_data` This can either be a `prop_scr_obj` created by calling `create_prop_scr()` or a tibble of the external data. If it is just a tibble the weights will be assumed to be 1. Only the external data for the arm(s) of interest should be included in this object (e.g., external control data if creating a power prior for the control Weibull shape and intercept parameters)

`response` Name of response variable

`event` Name of event indicator variable (1: event; 0: censored)

`intercept` Normal distributional object that is the initial prior for the intercept (i.e., log-inverse-scale) parameter

`shape` Integer value that is the scale of the half-normal prior for the shape parameter `approximation` Type of approximation to be used. Either `Laplace` or `MCMC`. `Laplace` is used by default because it is faster than `MCMC`.

`...` Arguments passed to `rstan::sampling` (e.g. iter, chains).

Details

Weighted participant-level response data from the control arm of an external study are incorporated into an approximated inverse probability weighted (IPW) power prior for the parameter vector $\boldsymbol{\theta}_C = \{\log(\alpha), \beta\}$, where $\beta = -\log(\sigma)$ is the intercept parameter of a Weibull proportional hazards regression model and α and σ are the Weibull shape and scale parameters, respectively. When borrowing information from an external dataset of size N_E, the IPW likelihood of the external response data $\boldsymbol{y}_E$ with event indicators $\boldsymbol{\nu}_E$ and weights $\hat{\boldsymbol{a}}_0$ is defined as

$$\mathcal{L}_E(\alpha, \sigma \mid \boldsymbol{y}_E, \boldsymbol{\nu}_E, \hat{\boldsymbol{a}}_0) \propto \prod_{i=1}^{N_E} \left\{ \left(\frac{\alpha}{\sigma}\right) \left(\frac{y_i}{\sigma}\right)^{\alpha-1} \exp\left(-\left(\frac{y_i}{\sigma}\right)^{\alpha}\right) \right\}^{\hat{a}_{0i}\nu_i} \left\{ \exp\left(-\left(\frac{y_i}{\sigma}\right)^{\alpha}\right) \right\}^{\hat{a}_{0i}(1-\nu_i)}.$$

The initial priors for the intercept parameter β and the shape parameter α are assumed to be normal and half-normal, respectively, which can be defined using the `intercept` and `shape` arguments.

The power prior for $\boldsymbol{\theta}_C$ does not have a closed form, and thus we approximate it via a bivariate normal distribution; i.e.,

$$\boldsymbol{\theta}_C \mid \boldsymbol{y}_E, \boldsymbol{\nu}_E, \hat{\boldsymbol{a}}_0 \;\dot{\sim}\; \text{MVN}\left(\tilde{\boldsymbol{\mu}}_0, \tilde{\boldsymbol{\Sigma}}_0\right)$$

. If `approximation` = `Laplace`, then $\tilde{\boldsymbol{\mu}}_0$ is the mode vector of the IPW power prior and $\tilde{\boldsymbol{\Sigma}}_0$ is the negative inverse of the Hessian of the log IPW power prior evaluated at the mode. If `approximation` = `MCMC`, then MCMC samples are obtained from the IPW power prior, and $\tilde{\boldsymbol{\mu}}_0$ and $\tilde{\boldsymbol{\Sigma}}_0$ are the estimated mean vector and covariance matrix of these MCMC samples. Note that the Laplace approximation method is faster due to its use of optimization instead of MCMC sampling.

The first element of the mean vector and the first row/column of covariance matrix correspond to the log-shape parameter ($\log(\alpha)$), and the second element corresponds to the intercept (β, the log-inverse-scale) parameter.

Value

Multivariate Normal Distributional Object

See Also

Other power prior: `calc_power_prior_beta()`, `calc_power_prior_norm()`

Examples

```
library(distributional)
library(dplyr)
# This function can be used directly on the data
calc_power_prior_weibull(ex_tte_df,
                         response = y,
                         event = event,
                         intercept = dist_normal(0, 10),
                         shape = 50,
                         approximation = ”Laplace”)

# Or this function can be used with a propensity score object
ps_obj <- calc_prop_scr(internal_df = filter(int_tte_df, trt == 0),
                        external_df = ex_tte_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)
calc_power_prior_weibull(ps_obj,
                         response = y,
                         event = event,
                         intercept = dist_normal(0, 10),
                         shape = 50,
                         approximation = ”Laplace”)
```

`calc_prop_scr` *Create a Propensity Score Object*

Description

Calculate the propensity scores and ATT inverse probability weights for participants from internal and external datasets. Only the relevant treatment arms from each dataset should be read in (e.g., only the control arm from each dataset if creating a hybrid control arm).

Usage

```
calc_prop_scr(internal_df, external_df, id_col, model, ...)
```

Arguments

`internal_df` Internal dataset with one row per subject and all the variables needed to run the model

`external_df` External dataset with one row per subject and all the variables needed to run the model

`id_col` Name of the column in both datasets used to identify each subject. It must be the same across datasets

`model` Model used to calculate propensity scores

`...` Optional arguments

Details

For the subset of participants in both the external and internal studies for which we want to balance the covariate distributions (e.g., external control and internal control participants if constructing a hybrid control arm), we define a study-inclusion propensity score for each participant as

$$e(x_i) = P(S_i = 1 \mid x_i),$$

where x_i denotes a vector of baseline covariates for the ith participant and S_i denotes the indicator that the participant is enrolled in the internal trial ($S_i = 1$ if internal, $S_i = 0$ if external). The estimated propensity score $\hat{e}(x_i)$ is obtained using logistic regression.

An ATT inverse probability weight is calculated for each individual as

$$\hat{a}_{0i} = \frac{\hat{e}(x_i)}{\hat{P}(S_i = s_i|x_i)} = s_i + (1 - s_i)\frac{\hat{e}(x_i)}{1 - \hat{e}(x_i)}.$$

In a weighted estimator, data from participants in the external study are given a weight of $\hat{e}(x_i)/(1-\hat{e}(x_i))$ whereas data from participants in the internal trial are given a weight of 1.

Value

`prop_scr_obj` object, with the internal and the external data and the propensity score and inverse probability weight calculated for each subject.

Examples

```
# This can be used for both continuous and binary data
library(dplyr)
# Continuous
calc_prop_scr(internal_df = filter(int_norm_df, trt == 0),
                       external_df = ex_norm_df,
                       id_col = subjid,
                       model = ~ cov1 + cov2 + cov3 + cov4)
# Binary
calc_prop_scr(internal_df = filter(int_binary_df, trt == 0),
                       external_df = ex_binary_df,
                       id_col = subjid,
                       model = ~ cov1 + cov2 + cov3 + cov4)
```

`ex_binary_df` *External Binary Control Data for Propensity Score Balancing*

Description

This is a simulated dataset used to illustrate Bayesian dynamic borrowing in the case when borrowing from an external control arm with a binary endpoint, where the baseline covariate distributions of the internal and external data are balanced via inverse probability weighting.

Usage

```
ex_binary_df
```

Format

`ex_binary_df`:

A data frame with 150 rows and 6 columns:

subjid Unique subject ID

cov1 Covariate 1, which is normally distributed around 65 with a SD of 10

cov2 Covariate 2, which is binary (0 vs. 1) with about 30% of participants having level 1

cov3 Covariate 3, which is binary (0 vs. 1) with about 40% of participants having level 1

cov4 Covariate 4, which is binary (0 vs. 1) with about 50% of participants having level 1

y Response, which is binary (0 vs. 1)

`ex_norm_df` *External Normal Control Data for Propensity Score Balancing*

Description

This is a simulated dataset used to illustrate Bayesian dynamic borrowing in the case when borrowing from an external control arm with a normal endpoint, where the baseline covariate distributions of the internal and external data are balanced via inverse probability weighting.

Usage

```
ex_norm_df
```

Format

`ex_norm_df`:

A data frame with 150 rows and 6 columns:

subjid Unique subject ID

cov1 Covariate 1, which is normally distributed around 50 with a SD of 10

cov2 Covariate 2, which is binary (0 vs. 1) with about 20% of participants having level 1

cov3 Covariate 3, which is binary (0 vs. 1) with about 60% of participants having level 1

cov4 Covariate 4, which is binary (0 vs. 1) with about 30% of participants having level 1

y Response, which is normally distributed with a SD of 0.15

`ex_tte_df` *External Time-to-Event Control Data for Propensity Score Balancing*

Description

This is a simulated dataset used to illustrate Bayesian dynamic borrowing in the case when borrowing from an external control arm with a time-to-event endpoint, where the baseline covariate distributions of the internal and external data are balanced via inverse probability weighting.

Usage

```
ex_tte_df
```

Format

`ex_tte_df`**:**

A data frame with 150 rows and 9 columns:

subjid Unique subject ID

y Response (observed time at which the participant either had an event or was censored)

enr_time Enrollment time

total_time Time from study start

event Event indicator (1: event; 0: censored)

cov1 Covariate 1, which is normally distributed around 65 with a SD of 10

cov2 Covariate 2, which is binary (0 vs. 1) with about 30% of participants having level 1

cov3 Covariate 3, which is binary (0 vs. 1) with about 40% of participants having level 1

cov4 Covariate 4, which is binary (0 vs. 1) with about 50% of participants having level 1

`int_binary_df` *Internal Binary Data for Propensity Score Balancing*

Description

This is a simulated dataset used to illustrate Bayesian dynamic borrowing in the case when borrowing from an external control arm with a binary endpoint, where the baseline covariate distributions of the internal and external data are balanced via inverse probability weighting.

Usage

```
int_binary_df
```

Format

`int_binary_df`:

A data frame with 160 rows and 7 columns:

- **subjid** Unique subject ID
- **cov1** Covariate 1, which is normally distributed around 62 with an sd of 8
- **cov2** Covariate 2, which is binary (0 vs. 1) with about 40% of participants having level 1
- **cov3** Covariate 3, which is binary (0 vs. 1) with about 40% of participants having level 1
- **cov4** Covariate 4, which is binary (0 vs. 1) with about 60% of participants having level 1
- **trt** Treatment indicator, where 0 = control and 1 = active treatment
- **y** Response, which is binary (0 vs. 1)

`int_norm_df` *Internal Normal Data for Propensity Score Balancing*

Description

This is a simulated dataset used to illustrate Bayesian dynamic borrowing in the case when borrowing from an external control arm with a normal endpoint, where the baseline covariate distributions of the internal and external data are balanced via inverse probability weighting.

Usage

```
int_norm_df
```

Format

`int_norm_df`:

A data frame with 120 rows and 7 columns:

- **subjid** Unique subject ID
- **cov1** Covariate 1, which is normally distributed around 55 with a SD of 8
- **cov2** Covariate 2, which is binary (0 vs. 1) with about 30% of participants having level 1
- **cov3** Covariate 3, which is binary (0 vs. 1) with about 50% of participants having level 1
- **cov4** Covariate 4, which is binary (0 vs. 1) with about 30% of participants having level 1
- **trt** Treatment indicator, where 0 = control and 1 = active treatment
- **y** Response, which is normally distributed with a SD of 0.15

`int_tte_df` *Internal Time-to-Event Control Data for Propensity Score Balancing*

Description

This is a simulated dataset used to illustrate Bayesian dynamic borrowing in the case when borrowing from an external control arm with a time-to-event endpoint, where the baseline covariate distributions of the internal and external data are balanced via inverse probability weighting.

Usage

```
int_tte_df
```

Format

`int_tte_df`**:**

A data frame with 160 rows and 10 columns:

subjid Unique subject ID

y Response (observed time at which the participant either had an event or was censored)

enr_time Enrollment time

total_time Time from study start

event Event indicator (1: event; 0: censored)

trt Treatment indicator, where 0 = control and 1 = active treatment

cov1 Covariate 1, which is normally distributed around 62 with a SD of 8

cov2 Covariate 2, which is binary (0 vs. 1) with about 40% of participants having level 1

cov3 Covariate 3, which is binary (0 vs. 1) with about 40% of participants having level 1

cov4 Covariate 4, which is binary (0 vs. 1) with about 60% of participants having level 1

`is_prop_scr` *Test If Propensity Score Object*

Description

Test If Propensity Score Object

Usage

```
is_prop_scr(x)
```

Arguments

`x`	Object to test

Value

Boolean

Examples

```
library(dplyr)
x <- calc_prop_scr(internal_df = filter(int_norm_df, trt == 0),
                       external_df = ex_norm_df,
                       id_col = subjid,
                       model = ~ cov1 + cov2 + cov3 + cov4)
is_prop_scr(x)
```

mix_means *Extract Means of Mixture Components*

Description

Extract Means of Mixture Components

Usage

```
mix_means(x)
```

Arguments

x	A mixture distributional object

Details

If a distributional object that is a mixture of two or more normal distributions is read in, the function will return a numeric object with the means of each normal component. If the distributional object is a mixture of two or more multivariate normal distributions, the function will return a list with the mean vectors of each multivariate normal component.

Value

numeric or list object

Examples

```
library(distributional)
mix_norm <- dist_mixture(comp1 = dist_normal(1, 10),
                         comp2 = dist_normal(1.5, 12),
                         weights = c(.5, .5))
mix_means(mix_norm)
```

mix_sigmas *Extract Standard Deviations of Mixture Components*

Description

Extract Standard Deviations of Mixture Components

Usage

```
mix_sigmas(x)
```

Arguments

x	A mixture distributional object

Details

If a distributional object that is a mixture of two or more normal distributions is read in, the function will return a numeric object with the standard deviations of each normal component. If the distributional object is a mixture of two or more multivariate normal distributions, the function will return a list with the covariance matrices of each multivariate normal component.

Value

numeric or list object

Examples

```
library(distributional)
mix_norm <- dist_mixture(comp1 = dist_normal(1, 10),
                         comp2 = dist_normal(1.5, 12),
                         weights = c(.5, .5))
mix_sigmas(mix_norm)
```

plot_dist *Plot Distribution*

Description

Plot Distribution

Usage

```
plot_dist(...)
```

Arguments

...	Distributional object(s) to plot. When passing multiple objects naming them will change the labels in the plot, else they will use the distributional format

Value

ggplot object that is the density of the provided distribution

Examples

```
library(distributional)
plot_dist(dist_normal(0, 1))
#Plotting Multiple
plot_dist(dist_normal(0, 1), dist_normal(10, 5))
plot_dist('Prior' = dist_normal(0, 1), 'Posterior' = dist_normal(10, 5))
```

prop_scr_dens *Density of the Propensity Score Object*

Description

Plot overlapping density curves of the propensity scores for both the internal and external participants, or plot external IPWs.

Usage

```
prop_scr_dens(
  x,
  variable = c("propensity score", "ps", "inverse probability weight", "ipw"),
  ...
)
```

Arguments

x	Propensity score object
variable	Variable to plot. It must be either a propensity score ("ps" or "propensity score") or inverse probability weight ("ipw" or "inverse probability weight")
...	Optional arguments for geom_density

Value

ggplot object

Examples

```
library(dplyr)
ps_obj <- calc_prop_scr(internal_df = filter(int_norm_df, trt == 0),
                        external_df = ex_norm_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)
# Plotting the Propensity Scores
prop_scr_dens(ps_obj)
# Or plotting the inverse probability weights
prop_scr_dens(ps_obj, variable = "ipw")
```

prop_scr_hist *Histogram of the Propensity Score Object*

Description

Plot overlapping histograms of the propensity scores for both the internal and external participants, or plot external IPWs.

Usage

```
prop_scr_hist(
  x,
  variable = c("propensity score", "ps", "inverse probability weight", "ipw"),
  ...
)
```

Arguments

`x`	Propensity score object
`variable`	Variable to plot. It must be either a propensity score ("ps" or "propensity score") or inverse probability weight ("ipw" or "inverse probability weight")
`...`	Optional arguments for `geom_histogram`

Value

ggplot object

Examples

```
library(dplyr)
ps_obj <- calc_prop_scr(internal_df = filter(int_norm_df, trt == 0),
                        external_df = ex_norm_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)
# Plotting the Propensity Scores
prop_scr_hist(ps_obj)
```

```
# Or plotting the inverse probability weights
prop_scr_hist(ps_obj, variable = "ipw")
```

prop_scr_love *Love Plot of the Absolute Standardized Mean Differences*

Description

Plot the unadjusted and IPW-adjusted absolute standardized mean differences for each covariate.

Usage

```
prop_scr_love(x, reference_line = NULL, ...)
```

Arguments

`x` Propensity score object

`reference_line` Numeric value of where along the x-axis the vertical reference line should be placed

`...` Optional options for `geom_point`

Value

ggplot object

Examples

```
library(dplyr)
ps_obj <- calc_prop_scr(internal_df = filter(int_norm_df, trt == 0),
                        external_df = ex_norm_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)
# Plotting the Propensity Scores
prop_scr_love(ps_obj, reference_line = 0.1)
```

robustify_mvnorm *Robustify Multivariate Normal Distributions*

Description

Adds vague normal component, where the level of vagueness is controlled by the n parameter

Usage

```
robustify_mvnorm(prior, n, weights = c(0.5, 0.5))
```

Arguments

prior	Multivariate Normal distributional object
n	Number of theoretical participants (or events, for time-to-event data)
weights	Vector of weights, where the first number corresponds to the informative component and the second is the vague

Details

In cases with a time-to-event endpoint, a robust mixture prior can be created by adding a vague multivariate normal component to any multivariate normal prior with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The vague component is calculated to have the same mean vector $\boldsymbol{\mu}$ and covariance matrix equal to $\boldsymbol{\Sigma} \times n$, where n is the specified number of theoretical events.

Value

mixture distribution

Examples

```
library(distributional)
robustify_mvnorm(
      dist_multivariate_normal(mu = list(c(1, 0)), sigma = list(c(10, 5))),
       n = 15)
```

robustify_norm *Robustify Normal Distributions*

Description

Adds vague normal component, where the level of vagueness is controlled by the n parameter

Usage

```
robustify_norm(prior, n, weights = c(0.5, 0.5))
```

Arguments

`prior` Normal or Multivariate Normal distributional object

`n` Number of theoretical participants (or events, for time-to-event data)

`weights` Vector of weights, where the first number corresponds to the informative component and the second is the vague

Details

In cases with a normal endpoint, a robust mixture prior can be created by adding a vague normal component to any normal prior with mean θ and variance σ^2.The vague component is calculated to have the same mean θ and variance equal to $\sigma^2 \times n$, where `n` is the specified number of theoretical participants. If robustifying a normal power prior that was calculated from external control data and `n` is defined as the number of external control participants, and the vague component would then correspond to one external control participant's worth of data.

Value

mixture distribution

Examples

```
library(distributional)
robustify_norm(dist_normal(0,1), n = 15)
```

tidy.prop_scr *Tidy a(n) prop_scr object*

Description

Tidy a(n) prop_scr object

Usage

```
## S3 method for class 'prop_scr'
tidy(x, ...)
```

Arguments

`x` a `prop_scr` obj

`...` Unused, included for generic consistency only.

Value

A tidy `tibble::tibble()` summarizing the results of the propensity score weighting. The tibble will have the id column of the external data, an `internal` column to indicate all the data is external, a `ps` column with the propensity scores and a `weight` column with the inverse probability weights

Examples

```
library(dplyr)
ps_obj <- calc_prop_scr(internal_df = filter(int_binary_df, trt == 0),
                        external_df = ex_binary_df,
                        id_col = subjid,
                        model = ~ cov1 + cov2 + cov3 + cov4)
tidy(ps_obj)
```

Index