# Package 'bayesDP'

February 3, 2020

**Type** Package

**Title** Tools for the Bayesian Discount Prior Function

**Version** 1.3.3

**Date** 2020-01-29

**Depends** R (>= 3.6.0), ggplot2, survival, methods

**Description** Functions for data augmentation using the
Bayesian discount prior function for 1 arm and 2 arm clinical trials.

**LazyLoad** yes

**License** GPL-3

**URL** https://github.com/graemeleehickey/bayesDP

**BugReports** https://github.com/graemeleehickey/bayesDP/issues

**NeedsCompilation** yes

**Repository** CRAN

**Imports** Rcpp, MCMCpack

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.0.2

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**Author** Shawn Balcome [aut],
Donnie Musgrove [aut],
Tarek Haddad [aut],
Hickey Graeme L. [cre, aut] (<https://orcid.org/0000-0002-4989-0054>),
Christopher Jackson [ctb] (For the ppexp R code that was ported to
C++.)

**Maintainer** Hickey Graeme L. <graemeleehickey@gmail.com>

**Date/Publication** 2020-02-03 19:50:07 UTC

# R topics documented:

---

alpha_discount                 *Bayesian Discount Prior: Historical Data Weight (alpha)*

---

## Description

alpha_discount can be used to estimate the weight applied to historical data in the context of a
one- or two-arm clinical trial. alpha_discount is not used internally but is given for educational
purposes.

## Usage

```
alpha_discount(
  p_hat = NULL,
  discount_function = "weibull",
  alpha_max = 1,
  weibull_scale = 0.135,
  weibull_shape = 3
)
```

## Arguments

p_hat              scalar. The posterior probability of a stochastic comparison. This value can be the output of `posterior_probability` or a value between 0 and 1.

discount_function
                   character. Specify the discount function to use. Currently supports `weibull`, `scaledweibull`, and `identity`. The discount function `scaledweibull` scales the output of the Weibull CDF to have a max value of 1. The `identity` discount function uses the posterior probability directly as the discount weight. Default value is `"weibull"`.

alpha_max          scalar. Maximum weight the discount function can apply. Default is 1.

weibull_scale      scalar. Scale parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 0.135.

weibull_shape      scalar. Shape parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 3.

## Details

This function is not used internally but is given for educational purposes. Given inputs `p_hat`, `discount_function`, `alpha_max`, `weibull_shape`, and `weibull_scale` the output is the weight that would be applied to historical data in the context of a one- or two-arm clinical trial.

## Value

`alpha_discount` returns an object of class "alpha_discount".

An object of class `alpha_discount` contains the following:

`alpha_hat`  scalar. The historical data weight.

## References

Haddad, T., Himes, A., Thompson, L., Irony, T., Nair, R. MDIC Computer Modeling and Simulation working group.(2017) Incorporation of stochastic engineering models as prior information in Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 1-15.

## Examples

```
alpha_discount(0.5)

alpha_discount(0.5, discount_function="identity")
```

---

bdpbinomial                        *Bayesian Discount Prior: Binomial counts*

---

### Description

bdpbinomial is used for estimating posterior samples from a binomial outcome where an informative prior is used. The prior weight is determined using a discount function. This code is modeled after the methodologies developed in Haddad et al. (2017).

### Usage

```
bdpbinomial(
  y_t = NULL,
  N_t = NULL,
  y0_t = NULL,
  N0_t = NULL,
  y_c = NULL,
  N_c = NULL,
  y0_c = NULL,
  N0_c = NULL,
  discount_function = "identity",
  alpha_max = 1,
  fix_alpha = FALSE,
  a0 = 1,
  b0 = 1,
  number_mcmc = 10000,
  weibull_scale = 0.135,
  weibull_shape = 3,
  method = "mc",
  compare = TRUE
)
```

### Arguments

| | |
|---|---|
| y_t | scalar. Number of events for the current treatment group. |
| N_t | scalar. Sample size of the current treatment group. |
| y0_t | scalar. Number of events for the historical treatment group. |
| N0_t | scalar. Sample size of the historical treatment group. |
| y_c | scalar. Number of events for the current control group. |
| N_c | scalar. Sample size of the current control group. |
| y0_c | scalar. Number of events for the historical control group. |
| N0_c | scalar. Sample size of the historical control group. |

discount_function

character. Specify the discount function to use. Currently supports `weibull`, `scaledweibull`, and `identity`. The discount function `scaledweibull` scales the output of the Weibull CDF to have a max value of 1. The `identity` discount function uses the posterior probability directly as the discount weight. Default value is `"identity"`.

alpha_max        scalar. Maximum weight the discount function can apply. Default is 1. For a two-arm trial, users may specify a vector of two values where the first value is used to weight the historical treatment group and the second value is used to weight the historical control group.

fix_alpha        logical. Fix alpha at alpha_max? Default value is FALSE.

a0               scalar. Prior value for the beta rate. Default is 1.

b0               scalar. Prior value for the beta rate. Default is 1.

number_mcmc      scalar. Number of Monte Carlo simulations. Default is 10000.

weibull_scale    scalar. Scale parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 0.135. For a two-arm trial, users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when `discount_function = "identity"`.

weibull_shape    scalar. Shape parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 3. For a two-arm trial, users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when `discount_function = "identity"`.

method           character. Analysis method with respect to estimation of the weight paramter alpha. Default method `"mc"` estimates alpha for each Monte Carlo iteration. Alternate value `"fixed"` estimates alpha once and holds it fixed throughout the analysis. See the the `bdpbinomial` vignette `vignette("bdpbinomial-vignette",package="bayesDP")` for more details.

compare          logical. Should a comparison object be included in the fit? For a one-arm analysis, the comparison object is simply the posterior chain of the treatment group parameter. For a two-arm analysis, the comparison object is the posterior chain of the treatment effect that compares treatment and control. If `compare=TRUE`, the comparison object is accessible in the `final` slot, else the `final` slot is NULL. Default is TRUE.

### Details

`bdpbinomial` uses a two-stage approach for determining the strength of historical data in estimation of a binomial count mean outcome. In the first stage, a *discount function* is used that that defines the maximum strength of the historical data and discounts based on disagreement with the current data. Disagreement between current and historical data is determined by stochastically comparing the respective posterior distributions under noninformative priors. With binomial data, the comparison is the proability (p) that the current count is less than the historical count. The comparison metric

p is then input into the Weibull discount function and the final strength of the historical data is returned (alpha).

In the second stage, posterior estimation is performed where the discount function parameter, `alpha`, is used incorporated in all posterior estimation procedures.

To carry out a single arm (OPC) analysis, data for the current treatment (`y_t` and `N_t`) and historical treatment (`y0_t` and `N0_t`) must be input. The results are then based on the posterior distribution of the current data augmented by the historical data.

To carry our a two-arm (RCT) analysis, data for the current treatment and at least one of current or historical control data must be input. The results are then based on the posterior distribution of the difference between current treatment and control, augmented by available historical data.

For more details, see the `bdpbinomial` vignette:
`vignette("bdpbinomial-vignette",package="bayesDP")`

**Value**

bdpbinomial returns an object of class "bdpbinomial". The functions [summary](#) and [print](#) are used to obtain and print a summary of the results, including user inputs. The [plot](#) function displays visual outputs as well.

An object of class `bdpbinomial` is a list containing at least the following components:

posterior_treatment  list. Entries contain values related to the treatment group:

- `alpha_discount` numeric. Alpha value, the weighting parameter of the historical data.
- `p_hat` numeric. The posterior probability of the stochastic comparison between the current and historical data.
- `posterior` vector. A vector of length `number_mcmc` containing posterior Monte Carlo samples of the event rate of the treatment group. If historical treatment data is present, the posterior incorporates the weighted historical data.
- `posterior_flat` vector. A vector of length `number_mcmc` containing Monte Carlo samples of the event rate of the current treatment group under a flat/non-informative prior, i.e., no incorporation of the historical data.
- `prior` vector. If historical treatment data is present, a vector of length `number_mcmc` containing Monte Carlo samples of the event rate of the historical treatment group under a flat/non-informative prior.

posterior_control  list. Similar entries as `posterior_treament`. Only present if a control group is specified.

final  list. Contains the final comparison object, dependent on the analysis type:

- One-arm analysis: vector. Posterior chain of binomial proportion.
- Two-arm analysis: vector. Posterior chain of binomial proportion difference comparing treatment and control groups.

args1  list. Entries contain user inputs. In addition, the following elements are ouput:

- `arm2` binary indicator. Used internally to indicate one-arm or two-arm analysis.
- `intent` character. Denotes current/historical status of treatment and control groups.

### References

Haddad, T., Himes, A., Thompson, L., Irony, T., Nair, R. MDIC Computer Modeling and Simulation working group.(2017) Incorporation of stochastic engineering models as prior information in Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 1-15.

### See Also

summary, print, and plot for details of each of the supported methods.

### Examples

```
# One-arm trial (OPC) example
fit <- bdpbinomial(y_t    = 10,
                   N_t    = 500,
                   y0_t   = 25,
                   N0_t   = 250,
                   method = "fixed")
summary(fit)
print(fit)
## Not run:
plot(fit)

## End(Not run)

# Two-arm (RCT) example
fit2 <- bdpbinomial(y_t    = 10,
                    N_t    = 500,
                    y0_t   = 25,
                    N0_t   = 250,
                    y_c    = 8,
                    N_c    = 500,
                    y0_c   = 20,
                    N0_c   = 250,
                    method = "fixed")
summary(fit2)
print(fit2)
## Not run:
plot(fit2)

## End(Not run)
```

---

bdplm                  *Bayesian Discount Prior: Two-Arm Linear Regression*

---

### Description

bdplm is used to estimate the treatment effect in the presence of covariates using the regression analysis implementation of the Bayesian discount prior. summary and print methods are supported.

Currently, the function only supports a two-arm clinical trial where all of current treatment, current control, historical treatment, and historical control data are present

## Usage

```
bdplm(
  formula = formula,
  data = data,
  data0 = NULL,
  prior_treatment_effect = NULL,
  prior_control_effect = NULL,
  prior_treatment_sd = NULL,
  prior_control_sd = NULL,
  prior_covariate_effect = 0,
  prior_covariate_sd = 10000,
  number_mcmc_alpha = 5000,
  number_mcmc_sigmagrid = 5000,
  number_mcmc_sigma = 100,
  number_mcmc_beta = 10000,
  discount_function = "identity",
  alpha_max = 1,
  fix_alpha = FALSE,
  weibull_scale = 0.135,
  weibull_shape = 3,
  method = "mc"
)
```

## Arguments

formula       an object of class "formula." See "Details" for more information, including spec-
              ification of treatment data indicators.

data          a data frame containing the current data variables in the model. A column named
              `treatment` must be present; `treatment` must be binary and indicate treatment
              group vs. control group.

data0         a data frame containing the historical data variables in the model. The column
              labels of data and data0 must match.

prior_treatment_effect
              scalar. The historical adjusted treatment effect. If left NULL, value is estimated
              from the historical data.

prior_control_effect
              scalar. The historical adjusted control effect. If left NULL, value is estimated
              from the historical data.

prior_treatment_sd
              scalar. The standard deviation of the historical adjusted treatment effect. If left
              NULL, value is estimated from the historical data.

prior_control_sd
              scalar. The standard deviation of the historical adjusted control effect. If left
              NULL, value is estimated from the historical data.

prior_covariate_effect

        vector. The prior mean(s) of the covariate effect(s). Default value is zero. If a single value is input, the the scalar is repeated to the length of the input covariates. Otherwise, care must be taken to ensure the length of the input matches the number of covariates.

prior_covariate_sd

        vector. The prior standard deviation(s) of the covariate effect(s). Default value is 1e4. If a single value is input, the the scalar is repeated to the length of the input covariates. Otherwise, care must be taken to ensure the length of the input matches the number of covariates.

number_mcmc_alpha

        scalar. Number of Monte Carlo simulations for estimating the historical data weight. Default is 5000.

number_mcmc_sigmagrid

        scalar. Grid size for computing sigma. Default is 5000. See "Details" for more information.

number_mcmc_sigma

        scalar. Number of Monte Carlo simulations for estimating sigma. Default is 1000. See "Details" for more information.

number_mcmc_beta

        scalar. Number of Monte Carlo simulations for estimating beta, the vector of regression coefficients. Default is 10000.

discount_function

        character. Specify the discount function to use. Currently supports `weibull`, `scaledweibull`, and `identity`. The discount function `scaledweibull` scales the output of the Weibull CDF to have a max value of 1. The `identity` discount function uses the posterior probability directly as the discount weight. Default value is "identity".

alpha_max         scalar. Maximum weight the discount function can apply. Default is 1. Users may specify a vector of two values where the first value is used to weight the historical treatment group and the second value is used to weight the historical control group.

fix_alpha         logical. Fix alpha at alpha_max? Default value is FALSE.

weibull_scale   scalar. Scale parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 0.135. Users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when `discount_function` = "identity".

weibull_shape   scalar. Shape parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 3. Users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when `discount_function` = "identity".

method         character. Analysis method with respect to estimation of the weight paramter alpha. Default method "mc" estimates alpha for each Monte Carlo iteration. Alternate value "fixed" estimates alpha once and holds it fixed throughout the

analysis. See the the bdplm vignette
vignette("bdplm-vignette",package="bayesDP") for more details.

**Details**

bdplm uses a two-stage approach for determining the strength of historical data in estimation of an adjusted mean or covariate effect. In the first stage, a *discount function* is used that that defines the maximum strength of the historical data and discounts based on disagreement with the current data. Disagreement between current and historical data is determined by stochastically comparing the respective posterior distributions under noninformative priors. Here with a two-arm regression analysis, the comparison is the proability (p) that the covariate effect of an historical data indicator is significantly different from zero. The comparison metric p is then input into the discount function and the final strength of the historical data is returned (alpha).

In the second stage, posterior estimation is performed where the discount function parameter, alpha, is used to weight the historical data effects.

The formula must include an intercept (i.e., do not use -1 in the formula) and both data and data0 must be present. The column names of data and data0 must match. See examples below for example usage.

The underlying model results in a marginal posterior distribution for the error variance sigma2 that does not have a known distribution. Thus, we use a grid search to draw samples of sigma2. First, the marginal posterior is evaluated at a grid of number_mcmc_sigmagrid sigma2 values. The bounds of the grid are taken as approximately six standard deviations from an estimate of sigma2 using the lm function. Next, number_mcmc_sigma posterior draws of sigma2 are made by sampling with replacement from the grid with each value having the corresponding marginal posterior probability of being selected. With posterior draws of sigma2 in hand, we can make posterior draws of the regression coefficients.

**Value**

bdplm returns an object of class "bdplm".

An object of class "bdplm" is a list containing at least the following components:

posterior  data frame. The posterior draws of the covariates, the intercept, and the treatment effect. The grid of sigma values are included.

alpha_discount  vector. The posterior probability of the stochastic comparison between the current and historical data for each of the treatment and control arms. If method="mc", the result is a matrix of estimates, otherwise for method="fixed", the result is a vector of estimates.

estimates  list. The posterior means and standard errors of the intercept, treatment effect, covariate effect(s) and error variance.

**Examples**

```
# Set sample sizes
n_t  <- 30    # Current treatment sample size
n_c  <- 30    # Current control sample size
n_t0 <- 80    # Historical treatment sample size
n_c0 <- 80    # Historical control sample size
```

```
# Treatment group vectors for current and historical data
treatment   <- c(rep(1,n_t), rep(0,n_c))
treatment0  <- c(rep(1,n_t0), rep(0,n_c0))

# Simulate a covariate effect for current and historical data
x  <- rnorm(n_t+n_c, 1, 5)
x0 <- rnorm(n_t0+n_c0, 1, 5)

# Simulate outcome:
# - Intercept of 10 for current and historical data
# - Treatment effect of 31 for current data
# - Treatment effect of 30 for historical data
# - Covariate effect of 3 for current and historical data
Y  <- 10 + 31*treatment  + x*3 + rnorm(n_t+n_c,0,5)
Y0 <- 10 + 30*treatment0 + x0*3 + rnorm(n_t0+n_c0,0,5)

# Place data into separate treatment and control data frames and
# assign historical = 0 (current) or historical = 1 (historical)
df_ <- data.frame(Y=Y, treatment=treatment, x=x)
df0 <- data.frame(Y=Y0, treatment=treatment0, x=x0)

# Fit model using default settings
fit <- bdplm(formula=Y ~ treatment+x, data=df_, data0=df0,
             method="fixed")

# Look at estimates and discount weight
summary(fit)
print(fit)
```

---

bdplogit                    *Bayesian Discount Prior: Two-Arm Logistic Regression*

---

### Description

bdplogit is used to estimate the treatment effect in the presence of covariates using the logistic regression analysis implementation of the Bayesian discount prior. summary and print methods are supported. Currently, the function only supports a two-arm clinical trial where all of current treatment, current control, historical treatment, and historical control data are present

### Usage

```
bdplogit(
  formula = formula,
  data = data,
  data0 = NULL,
  prior_treatment_effect = NULL,
  prior_control_effect = NULL,
  prior_treatment_sd = NULL,
```

```
    prior_control_sd = NULL,
    prior_covariate_effect = 0,
    prior_covariate_sd = 10000,
    number_mcmc_alpha = 5000,
    number_mcmc_beta = 10000,
    discount_function = "identity",
    alpha_max = 1,
    fix_alpha = FALSE,
    weibull_scale = 0.135,
    weibull_shape = 3,
    method = "mc"
)
```

## Arguments

| | |
|---|---|
| formula | an object of class "formula." See "Details" for more information, including specification of treatment data indicators. |
| data | a data frame containing the current data variables in the model. A column named `treatment` must be present; `treatment` must be binary and indicate treatment group vs. control group. |
| data0 | a data frame containing the historical data variables in the model. The column labels of data and data0 must match. |
| prior_treatment_effect | |
| | scalar. The historical adjusted treatment effect. If left NULL, value is estimated from the historical data. |
| prior_control_effect | |
| | scalar. The historical adjusted control effect. If left NULL, value is estimated from the historical data. |
| prior_treatment_sd | |
| | scalar. The standard deviation of the historical adjusted treatment effect. If left NULL, value is estimated from the historical data. |
| prior_control_sd | |
| | scalar. The standard deviation of the historical adjusted control effect. If left NULL, value is estimated from the historical data. |
| prior_covariate_effect | |
| | vector. The prior mean(s) of the covariate effect(s). Default value is zero. If a single value is input, the the scalar is repeated to the length of the input covariates. Otherwise, care must be taken to ensure the length of the input matches the number of covariates. |
| prior_covariate_sd | |
| | vector. The prior standard deviation(s) of the covariate effect(s). Default value is 1e4. If a single value is input, the the scalar is repeated to the length of the input covariates. Otherwise, care must be taken to ensure the length of the input matches the number of covariates. |
| number_mcmc_alpha | |
| | scalar. Number of Monte Carlo simulations for estimating the historical data weight. Default is 5000. |

number_mcmc_beta

>    scalar. Number of Monte Carlo simulations for estimating beta, the vector of regression coefficients. Default is 10000.

discount_function

>    character. Specify the discount function to use. Currently supports `weibull`, `scaledweibull`, and `identity`. The discount function `scaledweibull` scales the output of the Weibull CDF to have a max value of 1. The `identity` discount function uses the posterior probability directly as the discount weight. Default value is `"identity"`.

alpha_max       scalar. Maximum weight the discount function can apply. Default is 1. Users may specify a vector of two values where the first value is used to weight the historical treatment group and the second value is used to weight the historical control group.

fix_alpha       logical. Fix alpha at alpha_max? Default value is FALSE.

weibull_scale   scalar. Scale parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 0.135. Users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when `discount_function` = "identity".

weibull_shape   scalar. Shape parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 3. Users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when `discount_function` = "identity".

method          character. Analysis method with respect to estimation of the weight paramter alpha. Default method `"mc"` estimates alpha for each Monte Carlo iteration. Alternate value `"fixed"` estimates alpha once and holds it fixed throughout the analysis. See the the `bdplm` vignette
`vignette("bdplm-vignette",package="bayesDP")` for more details.

## Details

`bdplogit` uses a two-stage approach for determining the strength of historical data in estimation of an adjusted mean or covariate effect. In the first stage, a *discount function* is used that that defines the maximum strength of the historical data and discounts based on disagreement with the current data. Disagreement between current and historical data is determined by stochastically comparing the respective posterior distributions under noninformative priors. Here with a two-arm regression analysis, the comparison is the proability (p) that the covariate effect of an historical data indicator is significantly different from zero. The comparison metric p is then input into the discount function and the final strength of the historical data is returned (`alpha`).

In the second stage, posterior estimation is performed where the discount function parameter, `alpha`, is used to weight the historical data effects.

The formula must include an intercept (i.e., do not use -1 in the formula) and both data and data0 must be present. The column names of data and data0 must match. See `examples` below for example usage.

The underlying model uses the `MCMClogit` function of the MCMCpack package to carryout posterior estimation. Add more.

**Value**

`bdplogit` returns an object of class "bdplogit".

An object of class "bdplogit" is a list containing at least the following components:

`posterior` data frame. The posterior draws of the covariates, the intercept, and the treatment effect. The grid of sigma values are included.

`alpha_discount` vector. The posterior probability of the stochastic comparison between the current and historical data for each of the treatment and control arms. If `method="mc"`, the result is a matrix of estimates, otherwise for `method="fixed"`, the result is a vector of estimates.

`estimates` list. The posterior means and standard errors of the intercept, treatment effect, covariate effect(s) and error variance.

**Examples**

```
# Set sample sizes
n_t  <- 30     # Current treatment sample size
n_c  <- 30     # Current control sample size
n_t0 <- 80     # Historical treatment sample size
n_c0 <- 80     # Historical control sample size

# Treatment group vectors for current and historical data
treatment   <- c(rep(1,n_t), rep(0,n_c))
treatment0  <- c(rep(1,n_t0), rep(0,n_c0))

# Simulate a covariate effect for current and historical data
x  <- rnorm(n_t+n_c, 1, 5)
x0 <- rnorm(n_t0+n_c0, 1, 5)

# Simulate outcome:
# - Intercept of 10 for current and historical data
# - Treatment effect of 31 for current data
# - Treatment effect of 30 for historical data
# - Covariate effect of 3 for current and historical data
Y  <- 10 + 31*treatment  + x*3 + rnorm(n_t+n_c,0,5)
Y0 <- 10 + 30*treatment0 + x0*3 + rnorm(n_t0+n_c0,0,5)

# Place data into separate treatment and control data frames and
# assign historical = 0 (current) or historical = 1 (historical)
df_ <- data.frame(Y=Y, treatment=treatment, x=x)
df0 <- data.frame(Y=Y0, treatment=treatment0, x=x0)

# Fit model using default settings
fit <- bdplm(formula=Y ~ treatment+x, data=df_, data0=df0,
             method="fixed")

# Look at estimates and discount weight
summary(fit)
```

```
print(fit)
```

---

bdpnormal                    *Bayesian Discount Prior: Gaussian mean values*

---

### Description

bdpnormal is used for estimating posterior samples from a Gaussian outcome where an informative prior is used. The prior weight is determined using a discount function. This code is modeled after the methodologies developed in Haddad et al. (2017).

### Usage

```
bdpnormal(
  mu_t = NULL,
  sigma_t = NULL,
  N_t = NULL,
  mu0_t = NULL,
  sigma0_t = NULL,
  N0_t = NULL,
  mu_c = NULL,
  sigma_c = NULL,
  N_c = NULL,
  mu0_c = NULL,
  sigma0_c = NULL,
  N0_c = NULL,
  discount_function = "identity",
  alpha_max = 1,
  fix_alpha = FALSE,
  weibull_scale = 0.135,
  weibull_shape = 3,
  number_mcmc = 10000,
  method = "mc",
  compare = TRUE
)
```

### Arguments

| | |
|---|---|
| mu_t | scalar. Mean of the current treatment group. |
| sigma_t | scalar. Standard deviation of the current treatment group. |
| N_t | scalar. Number of observations of the current treatment group. |
| mu0_t | scalar. Mean of the historical treatment group. |
| sigma0_t | scalar. Standard deviation of the historical treatment group. |
| N0_t | scalar. Number of observations of the historical treatment group. |

| | |
|---|---|
| mu_c | scalar. Mean of the current control group. |
| sigma_c | scalar. Standard deviation of the current control group. |
| N_c | scalar. Number of observations of the current control group. |
| mu0_c | scalar. Mean of the historical control group. |
| sigma0_c | scalar. Standard deviation of the historical control group. |
| N0_c | scalar. Number of observations of the historical control group. |
| discount_function | |
| | character. Specify the discount function to use. Currently supports weibull, scaledweibull, and identity. The discount function scaledweibull scales the output of the Weibull CDF to have a max value of 1. The identity discount function uses the posterior probability directly as the discount weight. Default value is "identity". |
| alpha_max | scalar. Maximum weight the discount function can apply. Default is 1. For a two-arm trial, users may specify a vector of two values where the first value is used to weight the historical treatment group and the second value is used to weight the historical control group. |
| fix_alpha | logical. Fix alpha at alpha_max? Default value is FALSE. |
| weibull_scale | scalar. Scale parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 0.135. For a two-arm trial, users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when discount_function = "identity". |
| weibull_shape | scalar. Shape parameter of the Weibull discount function used to compute alpha, the weight parameter of the historical data. Default value is 3. For a two-arm trial, users may specify a vector of two values where the first value is used to estimate the weight of the historical treatment group and the second value is used to estimate the weight of the historical control group. Not used when discount_function = "identity". |
| number_mcmc | scalar. Number of Monte Carlo simulations. Default is 10000. |
| method | character. Analysis method with respect to estimation of the weight paramter alpha. Default method "mc" estimates alpha for each Monte Carlo iteration. Alternate value "fixed" estimates alpha once and holds it fixed throughout the analysis. See the the bdpnormal vignette vignette("bdpnormal-vignette",package="bayesDP") for more details. |
| compare | logical. Should a comparison object be included in the fit? For a one-arm analysis, the comparison object is simply the posterior chain of the treatment group parameter. For a two-arm analysis, the comparison object is the posterior chain of the treatment effect that compares treatment and control. If compare=TRUE, the comparison object is accessible in the final slot, else the final slot is NULL. Default is TRUE. |

### Details

bdpnormal uses a two-stage approach for determining the strength of historical data in estimation of a mean outcome. In the first stage, a *discount function* is used that that defines the maximum

strength of the historical data and discounts based on disagreement with the current data. Disagreement between current and historical data is determined by stochastically comparing the respective posterior distributions under noninformative priors. With Gaussian data, the comparison is the proability (p) that the current mean is less than the historical mean. The comparison metric p is then input into the discount function and the final strength of the historical data is returned (alpha).

In the second stage, posterior estimation is performed where the discount function parameter, `alpha`, is used incorporated in all posterior estimation procedures.

To carry out a single arm (OPC) analysis, data for the current treatment (`mu_t`, `sigma_t`, and `N_t`) and historical treatment (`mu0_t`, `sigma0_t`, and `N0_t`) must be input. The results are then based on the posterior distribution of the current data augmented by the historical data.

To carry our a two-arm (RCT) analysis, data for the current treatment and at least one of current or historical control data must be input. The results are then based on the posterior distribution of the difference between current treatment and control, augmented by available historical data.

For more details, see the bdpnormal vignette:
`vignette("bdpnormal-vignette",package="bayesDP")`

## Value

`bdpnormal` returns an object of class "bdpnormal". The functions [summary]{.link} and [print]{.link} are used to obtain and print a summary of the results, including user inputs. The [plot]{.link} function displays visual outputs as well.

An object of class `bdpnormal` is a list containing at least the following components:

`posterior_treatment` list. Entries contain values related to the treatment group:

- `alpha_discount` numeric. Alpha value, the weighting parameter of the historical data.
- `p_hat` numeric. The posterior probability of the stochastic comparison between the current and historical data.
- `posterior_mu` vector. A vector of length `number_mcmc` containing the posterior mean of the treatment group. If historical treatment data is present, the posterior incorporates the weighted historical data.
- `posterior_sigma2` vector. A vector of length `number_mcmc` containing the posterior variance of the treatment group. If historical treatment data is present, the posterior incorporates the weighted historical data.
- `posterior_flat_mu` vector. A vector of length `number_mcmc` containing Monte Carlo samples of the mean of the current treatment group under a flat/non-informative prior, i.e., no incorporation of the historical data.
- `posterior_flat_sigma2` vector. A vector of length `number_mcmc` containing Monte Carlo samples of the standard deviation of the current treatment group under a flat/non-informative prior, i.e., no incorporation of the historical data.
- `prior_mu` vector. If historical treatment data is present, a vector of length `number_mcmc` containing Monte Carlo samples of the mean of the historical treatment group under a flat/non-informative prior.
- `prior_sigma2` vector. If historical treatment data is present, a vector of length `number_mcmc` containing Monte Carlo samples of the standard deviation of the historical treatment group under a flat/non-informative prior.

posterior_control list. Similar entries as posterior_treament. Only present if a control group
is specified.

final list. Contains the final comparison object, dependent on the analysis type:

- One-arm analysis: vector. Posterior chain of the mean.
- Two-arm analysis: vector. Posterior chain of the mean difference comparing treatment
and control groups.

args1 list. Entries contain user inputs. In addition, the following elements are ouput:

- arm2 binary indicator. Used internally to indicate one-arm or two-arm analysis.
- intent character. Denotes current/historical status of treatment and control groups.

### References

Haddad, T., Himes, A., Thompson, L., Irony, T., Nair, R. MDIC Computer Modeling and Simula-
tion working group.(2017) Incorporation of stochastic engineering models as prior information in
Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 1-15.

### See Also

summary, print, and plot for details of each of the supported methods.

### Examples

```
# One-arm trial (OPC) example
fit <- bdpnormal(mu_t  = 30, sigma_t  = 10,  N_t = 50,
                 mu0_t = 32, sigma0_t = 10, N0_t = 50,
                 method = "fixed")
summary(fit)
## Not run:
plot(fit)

## End(Not run)

# Two-arm (RCT) example
fit2 <- bdpnormal(mu_t  = 30, sigma_t  = 10,  N_t = 50,
                  mu0_t = 32, sigma0_t = 10, N0_t = 50,
                  mu_c  = 25, sigma_c  = 10,  N_c = 50,
                  mu0_c = 25, sigma0_c = 10, N0_c = 50,
                  method = "fixed")
summary(fit2)
## Not run:
plot(fit2)

## End(Not run)
```

---

| bdpsurvival | *Bayesian Discount Prior: Survival Analysis* |

---

**Description**

bdpsurvival is used to estimate the survival probability (single arm trial; OPC) or hazard ratio (two-arm trial; RCT) for right-censored data using the survival analysis implementation of the Bayesian discount prior. In the current implementation, a two-arm analysis requires all of current treatment, current control, historical treatment, and historical control data. This code is modeled after the methodologies developed in Haddad et al. (2017).

**Usage**

```
bdpsurvival(
  formula = formula,
  data = data,
  data0 = NULL,
  breaks = NULL,
  a0 = 0.1,
  b0 = 0.1,
  surv_time = NULL,
  discount_function = "identity",
  alpha_max = 1,
  fix_alpha = FALSE,
  number_mcmc = 10000,
  weibull_scale = 0.135,
  weibull_shape = 3,
  method = "mc",
  compare = TRUE
)
```

**Arguments**

| | |
|---|---|
| formula | an object of class "formula." Must have a survival object on the left side and at most one input on the right side, treatment. See "Details" for more information. |
| data | a data frame containing the current data variables in the model. Columns denoting 'time' and 'status' must be present. See "Details" for required structure. |
| data0 | optional. A data frame containing the historical data variables in the model. If present, the column labels of data and data0 must match. |
| breaks | vector. Breaks (interval starts) used to compose the breaks of the piecewise exponential model. Do not include zero. Default breaks are the quantiles of the input times. |
| a0 | scalar. Prior value for the gamma shape of the piecewise exponential hazards. Default is 0.1. |
| b0 | scalar. Prior value for the gamma rate of the piecewise exponential hazards. Default is 0.1. |

surv_time        scalar. Survival time of interest for computing the probability of survival for a
                 single arm (OPC) trial. Default is overall, i.e., current+historical, median sur-
                 vival time.

discount_function

                 character. Specify the discount function to use. Currently supports weibull,
                 scaledweibull, and identity. The discount function scaledweibull scales
                 the output of the Weibull CDF to have a max value of 1. The identity discount
                 function uses the posterior probability directly as the discount weight. Default
                 value is "identity".

alpha_max        scalar. Maximum weight the discount function can apply. Default is 1. For a
                 two-arm trial, users may specify a vector of two values where the first value is
                 used to weight the historical treatment group and the second value is used to
                 weight the historical control group.

fix_alpha        logical. Fix alpha at alpha_max? Default value is FALSE.

number_mcmc      scalar. Number of Monte Carlo simulations. Default is 10000.

weibull_scale    scalar. Scale parameter of the Weibull discount function used to compute alpha,
                 the weight parameter of the historical data. Default value is 0.135. For a two-
                 arm trial, users may specify a vector of two values where the first value is used
                 to estimate the weight of the historical treatment group and the second value is
                 used to estimate the weight of the historical control group.

weibull_shape    scalar. Shape parameter of the Weibull discount function used to compute alpha,
                 the weight parameter of the historical data. Default value is 3. For a two-arm
                 trial, users may specify a vector of two values where the first value is used to
                 estimate the weight of the historical treatment group and the second value is
                 used to estimate the weight of the historical control group.

method           character. Analysis method with respect to estimation of the weight paramter
                 alpha. Default method "mc" estimates alpha for each Monte Carlo iteration.
                 Alternate value "fixed" estimates alpha once and holds it fixed throughout the
                 analysis. See the the bdpsurvival vignette
                 vignette("bdpsurvival-vignette",package="bayesDP") for more details.

compare          logical. Should a comparison object be included in the fit? For a one-arm anal-
                 ysis, the comparison object is simply the posterior chain of the treatment group
                 parameter. For a two-arm analysis, the comparison object is the posterior chain
                 of the treatment effect that compares treatment and control. If compare=TRUE,
                 the comparison object is accessible in the final slot, else the final slot is NULL.
                 Default is TRUE.

## Details

bdpsurvival uses a two-stage approach for determining the strength of historical data in estimation
of a survival probability outcome. In the first stage, a *discount function* is used that that defines the
maximum strength of the historical data and discounts based on disagreement with the current
data. Disagreement between current and historical data is determined by stochastically comparing
the respective posterior distributions under noninformative priors. With a single arm survival data
analysis, the comparison is the probability (p) that the current survival is less than the historical
survival. For a two-arm survival data, analysis the comparison is the probability that the hazard

ratio comparing treatment and control is different from zero. The comparison metric p is then input into the discount function and the final strength of the historical data is returned (alpha).

In the second stage, posterior estimation is performed where the discount function parameter, `alpha`, is used incorporated in all posterior estimation procedures.

To carry out a single arm (OPC) analysis, data for the current and historical treatments are specified in separate data frames, data and data0, respectively. The data frames must have matching columns denoting time and status. The 'time' column is the survival (censor) time of the event and the 'status' column is the event indicator. The results are then based on the posterior probability of survival at `surv_time` for the current data augmented by the historical data.

Two-arm (RCT) analyses are specified similarly to a single arm trial. Again the input data frames must have columns denoting time and status, but now an additional column named 'treatment' is required to denote treatment and control data. The 'treatment' column must use 0 to indicate the control group. The current data are augmented by historical data (if present) and the results are then based on the posterior distribution of the hazard ratio between the treatment and control groups.

For more details, see the bdpsurvival vignette:
`vignette("bdpsurvival-vignette",package="bayesDP")`

## Value

bdpsurvival returns an object of class "bdpsurvival". The functions summary and print are used to obtain and print a summary of the results, including user inputs. The plot function displays visual outputs as well.

An object of class "bdpsurvival" is a list containing at least the following components:

posterior_treatment  list. Entries contain values related to the treatment group:

- `alpha_discount` numeric. Alpha value, the weighting parameter of the historical data.
- `p_hat` numeric. The posterior probability of the stochastic comparison between the current and historical data.
- `posterior_survival` vector. If one-arm trial, a vector of length `number_mcmc` containing the posterior probability draws of survival at `surv_time`.
- `posterior_flat_survival` vector. If one-arm trial, a vector of length `number_mcmc` containing the probability draws of survival at `surv_time` for the current treatment not augmented by historical treatment.
- `prior_survival` vector. If one-arm trial, a vector of length `number_mcmc` containing the probability draws of survival at `surv_time` for the historical treatment.
- `posterior_hazard` matrix. A matrix with `number_mcmc` rows and `length(breaks)` columns containing the posterior draws of the piecewise hazards for each interval break point.
- `posterior_flat_hazard` matrix. A matrix with `number_mcmc` rows and `length(breaks)` columns containing the draws of piecewise hazards for each interval break point for current treatment not augmented by historical treatment.
- `prior_hazard` matrix. A matrix with `number_mcmc` rows and `length(breaks)` columns containing the draws of piecewise hazards for each interval break point for historical treatment.

posterior_control  list. If two-arm trial, contains values related to the control group analagous to the `posterior_treatment` output.

final  list. Contains the final comparison object, dependent on the analysis type:

- One-arm analysis: vector. Posterior chain of survival probability at requested time.
- Two-arm analysis: vector. Posterior chain of log-hazard rate comparing treatment and control groups.

args1 list. Entries contain user inputs. In addition, the following elements are ouput:

- S_t, S_c, S0_t, S0_c survival objects. Used internally to pass survival data between functions.
- arm2 logical. Used internally to indicate one-arm or two-arm analysis.

### References

Haddad, T., Himes, A., Thompson, L., Irony, T., Nair, R. MDIC Computer Modeling and Simulation working group.(2017) Incorporation of stochastic engineering models as prior information in Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 1-15.

### See Also

summary, print, and plot for details of each of the supported methods.

### Examples

```
# One-arm trial (OPC) example - survival probability at 5 years

# Collect data into data frames
df_ <- data.frame(status = rexp(50, rate=1/30),
                  time   = rexp(50, rate=1/20))
df_$status <- ifelse(df_$time < df_$status, 1, 0)

df0 <- data.frame(status = rexp(50, rate=1/30),
                  time   = rexp(50, rate=1/10))
df0$status <- ifelse(df0$time < df0$status, 1, 0)


fit1 <- bdpsurvival(Surv(time, status) ~ 1,
                    data  = df_,
                    data0 = df0,
                    surv_time = 5,
                    method = "fixed")

print(fit1)
## Not run:
plot(fit1)

## End(Not run)

# Two-arm trial example
# Collect data into data frames
df_ <- data.frame(time = c(rexp(50, rate=1/20),  # Current treatment
                           rexp(50, rate=1/10)), # Current control
                  status = rexp(100, rate=1/40),
```

```
                                treatment = c(rep(1,50), rep(0,50)))
df_$status <- ifelse(df_$time < df_$status, 1, 0)

df0 <- data.frame(time = c(rexp(50, rate=1/30),  # Historical treatment
                            rexp(50, rate=1/5)),  # Historical control
                  status =  rexp(100, rate=1/40),
                  treatment = c(rep(1,50), rep(0,50)))
df0$status <- ifelse(df0$time < df0$status, 1, 0)

fit2 <- bdpsurvival(Surv(time, status) ~ treatment,
                    data = df_,
                    data0 = df0,
                    method = "fixed")
summary(fit2)

### Fix alpha at 1
fit2_1 <- bdpsurvival(Surv(time, status) ~ treatment,
                      data = df_,
                      data0 = df0,
                      fix_alpha = TRUE,
                      method = "fixed")
summary(fit2_1)
```

---

```
plot,bdpbinomial-method
```
*bdpbinomial Object Plot*

---

### Description

plot method for class bdpbinomial.

### Usage

```
## S4 method for signature 'bdpbinomial'
plot(x, type = NULL, print = TRUE)
```

### Arguments

| | |
|---|---|
| x | object of class bdpbinomial. The result of a call to the [bdpbinomial](#) function. |
| type | character. Optional. Select plot type to print. Supports the following: "discount" gives the discount function; "posteriors" gives the posterior plots of the event rates; and "density" gives the augmented posterior density plot(s). Leave NULL to display all plots in sequence. |
| print | logical. Optional. Produce a plot (print = TRUE; default) or return a ggplot object (print = FALSE). When print = FALSE, it is possible to use ggplot2 syntax to modify the plot appearance. |

**Details**

The `plot` method for bdpbinomial returns up to three plots: (1) posterior density curves; (2) posterior density of the effect of interest; and (3) the discount function. A call to `plot` that omits the `type` input returns one plot at a time and prompts the user to click the plot or press return to proceed to the next plot. Otherwise, the user can specify a plot `type` to display the requested plot.

---

plot,bdpnormal-method    *bdpnormal Object Plot*

---

**Description**

`plot` method for class bdpnormal.

**Usage**

```
## S4 method for signature 'bdpnormal'
plot(x, type = NULL, print = TRUE)
```

**Arguments**

| | |
|---|---|
| x | object of class bdpnormal. The result of a call to the [bdpnormal](#) function. |
| type | character. Optional. Select plot type to print. Supports the following: "discount" gives the discount function; "posteriors" gives the posterior plots of the event rates; and "density" gives the augmented posterior density plot(s). Leave NULL to display all plots in sequence. |
| print | logical. Optional. Produce a plot (`print` = TRUE; default) or return a ggplot object (`print` = FALSE). When `print` = FALSE, it is possible to use ggplot2 syntax to modify the plot appearance. |

**Details**

The `plot` method for bdpnormal returns up to three plots: (1) posterior density curves; (2) posterior density of the effect of interest; and (3) the discount function. A call to `plot` that omits the `type` input returns one plot at a time and prompts the user to click the plot or press return to proceed to the next plot. Otherwise, the user can specify a plot `type` to display the requested plot.

---

```
plot,bdpsurvival-method
```
*bdpsurvival Object Plot*

---

### Description

plot method for class bdpsurvival.

### Usage

```
## S4 method for signature 'bdpsurvival'
plot(x, type = NULL, print = TRUE)
```

### Arguments

| | |
|---|---|
| x | object of class bdpsurvival. The result of a call to the [bdpsurvival](#) function. |
| type | character. Optional. Select plot type to print. Supports the following: "discount" gives the discount function and "survival" gives the survival curves. Leave NULL to display all plots in sequence. |
| print | logical. Optional. Produce a plot (print = TRUE; default) or return a ggplot object (print = FALSE). When print = FALSE, it is possible to use ggplot2 syntax to modify the plot appearance. |

### Details

The plot method for bdpsurvival returns up to two plots: (1) posterior survival curves and (2) the discount function. A call to plot that omits the type input returns one plot at a time and prompts the user to click the plot or press return to proceed to the next plot. Otherwise, the user can specify a plot type to display the requested plot.

---

ppexp                           *Compute cdf of the piecewise exponential distribution*

---

### Description

ppexp is used to compute the cumulative distribution function of the piecewise exponential distribution. The function is ported from R to C++ via code adapted from the msm package.

### Usage

```
ppexp(q, x, cuts)
```

## Arguments

| | |
|---|---|
| q | scalar. The time point at which the cdf is to be computed. |
| x | vector or matrix. The hazard rate(s). |
| cuts | vector. Times at which the rate changes, i.e., the interval cut points. The first element of cuts should be 0, and cuts should be in increasing order. Must be the same length as x (vector) or have the same number of columns as x (matrix) |

## Details

The code underlying ppexp is written in C++ and adapted from R code used in the msm package.

## Value

The cumulative distribution function computed at time q, hazard(s) x, and cut points cuts.

## Examples

```
# Single vector of hazard rates. Returns a single cdf value.
q    <- 12
x    <- c(0.25,0.3,0.35,0.4)
cuts <- c(0,6,12,18)
pp   <- ppexp(q,x,cuts)

# Matrix of multiple vectors of hazard rates. Returns 10 cdf values.
x  <- matrix(rgamma(4*10, 0.1, 0.1), nrow=10)
pp <- ppexp(q,x,cuts)
```

---

print,bdpbinomial-method
                              *bdpbinomial Object Print*

---

## Description

print method for class bdpbinomial.

## Usage

```
## S4 method for signature 'bdpbinomial'
print(x)
```

## Arguments

| | |
|---|---|
| x | object of class bdpbinomial. The result of a call to the bdpbinomial function. |

## Details

Returns same output as a call to summary.

---

`print,bdplm-method`     *bdplm Object Print*

---

### Description

`print` method for class `bdplm`.

### Usage

```
## S4 method for signature 'bdplm'
print(x)
```

### Arguments

x                object of class bdplm. The result of a call to the [bdplm](#) function.

### Details

Displays a print of the `bdplm` fit and the initial function call. The fit includes the estimate of the intercept, treatment effect, and covariate effects. The discount function weight estimates are displayed as well. If method="mc", the median estimate of alpha is displayed.

---

`print,bdpnormal-method`

                *bdpnormal Object Print*

---

### Description

`print` method for class `bdpnormal`.

### Usage

```
## S4 method for signature 'bdpnormal'
print(x)
```

### Arguments

x                object of class bdpnormal. The result of a call to the [bdpnormal](#) function.

### Details

Returns same output as a call to [summary](#).

---

print,bdpsurvival-method
                            *bdpsurvival Object Print*

---

### Description

print method for class bdpsurvival.

### Usage

```
## S4 method for signature 'bdpsurvival'
print(x)
```

### Arguments

x                    object of class bdpsurvival. The result of a call to the [bdpsurvival](#) function.

### Details

Displays a print of the bdpsurvival fit. The output is different, conditional on a one- or two-arm
survival analysis.

In the case of a one-arm analysis, a brief summary is displayed, including the current data sample
size, number of events, user input survival time, the augmented median survival probability, and
corresponding lower and upper 95 percent interval limits.

When a control arm is present, the output is the same as a call to [summary](#).

---

probability_discount     *Bayesian Discount Prior: Comparison Between Current and Histori-
                         cal Data*

---

### Description

probability_discount can be used to estimate the posterior probability of the comparison be-
tween historical and current data in the context of a clinical trial with normal (mean) data. probability_discount
is not used internally but is given for educational purposes.

### Usage

```
probability_discount(
  mu = NULL,
  sigma = NULL,
  N = NULL,
  mu0 = NULL,
  sigma0 = NULL,
  N0 = NULL,
```

```
  number_mcmc = 10000,
  method = "fixed"
)
```

## Arguments

| | |
|---|---|
| mu | scalar. Mean of the current data. |
| sigma | scalar. Standard deviation of the current data. |
| N | scalar. Number of observations of the current data. |
| mu0 | scalar. Mean of the historical data. |
| sigma0 | scalar. Standard deviation of the historical data. |
| N0 | scalar. Number of observations of the historical data. |
| number_mcmc | scalar. Number of Monte Carlo simulations. Default is 10000. |
| method | character. Analysis method. Default value "fixed" estimates the posterior probability and holds it fixed. Alternative method "mc" estimates the posterior probability for each Monte Carlo iteration. See the the bdpnormal vignette vignette("bdpnormal-vignette",package="bayesDP") for more details. |

## Details

This function is not used internally but is given for educational purposes. Given the inputs, the output is the posterior probability of the comparison between current and historical data in the context of a clinical trial with normal (mean) data.

## Value

probability_discount returns an object of class "probability_discount".

An object of class probability_discount contains the following:

p_hat scalar. The posterior probability of the comparison historical data weight. If method="mc", a vector of posterior probabilities of length number_mcmc is returned.

## References

Haddad, T., Himes, A., Thompson, L., Irony, T., Nair, R. MDIC Computer Modeling and Simulation working group.(2017) Incorporation of stochastic engineering models as prior information in Bayesian medical device trials. *Journal of Biopharmaceutical Statistics*, 1-15.

## Examples

```
probability_discount(mu  = 0,   sigma = 1, N  = 100,
                     mu0 = 0.1, sigma0 = 1, N0 = 100)
```

---

summary,bdpbinomial-method

*bdpbinomial Object Summary*

---

### Description

summary method for class bdpbinomial.

### Usage

```
## S4 method for signature 'bdpbinomial'
summary(object)
```

### Arguments

object          object of class bdpbinomial. The result of a call to the [bdpbinomial](#) function.

### Details

Displays a summary of the bdpbinomial fit including the input data, the stochastic comparison between current and historical data, and the resulting historical data weight (alpha). If historical data is missing then no stochastic comparison nor weight are displayed.

In the case of a one-arm analysis, the displayed 95 percent CI contains the lower and upper limits of the augmented event rate of the current data. The displayed sample estimate is the event rate of the current data augmented by the historical data.

When a control arm is present, a two-arm analysis is carried out. Now, the displayed 95 percent CI contains the lower and upper limits of the difference between the treatment and control arms with the historical data augmented to current data, if present. The displayed sample estimates are the event rates of the treatment and control arms, each of which are augmented when historical data are present.

---

summary,bdplm-method     *bdplm Object Summary*

---

### Description

summary method for class bdplm.

### Usage

```
## S4 method for signature 'bdplm'
summary(object)
```

### Arguments

object          an object of class bdplm, a result of a call to [bdplm](#).

**Details**

Displays a summary of the bdplm fit. Displayed output is similar to summary.lm. The function call, residuals, and coefficient table are displayed. The discount function weight estimates are displayed as well. If method="mc", the median estimate of alpha is displayed.

---

summary,bdpnormal-method

*bdpnormal Object Summary*

---

**Description**

summary method for class bdpnormal.

**Usage**

```
## S4 method for signature 'bdpnormal'
summary(object)
```

**Arguments**

object          object of class bdpnormal. The result of a call to the bdpnormal function.

**Details**

Displays a summary of the bdpnormal fit including the input data, the stochastic comparison between current and historical data, and the resulting historical data weight (alpha). If historical data is missing then no stochastic comparison nor weight are displayed.

In the case of a one-arm analysis, the displayed 95 percent CI contains the lower and upper limits of the augmented mean value of the current data. The displayed mean of treatment group is the mean of the current data augmented by the historical data.

When a control arm is present, a two-arm analysis is carried out. Now, the displayed 95 percent CI contains the lower and upper limits of the difference between the treatment and control arms with the historical data augmented to current data, if present. The displayed posterior sample estimates are the mean of the treatment and control arms, each of which are augmented when historical data are present.

---

summary,bdpsurvival-method
*bdpsurvival Object Summary*

---

## Description

summary method for class bdpsurvival.

## Usage

```
## S4 method for signature 'bdpsurvival'
summary(object)
```

## Arguments

object          an object of class bdpsurvival, a result of a call to bdpsurvival.

## Details

Displays a summary of the bdpsurvival fit. The output is different, conditional on a one- or two-arm survival analysis.

In the case of a one-arm analysis, the stochastic comparison between current and historical data and the resulting historical data weight (alpha) are displayed, followed by a survival table containing augmented posterior estimates of the survival probability at each time point for the current data.

When a control arm is present, a two-arm analysis is carried out. A two-arm survival analysis is similar to a cox proportional hazards analysis, and the displayed summary reflects this. First, the stochastic comparison between current and historical data and the resulting historical data weight (alpha) are displayed, when historical data is present for the respective arm. The displayed coef value is the log-hazard between the augmented treatment and control arms (log(treatment) - log(control)). The lower and upper 95 percent interval limits correspond to the coef value.

# Index