Package ‘GNGTools’

December 15, 2022

Type Package

Title Tools for Go/No-Go Decision-Making Framework

Version 1.0.0

Date 2022-11-22

URL https://github.com/gcicc/GNGTools

Maintainer Greg Cicconetti <greg.cicconetti@gmail.com>

Description Go/No-Go Decision-making Framework based on Bayesian posterior probabilities linked to the target product profile.

License GPL-2

Encoding UTF-8

LazyLoad no

Depends R (>= 3.5.0)

Imports survival, ggplot2, scales, stringr, plyr, grid, gridExtra, gtable, reshape2, grDevices, utils, extraDistr, janitor, parallel, latex2exp, knitr,tictoc, stats, dplyr, tibble, tidyr, purrr, data.table, mvtnorm, rlang

VignetteBuilder knitr

RoxygenNote 7.2.2

NeedsCompilation no

Author Greg Cicconetti [aut, cre] (<https://orcid.org/0000-0002-5380-7516>)

Repository CRAN

Date/Publication 2022-12-15 10:50:05 UTC

R topics documented:

`df_to_clip` *Copy a data.frame to the clipboard*

Description

Copy a data.frame to the clipboard

Usage

```
df_to_clip(df, digits = 4)
```

Arguments

`df`	A data.frame
`digits`	number of digits for rounding numerical columns

Value

the data.frame is copied to the clipboard allowing for easy pasting into excel (and subsequently to ppt, etc.)

`Discord_Go` *Discord_Go*

Description

Discord_Go

Usage

```
Discord_Go(go, n, fgo, N, prob, Weights = NA)
```

Arguments

`go`	go
`n`	n
`fgo`	fgo
`N`	N
`prob`	prob
`Weights`	weights

Value

These functions are used by other functions

`Discord_Grey` *Discord_Grey*

Description

Discord_Grey

Usage

```
Discord_Grey(low, up, n, fgo, fnogo, N, prob, Weights = NA)
```

Arguments

`low`	low
`up`	up
`n`	n
`fgo`	fgo
`fnogo`	gnogo
`N`	N
`prob`	prob
`Weights`	weights

Value

These functions are used by other functions

`Discord_NoGo` *Discord_NoGo*

Description

Discord_NoGo

Usage

```
Discord_NoGo(nogo, n, fnogo, N, prob, Weights = NA)
```

Arguments

`nogo`	nogo
`n`	n
`fnogo`	fnogo
`N`	N
`prob`	prob
`Weights`	Weights

Value

These functions are used by other functions

`gcurve` *gcurve*

Description

Returns a data.frame associated with a call to base::curve.

Usage

```
gcurve(
  expr,
  from = NULL,
  to = NULL,
  n = 101,
  add = FALSE,
  type = "l",
  xname = "x",
  xlab = xname,
  ylab = NULL,
  log = NULL,
  xlim = NULL,
  category = NULL,
  ...
)
```

Arguments

`expr`	expression to be plotted
`from`	lower bound of x-values
`to`	upper bound of x-values
`n`	number of points to plot
`add`	logical; if TRUE add to an already existing plot; if NA start a new plot taking the defaults for the limits and log-scaling of the x-axis from the previous plot. Taken as FALSE (with a warning if a different value is supplied) if no graphics device is open.
`type`	plot type: see plot.default.
`xname`	character string giving the name to be used for the x axis
`xlab`	labels and graphical parameters can also be specified as arguments.
`ylab`	For the "function" method of plot, ... can include any of the other arguments of curve, except expr
`log`	For the "function" method of plot, ... can include any of the other arguments of curve, except expr
`xlim`	NULL or a numeric vector of length 2; if non-NULL it provides the defaults for c(from, to) and, unless add = TRUE, selects the x-limits of the plot – see plot.window.
`category`	optional text column appends to data.frame returned
`...`	additional items passed to curve

Value

A data.frame is returned with x and y-values and an optional column called category

Examples

```
my.gcurve <- gcurve(expr = dnorm(x, mean=0, sd=1),from=-4, to = 4, n= 1001,
category= "Standard Normal")
head(my.gcurve)
```

`geom_stepribbon` *geom_stepribbon* `geom_stepribbon` *is an extension of the* `geom_ribbon`*, and is optimized for Kaplan-Meier plots with pointwise confidence intervals or a confidence band.*

Description

Step ribbon plots.

Usage

```
geom_stepribbon(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  kmplot = FALSE,
  ...
)
```

Arguments

`mapping`	see source reference
`data`	see source reference
`stat`	see source reference
`position`	see source reference
`na.rm`	see source reference
`show.legend`	see source reference
`inherit.aes`	see source reference
`kmplot`	If `TRUE`, missing values are replaced by the previous values. This option is needed to make Kaplan-Meier plots if the last observation has event, in which case the upper and lower values of the last observation are missing. This processing is optimized for results from the survfit function.
`...`	see source reference

Value

geom_stepribbon is an analog to geom_ribbon using step function.

Aesthetics

@section Aesthetics: `geom_ribbon()`understands the following aesthetics (required aesthetics are in bold):

- `x`***or***`y`
- `ymin`***or***`xmin`
- `ymax`***or***`xmax`
- `alpha`
- `colour`
- `fill`
- `group`
- `linetype`
- `size`

Learn more about setting these aesthetics in `vignette("ggplot2-specs")`.

See Also

`geom_ribbon` `geom_stepribbon` inherits from `geom_ribbon`.

`get.ng.post` *Normal-Gamma Posterior Updating*

Description

Normal-Gamma Posterior Updating

Usage

```
get.ng.post(
  mu.0 = 0,
  n.0 = 10,
  alpha.0 = 0.25,
  beta.0 = 1,
  xbar = 0.25,
  s = 3,
  n = 15,
  group = "Control"
)
```

Arguments

`mu.0`	prior mean
`n.0`	prior effective sample size
`alpha.0`	prior alpha parameter
`beta.0`	prior beta parameter
`xbar`	observed sampled mean
`s`	observed sample standard deviation
`n`	sample size
`group`	text string for group label

Value

Returns a data.frame with prior, data, and posterior parameters.

Author(s)

Greg Cicconetti

Examples

```
my.ng.post <- get.ng.post(mu.0 = 0, n.0 = 10, alpha.0 = .25, beta.0 = 1,
 xbar = .25, s = 3, n = 15, group = "Control")
my.ng.post
```

`get.ng.post.df` *Normal-Gamma Posterior Updating data.frame*

Description

Normal-Gamma Posterior Updating data.frame

Usage

```
get.ng.post.df(
  mu.0 = 0,
  n.0 = 10,
  alpha.0 = 0.25,
  beta.0 = 1,
  xbar = 0.25,
  s = c(1, 2, 3),
  n = 15,
  group = "Control"
)
```

Arguments

`mu.0`	prior mean
`n.0`	prior effective sample size
`alpha.0`	prior alpha parameter
`beta.0`	prior beta parameter
`xbar`	observed sampled mean
`s`	observed sample standard deviation
`n`	sample size
`group`	text string for group label

Value

Returns a data.frame with prior, data, and posterior parameters.

Examples

```
my.ng.post.df <- get.ng.post.df(mu.0 = 0, n.0 = 10, alpha.0 = .25, beta.0 = 1,
xbar = .25, s = c(1,2,3), n = 15, group = "Control")
my.ng.post.df
```

`get.normal.ss.post` *Get normal single sample posterior parameters*

Description

Get normal single sample posterior parameters

Usage

```
get.normal.ss.post(
  mu.0 = 0,
  n.0 = 10,
  alpha.0 = 0.25,
  beta.0 = 1,
  xbar = 4,
  s = 3,
  n = 15,
  group = "Placebo"
)
```

Arguments

`mu.0`	prior mean
`n.0`	prior effective sample size
`alpha.0`	prior alpha parameter
`beta.0`	prior beta parameter
`xbar`	observed sampled mean
`s`	observed sample standard deviation
`n`	sample size
`group`	text string for group label

Value

Returns normal single sample posterior parameters

Author(s)

Greg Cicconetti

Examples

```
my.normal.ss.post <- get.normal.ss.post(mu.0=0, n.0=10, alpha.0=.25,
beta.0=1, xbar=4, s=3, n=15, group=”Placebo”)
my.normal.ss.post
```

`get.ss.bin.df` *Get single sample binary data.frame*

Description

Get Go/No-go/Continue result

Usage

```
get.ss.bin.df(
  a.trt = seq(0.5, 1, 0.5),
  b.trt = seq(0.5, 1, 0.5),
  beta.mean = seq(0.3, 0.7, 0.01),
  eff.ss = 1:40,
  x.trt = 0:80,
  n.trt = c(40:80),
  Delta.tv = 0.4,
  Delta.lrv = 0.3,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  rp = FALSE
)
```

Arguments

`a.trt`	beta prior hyperparameter
`b.trt`	beta prior hyperparameter
`beta.mean`	mean of beta prior
`eff.ss`	effective sample size of beta prior
`x.trt`	sample responses
`n.trt`	sample size
`Delta.tv`	Base TPP
`Delta.lrv`	Min TPP
`tau.tv`	Base TPP threshold
`tau.lrv`	Min TPP threshold
`tau.ng`	No-Go threshold
`rp`	logical for reparameterized beta

Value

returns a data.frame with GO/No-Go probabilities and decisions

Examples

```
my.ss.bin.df <- get.ss.bin.df()
head(my.ss.bin.df)
```

`get.ss.bin.int.df` *Get single sample binary interim data.frame*

Description

get.ss.bin.interim.df

Usage

```
get.ss.bin.int.df(
  ss.bin.studyend.GNG = get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 50, x.trt =
    9, Delta.lrv = 0.62, Delta.tv = 0.7, tau.tv = 0.1, tau.lrv = 0.8, tau.ng = 0.65),
   goThreshold = 0.8,
   nogoThreshold = 0.8,
   include_nogo = FALSE
)
```

Arguments

`ss.bin.studyend.GNG`	The output from get.ss.bin.studyend.GNG
`goThreshold`	predictive posterior probability of a final Go threshold for an interim Go
`nogoThreshold`	predictive posterior probability of a final No Go threshold for an interim NoGo
`include_nogo`	logical

Value

A dataframe holding interim pred prob of Go and No-Go along with interim decision

---

`get.ss.bin.int.GNG` *Get single sample binary interim Go-NoGo*

Description

Get single sample binary interim Go-NoGo

Usage

```
get.ss.bin.int.GNG(ss.bin.int.df, Interims = 20, ss.bin.studyend.GNG)
```

Arguments

`ss.bin.int.df`	output from get.ss.bin.interim.df
`Interims`	vector of sample sizes at which interims will occur
`ss.bin.studyend.GNG`	output from call to get.ss.bin.studyend.GNG

Value

A tibble is returned holding the min and max needed for Go and No-Go resp.

Examples

```
{
my.ss.bin.int.df <- get.ss.bin.int.df(ss.bin.studyend.GNG =
get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
Delta.lrv = .2, Delta.tv = .35, tau.tv = 0.10, tau.lrv = .80, tau.ng = .65),
goThreshold = .8, nogoThreshold = 1.2, include_nogo = TRUE)
my.ss.bin.studyend.GNG <-  get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
                                                Delta.lrv = .2, Delta.tv = .35,
                                           tau.tv = 0.10, tau.lrv = .80, tau.ng = .65)
   my.ss.bin.int.GNG <- get.ss.bin.int.GNG(
   ss.bin.int.df = my.ss.bin.int.df,
   Interims = 20,
   ss.bin.studyend.GNG = my.ss.bin.studyend.GNG)
 }
```

`get.ss.bin.int.oc` *Get single sample binary interim operating characteristics simulation*

Description

Get single sample binary interim operating characteristics simulation

Usage

```
get.ss.bin.int.oc(
  ss.bin.int.df,
  ss.bin.int.GNG,
  lower = 0,
  upper = 1,
  step = 0.025,
  include_nogo = TRUE
)
```

Arguments

`ss.bin.int.df`	call to get.ss.bin.interim.df
`ss.bin.int.GNG`	call to get.ss.bin.interim.GNG
`lower`	lower bound
`upper`	upper bound
`step`	stepsize
`include_nogo`	logical

Value

Returns a data.frame holding results at interim and final

Examples

```
my.ss.bin.int.df = get.ss.bin.int.df(ss.bin.studyend.GNG =
get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
Delta.lrv = .2, Delta.tv = .35,
tau.tv = 0.10, tau.lrv = .80, tau.ng = .65),
goThreshold = .8,
nogoThreshold = 1.2,
include_nogo = TRUE)
my.ss.bin.studyend.GNG <- get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
                                                  Delta.lrv = .2, Delta.tv = .35,
     tau.tv = 0.10, tau.lrv = .80, tau.ng = .65)
my.ss.bin.int.GNG <- get.ss.bin.int.GNG( ss.bin.int.df = my.ss.bin.int.df,
                                        Interims = 20,
ss.bin.studyend.GNG = my.ss.bin.studyend.GNG)
```

```
my.ss.bin.int.GNG
```

---

get.ss.bin.int.oc.sim *Get single sample binary interim operating charactersitics simulations*

Description

Get single sample binary interim operating charactersitics simulations

Usage

```
get.ss.bin.int.oc.sim(InterimGNG, TrueRate, runs = 500)
```

Arguments

`InterimGNG`	GNG decision value outputs from get.ss.bin.int.GNG
`TrueRate`	assumed true response rate. Can be a vector
`runs`	number of simulation runs

Value

Returns results of a simulation.

Examples

```
my.ss.bin.int.df <- get.ss.bin.int.df(ss.bin.studyend.GNG =
get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
                        Delta.lrv = .2, Delta.tv = .35,
                        tau.tv = 0.10, tau.lrv = .80, tau.ng = .65),
                   goThreshold = .8,
                   nogoThreshold = 1.2,
                   include_nogo = TRUE)
my.ss.bin.studyend.GNG <- get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
                                                  Delta.lrv = .2, Delta.tv = .35,
                                              tau.tv = 0.10, tau.lrv = .80, tau.ng = .65)
my.ss.bin.int.GNG <- get.ss.bin.int.GNG( ss.bin.int.df = my.ss.bin.int.df,
                                         Interims = 20,
                                         ss.bin.studyend.GNG = my.ss.bin.studyend.GNG)
my.ss.bin.int.GNG
```

`get.ss.bin.ssize.oc.df` *Get single sample binary sample size operating charactersitics data.frame*

Description

Get single sample binary sample size operating charactersitics data.frame

Usage

```
get.ss.bin.ssize.oc.df(
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  Delta.lrv = 0.35,
  Delta.tv = 0.35,
  Delta.user = 0.4,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  SS.OC.N.LB = 20,
  SS.OC.N.UB = 60
)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`n.trt`	observed sample size
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.user`	User-specified rate
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`SS.OC.N.LB`	Sample size OC Curve's Lower sample size bound
`SS.OC.N.UB`	Sample size OC Curve's Upper sample size bound

Value

A data.frame is returned

Examples

```
my.ss.bin.ssize.oc.df <- get.ss.bin.ssize.oc.df()
head(my.ss.bin.ssize.oc.df)
```

---

get.ss.bin.studyend.GNG

Get single sample binary study end GNG criteria

Description

Get single sample binary study end GNG criteria

Usage

```
get.ss.bin.studyend.GNG(
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  x.trt = 9,
  Delta.lrv = 0.2,
  Delta.tv = 0.35,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`n.trt`	observed sample size
`x.trt`	observed number of responders
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

Returns a list of data.frames holding what is needed from data to acheive Go/No-Go

Examples

```
get.ss.bin.studyend.GNG()
```

`get.ss.bin.trt.int.oc.df` *Get single sample binary interim treatment OC data.frame*

Description

Get single sample binary interim treatment OC data.frame

Usage

```
get.ss.bin.trt.int.oc.df(
  a.trt = 1,
  b.trt = 1,
  Delta.tv = 0.35,
  Delta.lrv = 0.2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  interim.n.t = c(10),
  final.n.t = 100,
  x.ng = NULL,
  x.go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`interim.n.t`	number of trials at interim
`final.n.t`	number of trials at final
`x.ng`	responses needed for no-go; leave null for standard rule
`x.go`	responses needed for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability
`ng.thresh`	no-go threshold for predictive probability

Value

A data.frame is returned

Examples

```
my.ss.bin.trt.int.oc.df <- get.ss.bin.trt.int.oc.df()
my.ss.bin.trt.int.oc.df[[1]]
my.ss.bin.trt.int.oc.df[[2]]
```

`get.ss.bin.trt.oc.df` *Get single sample binary treatment OC data.frame*

Description

Get single sample binary treatment OC data.frame

Usage

```
get.ss.bin.trt.oc.df(
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  Delta.tv = 0.35,
  Delta.lrv = 0.2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`n.trt`	observed sample size
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data.frame is returned.

Examples

```
my.ss.bin.trt.oc.df <- get.ss.bin.trt.oc.df()
my.ss.bin.trt.oc.df
```

`get.ss.ng.df` *Get single sample normal-gamma data.frame with decision output*

Description

Get single sample normal-gamma data.frame with decision output

Usage

```
get.ss.ng.df(
  mu.0.t = 0,
  n.0.t = 10,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = seq(-1, 5, 0.1),
  s.t = seq(1, 6, 0.1),
  n.t = 50,
  Delta.tv = 1.75,
  Delta.lrv = 1.5,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data.frame is returned

Examples

```
my.ss.ng.df <- get.ss.ng.df()
head(my.ss.ng.df)
```

`get.ss.ng.ssize.oc.df` *Get single sample normal-gamma sample size Oc curve data.frame*

Description

Get single sample normal-gamma sample size Oc curve data.frame

Usage

```
get.ss.ng.ssize.oc.df(
  mu.0.t = 3,
  n.0.t = 10,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  s.t = 5,
  n.t = 50,
  SS.OC.N.LB = floor(50 * 0.75),
  SS.OC.N.UB = floor(50 * 2),
  npoints = 15,
  Delta.lrv = 2.5,
  Delta.tv = 4,
  Delta.user = 3,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group

`SS.OC.N.LB`	lower bound for OC curve
`SS.OC.N.UB`	upper bound for OC curve
`npoints`	number of points
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.user`	User's value for underlying treatment response
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data.frame is returned

Examples

```
my.ss.ng.ssize.oc.df <- get.ss.ng.ssize.oc.df()
head(my.ss.ng.ssize.oc.df)
```

`get.ss.ng.studyend.GNG` *Get single sample normal-gamma study-end GNG*

Description

Get single sample normal-gamma study end GNG

Usage

```
get.ss.ng.studyend.GNG(
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  n.0.t = 10,
  xbar.t = 1.97,
  s.t = 2,
  n.t = 20,
  Delta.lrv = 1.25,
  Delta.tv = 1.75,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`mu.0.t`	prior mean for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`n.0.t`	prior effective sample size for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

returns a list of data.frames holding what is needed from data for study-end Go/No-Go

Examples

```
my.ss.mg.studyend.GNG <- get.ss.ng.studyend.GNG(mu.0.t = 0, alpha.0.t=.25, beta.0.t = 1,
n.0.t = 10, xbar.t = 1.97, s.t = 2, n.t = 20, Delta.lrv = 1.25, Delta.tv = 1.75,
tau.tv=.1, tau.lrv=.8, tau.ng=.65)
my.ss.mg.studyend.GNG
```

`get.ss.ng.trt.int.oc.df` *Get single sample normal-gamma interim operating characteristic data.frame*

Description

Get single sample normal-gamma interim operating characteristic data.frame

Usage

```
get.ss.ng.trt.int.oc.df(
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  Delta.lrv = 1.25,
  Delta.tv = 1.75,
  Delta.OC.LB = -5,
```

```
  Delta.OC.UB = 5,
  npoints = 10,
  n.MC = 1000,
  s.t = 2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  interim.n.t = c(10, 20, 30),
  final.n.t = 40,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  include_nogo = TRUE
)
```

Arguments

`mu.0.t`	prior mean
`n.0.t`	prior effective sample size
`alpha.0.t`	prior alpha parameter
`beta.0.t`	prior beta parameter
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.OC.LB`	OC Lower bound
`Delta.OC.UB`	OC Uppder bound
`npoints`	number of points to span
`n.MC`	number of trials at each point
`s.t`	standard deviation
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`interim.n.t`	interim sample sizes
`final.n.t`	final sample size
`go.thresh`	interim go threshold
`ng.thresh`	interim no-go threshold
`include_nogo`	logical

Value

A data.frame is returned.

Examples

```
holdit <- get.ss.ng.trt.int.oc.df(n.MC=100, npoints=3)
head(holdit)
```

`get.ss.ng.trt.oc.df` *Get single sample normal-gamma treament OC data.frame*

Description

Get single sample normal-gamma treament OC data.frame

Usage

```
get.ss.ng.trt.oc.df(
  mu.0.t = 0,
  n.0.t = 10,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  s.t = 2,
  n.t = 40,
  from.here = 0,
  to.here = 4,
  length.out = 1000,
  Delta.tv = 1.75,
  Delta.lrv = 1,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`mu.0.t`	prior mean
`n.0.t`	prior effective sample size
`alpha.0.t`	prior alpha parameter
`beta.0.t`	prior beta parameter
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group
`from.here`	treatment effect lower bound
`to.here`	treatment effect upper bound
`length.out`	number of points used to span (from.here, to.here)
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

Returns a data.frame ready to create a treatment effect OC curve

Author(s)

Greg Cicconetti

Examples

```
my.ss.ng.trt.oc.df <- get.ss.ng.trt.oc.df()
head(my.ss.ng.trt.oc.df)
```

`get.ss.normal.post` *Get single sample normal (with known variance) posterior distribution parameters*

Description

Returns the parameters of the posterior distribution in the case of normal prior with known variance and normal data

Usage

```
get.ss.normal.post(
  prior.mean = 0,
  prior.var = 1e+06,
  sample.n = 10,
  sample.xbar = seq(-1, 1, 0.01),
  sample.var = 1
)
```

Arguments

`prior.mean`	prior mean
`prior.var`	prior variance
`sample.n`	sample size
`sample.xbar`	sample mean
`sample.var`	sample variance

Value

returns a data.frame hold posterior parameters, sample x.bar, sample variance and posterior mean and variances

Author(s)

Greg Cicconetti

Examples

```
my.ss.normal.post <- get.ss.normal.post()
head(my.ss.normal.post)
```

`get.ts.bin.dec` *Get two sample binary decision*

Description

Get two sample binary decision

Usage

```
get.ts.bin.dec(
  a.con = 1,
  b.con = 1,
  n.con = 40,
  x.con = 5,
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  x.trt = 20,
  Delta.tv = 0.25,
  Delta.lrv = 0.2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`n.con`	observed sample size for control group
`x.con`	observed number of responders for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`n.trt`	observed sample size for treatment group
`x.trt`	observed number of responders for treatment group
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

returns a data.frame holding Posterior probabilties of interest and Go/No-Go result

Author(s)

Greg Cicconetti

Examples

```
get.ts.bin.dec()
```

`get.ts.bin.dec.df` *Get Two-sample binary decision data.frame*

Description

Get Two-sample binary decision data.frame

Usage

```
get.ts.bin.dec.df(
  a.con = 1,
  b.con = 1,
  n.con = 40,
  x.con = 0:40,
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  x.trt = 0:40,
  Delta.tv = 0.25,
  Delta.lrv = 0.2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`n.con`	sample size for control
`x.con`	responders on control
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`n.trt`	sample size for treatment

`x.trt`	responders for treatment
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

a dataframe is returned

Examples

```
holdit <- get.ts.bin.dec.df()
head(holdit)
```

`get.ts.bin.int.dec` *Get two-sample binary interim decision*

Description

Get two-sample binary interim decision

Usage

```
get.ts.bin.int.dec(
  a.con = 1,
  b.con = 1,
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  n.con = 40,
  n.int.c = c(10, 20, 30),
  n.int.t = c(10, 20, 30),
  Delta.lrv = 0.15,
  Delta.tv = 0.3,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  runs = 500,
  include_nogo = TRUE
)
```

Arguments

`a.con`	alpha parameter for control
`b.con`	beta parameter for control
`a.trt`	alpha parameter for treatment
`b.trt`	beta parameter for treatment
`n.trt`	final sample size for treatment
`n.con`	final sample size for control
`n.int.c`	interim sample sizes for control
`n.int.t`	interim sample sizes for treatment
`Delta.lrv`	min TPP
`Delta.tv`	Base TPP
`tau.tv`	study-end threshold for Base TPP
`tau.lrv`	study-end threshold for Min TPP
`tau.ng`	study-end threshold for NG
`go.thresh`	threshold for predictive probabilities at interim
`ng.thresh`	threshold for predictive probabilities at interim
`runs`	the number of simulation runs
`include_nogo`	logical

Value

a Data frame with the following columns * **IntermR_C** the number of control responses * **IntermR_T** the number of treatment responses * **Interm** the index of which interim analysis is being assessed. * **Go/No-Go/Consider** the proportion of simulations generating a final go/No-Go/Consider decision. * **Decision** The interim decision based off of the simulated results.

Examples

```
my.ts.bin.int.dec <- get.ts.bin.int.dec()
my.ts.bin.int.dec
```

`get.ts.bin.int.oc` *Get two sample binary interim OC*

Description

Get two sample binary interim OC

Usage

```
get.ts.bin.int.oc(
  a.con = 1,
  b.con = 1,
  a.trt = 1,
  b.trt = 1,
  Delta.tv = 0.3,
  Delta.lrv = 0.2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  n.con = 40,
  n.trt = 40,
  n.int.c = c(10, 20, 30),
  n.int.t = c(10, 20, 30),
  DecisionTable = NULL,
  runs = 500,
  ControlRate = 0.2,
  TreatmentEffect = seq(0, 0.8, 0.1),
  include_nogo = TRUE
)
```

Arguments

`a.con`	alpha parameter for control
`b.con`	beta parameter for control
`a.trt`	alpha parameter for treatment
`b.trt`	beta parameter for treatment
`Delta.tv`	Base TPP
`Delta.lrv`	min TPP
`tau.tv`	study-end threshold for Base TPP
`tau.lrv`	study-end threshold for Min TPP
`tau.ng`	study-end threshold for NG
`go.thresh`	threshold for predictive probabilities at interim
`ng.thresh`	threshold for predictive probabilities at interim
`n.con`	final sample size for control
`n.trt`	final sample size for treatment
`n.int.c`	interim sample sizes for control
`n.int.t`	interim sample sizes for treatment
`DecisionTable`	results from get.ts.bin.int.dec
`runs`	number of simulation runs

`ControlRate`	assumed rate in the control group
`TreatmentEffect`	vector of treatment effects relative to control group.
`include_nogo`	logical

Value

dataframe of simulation results with the following columns: * **Effect** the difference between the control Rate and treatment rate. (TreatmentRate = ControlRate + Effect) * **run** the grouping variable by run. * **assessment** the assessment in order, the last assessment is the final analysis results. * **TreatmentEffect** The true response rate for the treatment group. (ControlRate is the same for all runs, it is not in the output, only the function input. We may want to change this for completeness, or include it in Param) * **TotalResponse** for control (_C) and treatment(_T) respectively the total number of responses observed at that particular assessment. This is the cumulative sum over the assessments. * **Decision** The decision at the assessment given the total response up to that point. Note that the decision is calculated differently depending the assessment, the last assessment uses final assessment calculation while the earlier assessments use the interim assessment calculations.

Examples

```
my.ts.bin.int.oc <-  get.ts.bin.int.oc()
head(my.ts.bin.int.oc)
```

`get.ts.bin.ssize.oc.df`

Get two-sample binary sample size OC data.frame

Description

Get two-sample binary sample size OC data.frame

Usage

```
get.ts.bin.ssize.oc.df(
  a.con = 1,
  b.con = 1,
  a.trt = 1,
  b.trt = 1,
  dcurve.con = 0.12,
  Aratio = 2,
  SS.OC.N.LB = 40,
  SS.OC.N.UB = 160,
  Delta.lrv = 0.2,
  Delta.tv = 0.25,
  SS.OC.Delta = 0.25,
  tau.tv = 0.1,
```

```
  tau.lrv = 0.8,
  tau.ng = 0.65,
  npoints = 3
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`dcurve.con`	underlying rate for control group
`Aratio`	randomization ratio
`SS.OC.N.LB`	lower bound for sample size
`SS.OC.N.UB`	upper bound for sample size
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`SS.OC.Delta`	user's treatment effect
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`npoints`	number of points

Value

returns a data.frame ready to create two-sample binary sample size operating characteristics curve

Author(s)

Greg Cicconetti

Examples

```
my.ts.bin.ssize.oc.df <- get.ts.bin.ssize.oc.df()
```

`get.ts.bin.studyend.GNG`

Get Two-sample binary study end GNG

Description

Get Two-sample binary study end GNG

Usage

```
get.ts.bin.studyend.GNG(
  a.con = 1,
  b.con = 1,
  n.con = 30,
  x.con = 18,
  a.trt = 1,
  b.trt = 1,
  n.trt = 30,
  x.trt = 14,
  Delta.lrv = 0.3,
  Delta.tv = 0.4,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`n.con`	number of trials for control group
`x.con`	number of responses for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`n.trt`	number of trials for treatment group
`x.trt`	number of responses for treatment group
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

returns a list holding data.frames alterting to what is needed from data to acheive go/no-go

Examples

```
get.ts.bin.studyend.GNG()
```

`get.ts.bin.trt.oc.df` *Get two-sample binary treatment OC data.frame*

Description

Get two-sample binary treatment OC data.frame

Usage

```
get.ts.bin.trt.oc.df(
  a.con = 1,
  b.con = 1,
  dcurve.con = 0.12,
  a.trt = 1,
  b.trt = 1,
  TE.OC.N = 80,
  Aratio = 2,
  TE.OC.Delta.LB = 0,
  TE.OC.Delta.UB = 1 - 0.12,
  Delta.tv = 0.35,
  Delta.lrv = 0.2,
  tau.tv = 0.01,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`dcurve.con`	Response rate for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`TE.OC.N`	total sample size for OC
`Aratio`	Randomization ratio
`TE.OC.Delta.LB`	Lower bound for OC curve
`TE.OC.Delta.UB`	Upper bound for OC curve

`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data.frame is returned that allows for creation of treatment effect operating characteristics curve

References

Sverdlov O, Ryeznik Y, Wu S. Exact Bayesian Inference Comparing Binomial Proportions, With Application to Proof-of-Concept Clinical Trials. Therapeutic Innovation & Regulatory Science. 2015;49(1):163-174. doi:10.1177/2168479014547420

Examples

```
my.ts.bin.trt.oc.df <- get.ts.bin.trt.oc.df()
head(my.ts.bin.trt.oc.df)
```

`get.ts.ng.mc` *Get two-sample normal gamma MC sampling*

Description

Get two-sample normal gamma MC sampling

Usage

```
get.ts.ng.mc(
  mu.0.c = 0,
  n.0.c = 1e-04,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  xbar.c = 0,
  s.c = 2.3,
  n.c = 20000,
  group.c = "Control",
  mu.0.t = 1.5,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = 1.5,
  s.t = 2.3,
```

```
  n.t = 20000,
  group.t = "Treatment",
  Delta.tv = 1.5,
  Delta.lrv = 1,
  tau.tv = 0,
  tau.lrv = 0.8,
  tau.ng = 0.75,
  seed = 1234,
  n.MC = 5000
)
```

Arguments

`mu.0.c`	prior mean for control group
`n.0.c`	prior effective sample size for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`xbar.c`	sample mean for control group
`s.c`	sample sd for control group
`n.c`	sample size for control group
`group.c`	group label for control group
`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group
`group.t`	group label for treatment group
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`seed`	random seed
`n.MC`	n for MC sampling

Value

A data.frame is returned

Examples

```
my.ts.ng.mc <- get.ts.ng.mc()
my.ts.ng.mc
```

`get.ts.ng.mc.df` *Get two-sample normal-gamma MC-based data.frame*

Description

Get two-sample normal-gamma MC-based data.frame

Usage

```
get.ts.ng.mc.df(
  mu.0.c = 0,
  n.0.c = 10,
  alpha.0.c = 0.25 * 4,
  beta.0.c = 1 * 4,
  xbar.c = seq(-3, 3, length.out = 20),
  s.c = 3,
  n.c = 25,
  group.c = "Control",
  mu.0.t = 0,
  n.0.t = 10,
  alpha.0.t = 0.25 * 4,
  beta.0.t = 1 * 4,
  xbar.t = seq(0, 6, length.out = 20),
  s.t = 2,
  n.t = 25,
  group.t = "Treatment",
  Delta.tv = 1.75,
  Delta.lrv = 1.5,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0,
  n.MC = 1000,
  seed = 1234,
  expand = TRUE
)
```

Arguments

`mu.0.c`	prior mean for control group
`n.0.c`	prior effective sample size for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`xbar.c`	sample mean for control group
`s.c`	sample sd for control group
`n.c`	sample size for control group

`group.c`	group label for control group
`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group
`group.t`	group label for treatment group
`Delta.tv`	TPP Target Value aka Base TPP
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`n.MC`	number of MC sampling
`seed`	random seed
`expand`	logical; if true expand.grid is employed; else data.frame is employed. Former provides all combinations

Value

A data.frame is returned

Examples

```
my.ts.ng.mc.df <- get.ts.ng.mc.df()
my.ts.ng.mc.df
```

`get.ts.ng.ssize.oc.df` *Get two sample normal-gamma sample size OC data.frame*

Description

Get two sample normal-gamma sample size OC data.frame

Usage

```
get.ts.ng.ssize.oc.df(
  mu.0.c = 0,
  n.0.c = 1e-04,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  s.c = 4,
  group.c = "Control",
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  s.t = 4,
  group.t = "Treatment",
  ARatio = 1,
  SS.OC.N.LB = 20,
  SS.OC.N.UB = 200,
  Delta.tv = 1.5,
  Delta.lrv = 1,
  Delta.user = 1.4,
  tau.tv = 0,
  tau.lrv = 0.8,
  tau.ng = 0.75,
  npoints = 1,
  n.MC = 500,
  seed = 1234,
  goparallel = FALSE,
  cl = cl
)
```

Arguments

`mu.0.c`	prior mean for control group
`n.0.c`	prior effective sample size for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`s.c`	sample sd for control group
`group.c`	group label for control group
`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`s.t`	sample sd for treatment group
`group.t`	group label for treatment group
`ARatio`	randomization ratio

`SS.OC.N.LB`	lower bound for OC curve
`SS.OC.N.UB`	Upper bound for OC Curve
`Delta.tv`	Base TPP
`Delta.lrv`	Min TPP
`Delta.user`	User's delta
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`npoints`	number of points
`n.MC`	n for MC sampling
`seed`	random seed
`goparallel`	logical to use parallel programming
`cl`	cluster
`N`	Total sample size

Value

A data.frame is returned

Examples

```
my.ts.ng.ssize.oc.df <- get.ts.ng.ssize.oc.df()
my.ts.ng.ssize.oc.df
```

`get.ts.ng.studyend.GNG` *Get two-sample normal gamma study end GNG*

Description

Get two-sample normal gamma study end GNG

Usage

```
get.ts.ng.studyend.GNG(
  mu.0.c = 0,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  n.0.c = 1,
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
```

```
    n.0.t = 1,
    xbar.c = 1.5,
    s.c = 4,
    n.c = 40,
    xbar.t = 26,
    s.t = 4,
    n.t = 40,
    Delta.lrv = 1,
    Delta.tv = 1.5,
    tau.ng = 0.65,
    tau.lrv = 0.8,
    tau.tv = 0.1,
    n.MC = 1000
  )
```

Arguments

`mu.0.c`	prior mean for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`n.0.c`	prior effective sample size for control group
`mu.0.t`	prior mean for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`n.0.t`	prior effective sample size for treatment group
`xbar.c`	sample mean for control group
`s.c`	sample sd for control group
`n.c`	sample size for control group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size of treatment group
`Delta.lrv`	Min TPP
`Delta.tv`	Base TPP
`tau.ng`	threshold associated with No-Go
`tau.lrv`	threshold associated with Min TPP
`tau.tv`	threshold associated with Base TPP
`n.MC`	n for MC sampling

Value

A list is returned holding what is needed of data to achieve Go/No-Go

Author(s)

Greg Cicconetti

Examples

```
my.ts.ng.studyend.GNG <- get.ts.ng.studyend.GNG()
my.ts.ng.studyend.GNG
```

`get.ts.ng.trt.int.oc.df`

Get two-sample normal-gamma interim OC curve data.frame

Description

Get two-sample normal-gamma interim OC curve data.frame

Usage

```
get.ts.ng.trt.int.oc.df(
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  mu.0.c = 0,
  n.0.c = 10,
  alpha.0.c = 2.5,
  beta.0.c = 10,
  Delta.lrv = 1.5,
  Delta.tv = 3,
  mu.c = 0.25,
  s.t = 1.5,
  s.c = 1.5,
  npointsLookup = 20,
  npoints = 20,
  n.MC.lookup = 500,
  n.MC = 500,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  n.int.t = c(27, 40),
  n.int.c = c(27, 40),
  final.n.t = 55,
  final.n.c = 55,
  go.thresh = 0.6,
  ng.thresh = 0.6,
  go.parallel = TRUE,
  cl = cl,
```

```
    seed = 1234,
    include_nogo = TRUE
  )
```

Arguments

`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size parameter for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`mu.0.c`	prior mean for control group
`n.0.c`	prior effective sample size parameter for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`mu.c`	assumed mean for control group
`s.t`	treatment standard deviation
`s.c`	control standard deviation
`npointsLookup`	number of points for lookup table
`npoints`	number of points to run simulations
`n.MC.lookup`	number of trials used for lookup table
`n.MC`	number of trials run at each point
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`n.int.t`	interim sample sizes for treatment arm
`n.int.c`	interim sample sizes for control arm
`final.n.t`	final sample size: treatment arm
`final.n.c`	final sample sizE: control arm
`go.thresh`	interim predictive probability threshold for go
`ng.thresh`	interim predictive probability threshold for no-go
`go.parallel`	logical for parallel processing
`cl`	cl
`seed`	random seed
`include_nogo`	logical

Value

A data.frame is returned.

Examples

```
my.ts.ng.trt.int.oc.df <- get.ts.ng.trt.int.oc.df(npointsLookup = 2, npoints=3, n.MC.lookup=5,
n.MC=5, go.parallel=FALSE)
```

`get.ts.ng.trt.oc.df` *Get two-sample normal-gamma treatment OC data.frame*

Description

Get two-sample normal-gamma treatment OC data.frame

Usage

```
get.ts.ng.trt.oc.df(
  mu.0.c = 0,
  n.0.c = 1e-04,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  xbar.c = 0,
  s.c = 4,
  group.c = "Control",
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = 5,
  s.t = 4,
  group.t = "Treatment",
  Delta.LB = 0,
  Delta.UB = 1.5,
  ARatio = 1,
  N = 50,
  Delta.tv = 1.5,
  Delta.lrv = 1,
  tau.tv = 0.1,
  tau.lrv = 0.65,
  tau.ng = 0,
  npoints = 2,
  n.MC = 500,
  seed = 1234,
  cl = cl,
  goparallel = TRUE
)
```

Arguments

`mu.0.c`	prior mean for control group
`n.0.c`	prior effective sample size for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`xbar.c`	sample mean for control group
`s.c`	sample sd for control group
`group.c`	label for control group
`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`group.t`	label for treatment group
`Delta.LB`	Lower bound for Delta
`Delta.UB`	upper bound for delta
`ARatio`	randomization ratio
`N`	total sample size
`Delta.tv`	Base TPP
`Delta.lrv`	Min TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`npoints`	number of points
`n.MC`	n for MC sampling
`seed`	random seed
`cl`	cluster
`goparallel`	a logical to indicate if parallel computing is employed

Value

A data.frame is returned

Author(s)

Greg Cicconetti

Examples

```
my.ts.ng.trt.oc.df <- get.ts.ng.trt.oc.df(goparallel=FALSE)
head(my.ts.ng.trt.oc.df)
```

`get.tte.df` *Get time to event data.frame*

Description

Get time to event data.frame

Usage

```
get.tte.df(
  m.con.prior = 50,
  m.trt.prior = 50,
  HR.prior = 0.845,
  HR.obs = seq(0.3, 1, 0.01),
  m.obs = seq(10, 200, 5),
  ARatio = 0.5,
  HR.tv = 0.8,
  HR.lrv = 0.9,
  tau.tv = 0.1,
  tau.lrv = 0.2,
  tau.ng = 0.35
)
```

Arguments

`m.con.prior`	number of prior events for control
`m.trt.prior`	number of prior events for treamtent
`HR.prior`	HR estimate
`HR.obs`	Observed HR
`m.obs`	Obeserved number of events
`ARatio`	Randomization ratio
`HR.tv`	Base TPP for HR
`HR.lrv`	Min TPP for HR
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data.frame is returned

Examples

```
my.tte.df <- get.tte.df()
head(my.tte.df)
```

`get.tte.int.oc.df` *Make time to event predictive probability OC curve*

Description

Make time to event predictive probability OC curve

Usage

```
get.tte.int.oc.df(
  m.con.prior = 50,
  m.trt.prior = 50,
  HR.prior = 0.845,
  ARatio = 1,
  HR.lower = 0.0025,
  HR.upper = 2,
  npoints = 20,
  interim.m = c(428, 750, 1000),
  final.m = 1500,
  HR.tv = 0.75,
  HR.lrv = 0.9,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  n.MC = 2000,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  include_nogo = TRUE
)
```

Arguments

`m.con.prior`	prior number of events for control group
`m.trt.prior`	prior number of events for treamtnet group
`HR.prior`	hazard ratio estimate
`ARatio`	Randomization ratio
`HR.lower`	lower bound for HR
`HR.upper`	upper bound for HR
`npoints`	number of points for OC Curve
`interim.m`	interim number of events
`final.m`	final number of events
`HR.tv`	Base TPP for HR
`HR.lrv`	Min TPP for HR
`tau.tv`	threshold associated with Base TPP

`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`n.MC`	Monte Carlo sample size
`go.thresh`	predictive probability go threshold
`ng.thresh`	predictive probability no-go threshold
`include_nogo`	logical

Value

A data.frame is returned.

Examples

```
my.tte.int.oc.df <- get.tte.int.oc.df()
my.tte.int.oc.df
```

`get.tte.post` *Get posterior parameters for two-sample time to event case using normal approximation to log(HR) - expand.grid*

Description

Returns the parameters of the posterior distribution in the case of normal approximation to hazard ratio

Usage

```
get.tte.post(
  m.con.prior = 10,
  m.trt.prior = 10,
  HR.prior = 0.7,
  ARatio = 1,
  HR.obs = 0.8,
  m.obs = 50
)
```

Arguments

`m.con.prior`	prior number of events on control
`m.trt.prior`	prior number of events on treatment
`HR.prior`	Prior Hazard ratio
`ARatio`	Randomization ratio in current trial
`HR.obs`	Observed HR in current trial
`m.obs`	Total events in current trail

Value

A data.frame is returned

Examples

```
my.tte.post <- get.tte.post()
my.tte.post
```

`get.tte.post.df` *Get posterior parameters for two-sample time to event case using normal approximation to log(HR) - data.frame*

Description

Returns the parameters of the posterior distribution in the case of normal approximation to hazard ratio - this version does not expand grid

Usage

```
get.tte.post.df(
  m.con.prior = 10,
  m.trt.prior = 10,
  HR.prior = 0.7,
  ARatio = 1,
  HR.obs = 0.8,
  m.obs = 50
)
```

Arguments

`m.con.prior`	prior number of events on control
`m.trt.prior`	prior number of events on treatment
`HR.prior`	Prior Hazard ratio
`ARatio`	Randomization ratio in current trial
`HR.obs`	Observed HR in current trial
`m.obs`	Total events in current trail

Value

a data.frame is returned

Examples

```
my.tte.post.df <- get.tte.post.df()
head(my.tte.post.df)
```

`get.tte.post.param` *Get TTE posterior parameters*

Description

Update parameters in time to event case

Usage

```
get.tte.post.param(
  m.con.prior = 10,
  m.trt.prior = 10,
  HR.prior = 0.7,
  ARatio = 1,
  HR.obs = 0.8,
  m.obs = 50
)
```

Arguments

`m.con.prior`	prior number of events for control group
`m.trt.prior`	prior number of events for treatment group
`HR.prior`	hazard ratio estimate
`ARatio`	randomization ratio
`HR.obs`	HR observed
`m.obs`	number of events

Value

A data.frame is returned.

Examples

```
my.tte.post.param <- get.tte.post.param()
my.tte.post.param
```

`get.tte.ssize.oc.df` *Get time to event sample size OC data.frame*

Description

Get time to event sample size OC data.frame

Usage

```
get.tte.ssize.oc.df(
  m.con.prior = 10,
  m.trt.prior = 10,
  HR.prior = 0.75,
  ARatio = 1,
  m.obs = 50,
  m.lower = 40,
  m.upper = 120,
  HR.lrv = 0.75,
  HR.tv = 0.75,
  HR.user = 0.845,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`m.con.prior`	number of prior events for control
`m.trt.prior`	number of prior events for treamtent
`HR.prior`	HR estimate
`ARatio`	randomization ratio
`m.obs`	observed number of events
`m.lower`	lower bound on number of events
`m.upper`	upper bound on number of events
`HR.lrv`	Min TPP for HR
`HR.tv`	Base TPP for HR
`HR.user`	user's HR
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data frame is returned

Examples

```
my.tte.ssize.oc.df <- get.tte.ssize.oc.df()
my.tte.ssize.oc.df
```

`get.tte.studyend.GNG` *Get TTE studyend GNG decision*

Description

Get TTE studyend GNG decision

Usage

```
get.tte.studyend.GNG(
  m.con.prior = 50,
  m.trt.prior = 50,
  HR.prior = 1.2,
  ARatio = 1,
  HR.obs = 1.3,
  m.obs = 200,
  HR.tv = 1.4,
  HR.lrv = 1.25,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`m.con.prior`	number of prior events for control
`m.trt.prior`	number of prior events for treatment
`HR.prior`	HR estimate
`ARatio`	randomization ratio
`HR.obs`	Observed HR
`m.obs`	observed events
`HR.tv`	Base TPP for HR
`HR.lrv`	Min TPP for HR
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

a list is returned holding what is needed from data to achieve Go/No-Go

Examples

```
my.tte.studyend.GNG <- get.tte.studyend.GNG(m.con.prior = 50,m.trt.prior = 50, HR.prior=1.2,
ARatio=1, HR.obs=1.3, m.obs = 200, HR.tv= 1.4, HR.lrv = 1.25, tau.tv=.1, tau.lrv=.8, tau.ng=.65)
my.tte.studyend.GNG
```

`get.tte.trt.oc.df` *Get TTE treatment effect OC curve data.frame*

Description

Get time to event treatment oc data.frame

Usage

```
get.tte.trt.oc.df(
  m.con.prior = 10,
  m.trt.prior = 10,
  HR.prior = 0.8,
  ARatio = 0.5,
  m.obs = 50,
  HR.tv = 0.7,
  HR.lrv = 0.9,
  HR.lower = 0.3,
  HR.upper = 2,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65
)
```

Arguments

`m.con.prior`	number of prior events for control
`m.trt.prior`	number of prior events for treamtent
`HR.prior`	HR estimate
`ARatio`	randomization ratio
`m.obs`	observed events
`HR.tv`	HR for Base TPP
`HR.lrv`	HR for Min TPP
`HR.lower`	Lower bound for OC curve
`HR.upper`	Upper bound for OC curve
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go

Value

A data.frame is returned

Author(s)

Greg Cicconetti

Examples

```
my.tte.trt.oc.df <- get.tte.trt.oc.df()
my.tte.trt.oc.df
```

`goparallel` *Goparallel - initiate a parallel computing environment*

Description

Goparallel - initiate a parallel computing environment

Usage

```
goparallel(ncores = 7)
```

Arguments

`ncores`	number of cores

Value

This function initiates a parallel computing environment based on the parallel package

---

`interim.calculator` *interim.calculator*

Description

interim.calculator

Usage

```
interim.calculator(
  a.trt = 1,
  b.trt = 1,
  Delta.lrv = 0.3,
  Delta.tv = 0.45,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  interim.n.t = 15,
  final.n.t = 100,
  p.success = 0.4,
  responses = 10,
  x_ng = NULL,
  x_go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`a.trt`	alpha parameter
`b.trt`	beta parameter
`Delta.lrv`	TPP info
`Delta.tv`	TPP info
`tau.tv`	thresholds
`tau.lrv`	info
`tau.ng`	info
`interim.n.t`	interim sample size
`final.n.t`	final sample size
`p.success`	probability of success
`responses`	number of responders
`x_ng`	number needed for no-go
`x_go`	number needed for go
`go.thresh`	go threshold
`ng.thresh`	no-go threshold

Value

a data.frame is returned

Examples

```
interim.calculator(a.trt = 1, b.trt = 1,Delta.lrv = .3, Delta.tv = .45,tau.tv = 0.10,
tau.lrv = .80, tau.ng = .65,interim.n.t = 15,final.n.t = 100, p.success = .4,
responses=10,x_ng = NULL, x_go=NULL,go.thresh=0.8, ng.thresh=0.8)
```

`Intr_func` *Intr_func*

Description

Intr_func

Usage

```
Intr_func(M, R, ORR)
```

Arguments

`M`	M
`R`	R
`ORR`	ORR

Value

These functions are used by other functions

`list_to_dataframe` *List to data frame from plyr*

Description

List to data frame from plyr

Usage

```
list_to_dataframe(res, labels = NULL, id_name = NULL, id_as_factor = FALSE)
```

Arguments

`res`	list of input data
`labels`	a data frame of labels, one row for each element of res
`id_name`	the name of the index column, NULL for no index column
`id_as_factor`	Logical

Value

A list is converted to a data.frame

`Location.scale.t` *Location-scale t distribution functions*

Description

Location-scale t distribution functions

Usage

```
dt_ls(x, df, mu, sigma)

pt_ls(x, df, mu, sigma)

qt_ls(prob, df, mu, sigma)

rt_ls(n, df, mu, sigma)
```

Arguments

`x`	likelihood function evaluates at point x
`df`	degrees of freedom
`mu`	mean
`sigma`	scale parameter
`prob`	cumulative probability
`n`	sample size

Value

Returns density values, cumulative probabilities, quantiles and random samples from a Location-scale t distribution.

Author(s)

Greg Cicconetti

Examples

```
{
dt_ls(0, 100, 0, 1)
pt_ls(0, 100, 0, 1)
qt_ls(0.5, 100, 0, 1)
rt_ls(100, 100, 0, 1)
}
```

`make.ss.bin.int.dec.GridPlot` *Make single sample binary interim decision grid*

Description

Make single sample binary interim decision grid

Usage

```
make.ss.bin.int.dec.GridPlot(
  InterimDF,
  goThreshold = 0.8,
  nogoThreshold = 0.8,
  include_nogo = TRUE,
  lower.bound.go = 5,
  lower.bound.ng = 10,
  add.black = FALSE
)
```

Arguments

`InterimDF`	call to get.ss.bin.interim.df
`goThreshold`	predictive probability threshold
`nogoThreshold`	predictive probability threshold
`include_nogo`	logical
`lower.bound.go`	lower bound for Go
`lower.bound.ng`	Lower (upper!) bound for no-go
`add.black`	logical

Value

A ggplot object is returned.

`make.ss.bin.int.GStable` *Make single sample binary interim Group-sequential style table*

Description

Make single sample binary interim Group-sequential style table

Usage

```
make.ss.bin.int.GStable(InterimDF, goThreshold = 0.8, nogoThreshold = 0.8)
```

Arguments

`InterimDF`	call to get.ss.bin.interim.df
`goThreshold`	predictive probability threshold
`nogoThreshold`	predictive probability threshold

Value

A data.frame is returned

`make.ss.bin.int.oc.plotSplit`

Make single sample binary interim OC Split plot

Description

Make single sample binary interim OC Split plot

Usage

```
make.ss.bin.int.oc.plotSplit(
  ss.bin.studyend.GNG,
  ss.bin.int.oc,
  ss.bin.int.GNG,
  ss.bin.int.df,
  lower = 0,
  upper = 0.75,
  step = 0.025,
  include_nogo = FALSE
)
```

Arguments

`ss.bin.studyend.GNG`	call to get.ss.bin.studyend.GNG
`ss.bin.int.oc`	call to get.ss.bin.interim.oc
`ss.bin.int.GNG`	call to get.ss.bin.interim.GNG
`ss.bin.int.df`	call to get.ss.bin.int.df
`lower`	lower bound
`upper`	upper bound
`step`	stepsize
`include_nogo`	logical

Value

A ggplot object is returned

Examples

```
my.ss.bin.studyend.GNG = get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
                                       Delta.lrv = .2, Delta.tv = .35,
                                       tau.tv = 0.10, tau.lrv = .80, tau.ng = .65)
my.ss.bin.int.df <- get.ss.bin.int.df(ss.bin.studyend.GNG = my.ss.bin.studyend.GNG,
                                 goThreshold = .8, nogoThreshold = .8, include_nogo =TRUE)
my.ss.bin.int.GNG <- get.ss.bin.int.GNG(ss.bin.int.df = my.ss.bin.int.df,
                                     Interims = 20,
                                     ss.bin.studyend.GNG = my.ss.bin.studyend.GNG)

 my.ss.bin.int.oc <- get.ss.bin.int.oc(
 ss.bin.int.df = my.ss.bin.int.df,
 ss.bin.int.GNG=my.ss.bin.int.GNG)
 my.ss.bin.int.oc.plot <- make.ss.bin.int.oc.plotSplit(
                 ss.bin.studyend.GNG = my.ss.bin.studyend.GNG,
                 ss.bin.int.oc = my.ss.bin.int.oc,
                 ss.bin.int.GNG=my.ss.bin.int.GNG,
                 ss.bin.int.df = my.ss.bin.int.df,
                 include_nogo =TRUE, lower=0, upper=1)
```

`make.ss.bin.int.oc.table`

Make single sample binary interim OC table

Description

Make single sample binary interim OC table

Usage

```
make.ss.bin.int.oc.table(Sims, assessment = "all", rate = 0.3)
```

Arguments

`Sims`	call to get.ss.bin.int.oc.sim
`assessment`	assessment
`rate`	rate

Value

A data.frame is returned

`make.ss.bin.int.ocplot`

Make single sample binary interim OC curve

Description

make.ss.bin.interim.ocplot - creates OC curve with Final and Any Interim

Usage

```
make.ss.bin.int.ocplot(
  ss.bin.studyend.GNG,
  ss.bin.int.oc,
  ss.bin.int.GNG,
  ss.bin.int.df,
  goThreshold = 0.8,
  nogoThreshold = 0.8,
  include_nogo = TRUE,
  lower = 0,
  upper = 1
)
```

Arguments

`ss.bin.studyend.GNG`	call to get.ss.bin.studyend.GNG
`ss.bin.int.oc`	call to get.ss.bin.interim.oc
`ss.bin.int.GNG`	call to get.ss.bin.interim.GNG
`ss.bin.int.df`	call to get.ss.bin.int.df
`goThreshold`	predictive probability threshold for interim
`nogoThreshold`	predictive probability threshold for interim
`include_nogo`	logical
`lower`	lower bound
`upper`	upper bound

Value

A ggplot object is returned

Examples

```
my.ss.bin.studyend.GNG = get.ss.bin.studyend.GNG(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 9,
                                  Delta.lrv = .2, Delta.tv = .35,
                                  tau.tv = 0.10, tau.lrv = .80, tau.ng = .65)
```

```
my.ss.bin.int.df <- get.ss.bin.int.df(ss.bin.studyend.GNG = my.ss.bin.studyend.GNG,
                                goThreshold = .8, nogoThreshold = .8, include_nogo =TRUE)
my.ss.bin.int.GNG <- get.ss.bin.int.GNG(ss.bin.int.df = my.ss.bin.int.df,
                                    Interims = 20,
                                    ss.bin.studyend.GNG = my.ss.bin.studyend.GNG)
my.ss.bin.int.oc <- get.ss.bin.int.oc(ss.bin.int.df = my.ss.bin.int.df,
ss.bin.int.GNG= my.ss.bin.int.GNG)
 my.ss.bin.int.ocplot <- make.ss.bin.int.ocplot(
                 ss.bin.studyend.GNG = my.ss.bin.studyend.GNG,
                 ss.bin.int.oc = my.ss.bin.int.oc,
                 ss.bin.int.GNG=my.ss.bin.int.GNG,
                 ss.bin.int.df = my.ss.bin.int.df,
                 goThreshold = .8,
                 nogoThreshold = .8,
                 include_nogo =TRUE, lower=0, upper=1)
my.ss.bin.int.ocplot
```

`make.ss.bin.int.req.plot` *Make single sample binary predictive probability plot*

Description

Make single sample binary predictive probability plot

Usage

```
make.ss.bin.int.req.plot(my.table = make.ss.bin.int.req.table())
```

Arguments

`my.table`	output from make.ss.bin.pp.table

Value

A ggplot object is returned.

Examples

```
my.ss.bin.int.data.req.plot<- make.ss.bin.int.req.plot()
my.ss.bin.int.data.req.plot
```

`make.ss.bin.int.req.plot2` *Make single sample binary interim Group-sequential style plot*

Description

Make single sample binary interim Group-sequential style plot

Usage

```
make.ss.bin.int.req.plot2(
  InterimDF,
  Delta.lrv = 0.2,
  Delta.tv = 0.3,
  tsize = 5
)
```

Arguments

`InterimDF`	call to get.ss.bin.interim.df
`Delta.lrv`	min TPP
`Delta.tv`	Base TPP
`tsize`	text size - default is for app

Value

A ggplot object is returned

`make.ss.bin.int.req.table` *Make single sample binary pred prob table*

Description

Make single sample binary pred prob table

Usage

```
make.ss.bin.int.req.table(my.df = return.ss.bin.int.req())
```

Arguments

`my.df`	output from return.ss.bin.int.data.req

Value

A data.frame is returned

Examples

```
my.ss.bin.int.req.table <- make.ss.bin.int.req.table()
head(my.ss.bin.int.req.table)
```

`make.ss.bin.ppp` *Make single sample binary prior/posterior plot*

Description

Make single sample binary prior/posterior plot

Usage

```
make.ss.bin.ppp(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 20)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`n.trt`	number of trials
`x.trt`	number of responses

Value

A ggplot object is returned.

Author(s)

Greg Cicconetti

Examples

```
my.ss.bin.ppp <- make.ss.bin.ppp(a.trt = 1, b.trt = 1, n.trt = 40, x.trt = 20)
my.ss.bin.ppp[[1]]
my.ss.bin.ppp[[2]]
```

`make.ss.bin.ria` *Make single sample binary rule in action plot*

Description

Make single sample binary rule in action plot

Usage

```
make.ss.bin.ria(
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  x.trt = 9,
  Delta.lrv = 0.2,
  Delta.tv = 0.35,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  seed = 1234,
  nlines = 25,
  tsize = 4,
  nlines.ria = 20,
  add.table = TRUE
)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`n.trt`	observed sample size
`x.trt`	observed number of responders
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`seed`	random seed
`nlines`	Control for text spacing
`tsize`	Control for text size
`nlines.ria`	number of lines
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ss.bin.ria <- make.ss.bin.ria(x.trt=10, add.table=TRUE)
plot(my.ss.bin.ria[[1]])
my.ss.bin.ria[[2]]
my.ss.bin.ria[[3]]
my.ss.bin.ria[[4]]
```

`make.ss.bin.ssize.oc` *Make single sample binary sample size OC curve*

Description

Make single sample binary sample size OC curve

Usage

```
make.ss.bin.ssize.oc(
  for.plot = get.ss.bin.ssize.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`for.plot`	ouput from get.ss.bin.ssize.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ss.bin.ssize.oc.df <- get.ss.bin.ssize.oc.df()
my.ss.bin.ssize.oc <- make.ss.bin.ssize.oc(for.plot= my.ss.bin.ssize.oc.df, add.table=TRUE)
plot(my.ss.bin.ssize.oc)
```

`make.ss.bin.ssize.oc2` *Make single sample binary sample size OC - trend version*

Description

Make single sample binary sample size OC - trend version

Usage

```
make.ss.bin.ssize.oc2(
  for.plot = get.ss.bin.ssize.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ss.bin.ssize.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A list is returned

Author(s)

Greg Cicconetti

Examples

```
my.ss.bin.ssize.oc.df <-get.ss.bin.ssize.oc.df()
my.ss.bin.ssize.oc2 <- make.ss.bin.ssize.oc2(for.plot = my.ss.bin.ssize.oc.df, add.table=TRUE)
plot(my.ss.bin.ssize.oc2)
```

`make.ss.bin.trt.int.oc` *Make single sample binary Operating characteristic function vs. treatment effect*

Description

Make single sample binary Operating characteristic function vs. treatment effect

Usage

```
make.ss.bin.trt.int.oc(results = get.ss.bin.trt.int.oc.df())
```

Arguments

`results`	output from get.ss.bin.trt.int.oc.df

Value

A ggplot object is returned

Author(s)

Greg Cicconetti

Examples

```
make.ss.bin.trt.int.oc()
```

`make.ss.bin.trt.oc1` *Make single sample binary treatment oc curve*

Description

Make single sample binary treatment oc curve

Usage

```
make.ss.bin.trt.oc1(
  my.df = get.ss.bin.trt.oc.df(),
  tsize = 4,
  nlines = 25,
  nlines.ria = 20,
  Delta_OC_LB = 0,
  Delta_OC_UB = 0.41,
  add.table = TRUE
)
```

Arguments

`my.df`	output from get.ss.bin.trt.oc.df
`tsize`	Control for text size
`nlines`	Control for text spacing
`nlines.ria`	Control for text spacing
`Delta_OC_LB`	Lower bound for OC curve
`Delta_OC_UB`	Upper bound for OC curve
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ss.bin.trt.oc.df <- get.ss.bin.trt.oc.df()
make.ss.bin.trt.oc1(my.df = my.ss.bin.trt.oc.df)
```

`make.ss.bin.trt.oc2` *Make single sample binary treatment oc curve version 2*

Description

Make single sample binary treatment oc curve version 2

Usage

```
make.ss.bin.trt.oc2(
  my.df = get.ss.bin.trt.oc.df(),
  tsize = 4,
  nlines = 25,
  nlines.ria = 20,
  Delta_OC_LB = 0,
  Delta_OC_UB = 1,
  add.table = TRUE
)
```

Arguments

`my.df`	output from get.ss.bin.trt.oc.df
`tsize`	Control for text size
`nlines`	Control for text spacing
`nlines.ria`	Control for text spacing
`Delta_OC_LB`	Lower bound for OC curve
`Delta_OC_UB`	Upper bound for OC curve
`add.table`	provides extended output summaries

Value

A ggplot object is returned.

Examples

```
make.ss.bin.trt.oc2()
```

---

make.ss.ng.int.req.plot

Make single sample normal-gamma interim data requirement plot

Description

Make single sample normal-gamma interim data requirement plot

Usage

```
make.ss.ng.int.req.plot(my.table = make.ss.ng.int.req.table())
```

Arguments

`my.table`	output from make.ss.ng.pp.table

Value

A ggplot object is returned

Examples

```
my.ss.ng.int.req.table <- make.ss.ng.int.req.table()
my.ss.ng.int.req.plot <- make.ss.ng.int.req.plot(my.table = my.ss.ng.int.req.table)
my.ss.ng.int.req.plot
```

`make.ss.ng.int.req.table` *Make single sample normal-gamma interim data requirements table*

Description

Make single sample normal-gamma interim data requirements table

Usage

```
make.ss.ng.int.req.table(my.params = return.ss.ng.int.req())
```

Arguments

`my.params`	output from return.ss.ng.int.req

Value

A data.frame is returned

Examples

```
my.ss.ng.int.req.table <- make.ss.ng.int.req.table()
head(my.ss.ng.int.req.table)
```

`make.ss.ng.ppp` *Make singple sample normal-gamma prior posterior plot*

Description

Make singple sample normal-gamma prior posterior plot

Usage

```
make.ss.ng.ppp(
  mu.0.t = 0,
  n.0.t = 1,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = 1.75,
  s.t = 2,
  n.t = 50,
 gamma.IG.sd.limits = c(as.numeric(trimws(unlist(strsplit("5, 25, 25", ",")))))
)
```

Arguments

`mu.0.t`	prior mean
`n.0.t`	prior effective sample size
`alpha.0.t`	prior alpha parameter
`beta.0.t`	prior beta parameter
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`n.t`	sample size for treatment group
`gamma.IG.sd.limits`	limits used for Precision, Variance and standard deviation visualizers
`limits`	upper limits used for visualizations

Value

A ggplot object is returned

Examples

```
my.ss.ng.ppp <- make.ss.ng.ppp()
my.ss.ng.ppp[[1]][[1]]
my.ss.ng.ppp[[1]][[2]]
gridExtra::grid.arrange(my.ss.ng.ppp[[1]][[1]], my.ss.ng.ppp[[1]][[2]], ncol=2)
my.ss.ng.ppp[[2]]
my.ss.ng.ppp[[3]]
my.ss.ng.ppp[[4]]
my.ss.ng.ppp[[5]]
my.ss.ng.ppp[[6]]
my.ss.ng.ppp[[7]]
```

`make.ss.ng.ria` *Make single sample normal-gamma rule in action plot*

Description

Make single sample normal-gamma rule in action plot

Usage

```
make.ss.ng.ria(
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  n.0.t = 1,
  xbar.t = -0.05,
  s.t = 3,
```

```
    n.t = 10,
    Delta.lrv = 0,
    Delta.tv = 1,
    tau.tv = 0.1,
    tau.lrv = 0.8,
    tau.ng = 0.65,
    tsize = 4,
    nlines = 25,
    nlines.ria = 20,
    add.table = TRUE
  )
```

Arguments

`mu.0.t`	prior mean
`alpha.0.t`	prior alpha parameter
`beta.0.t`	prior beta parameter
`n.0.t`	prior effective sample size
`xbar.t`	observed sample mean
`s.t`	observed sample standard deviation
`n.t`	sample size
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`tsize`	Control for text size
`nlines`	Control for text spacing
`nlines.ria`	Control for text spacing
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ss.ng.ria <- make.ss.ng.ria(add.table=TRUE)
plot(my.ss.ng.ria[[1]])
my.ss.ng.ria[[2]]
my.ss.ng.ria[[3]]
my.ss.ng.ria[[4]]
```

`make.ss.ng.ssize.oc` *Make single sample normal gamma sample size OC*

Description

Make single sample normal gamma sample size OC

Usage

```
make.ss.ng.ssize.oc(
  for.plot = get.ss.ng.ssize.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ss.ng.ssize.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	use TRUE for verbose output

Value

a list is returned

Examples

```
my.ss.ng.ssize.oc.df <- get.ss.ng.ssize.oc.df()
my.ss.ng.ssize.oc <- make.ss.ng.ssize.oc(for.plot=my.ss.ng.ssize.oc.df, add.table=TRUE)
plot(my.ss.ng.ssize.oc)
```

`make.ss.ng.trt.int.oc` *Make single sample normal-gamma interim treatment oc curve*

Description

Make single sample normal-gamma interim treatment oc curve

Usage

```
make.ss.ng.trt.int.oc(
  my.df = get.ss.ng.trt.int.oc.df(npoints = 20, n.MC = 1000, include_nogo = FALSE),
   include_nogo = FALSE
)
```

Arguments

`my.df`	output from get.ss.ng.trt.int.oc.df
`include_nogo`	logical

Value

A ggplot object is returned.

Examples

```
my.ss.ng.trt.int.oc.df <- get.ss.ng.trt.int.oc.df(npoints = 20, n.MC = 1000,
include_nogo = FALSE)
my.ss.ng.trt.int.oc <- make.ss.ng.trt.int.oc(my.df = my.ss.ng.trt.int.oc.df,
include_nogo=FALSE)
my.ss.ng.trt.int.oc[[1]]
my.ss.ng.trt.int.oc[[2]]
my.ss.ng.trt.int.oc[[3]]
```

`make.ss.ng.trt.oc1` *Make single sample normal-gamma treatment OC curve*

Description

Make single sample normal-gamma treatment OC curve

Usage

```
make.ss.ng.trt.oc1(
  my.df = get.ss.ng.trt.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`my.df`	output from get.ss.ng.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ss.ng.trt.oc.df <- get.ss.ng.trt.oc.df()
my.ss.ng.trt.oc1 <- make.ss.ng.trt.oc1(my.df =my.ss.ng.trt.oc.df, add.table=TRUE)
plot(my.ss.ng.trt.oc1[[1]])
```

`make.ss.ng.trt.oc2` *Make single sample normal-gamma treatment OC curve V2*

Description

Make single sample normal-gamma treatment OC curve V2

Usage

```
make.ss.ng.trt.oc2(
  my.df = get.ss.ng.trt.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`my.df`	output from get.ss.ng.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned.

Author(s)

Greg Cicconetti

Examples

```
make.ss.ng.trt.oc2()
```

`make.ts.bin.int.dec.df`
Make two-sample binary interim decision table

Description

Make two-sample binary interim decision table

Usage

```
make.ts.bin.int.dec.df(DecisionTable)
```

Arguments

`DecisionTable`	the results from get.ts.bin.int.dec

Value

a simplified table showing the go/nogo thresholds for each number of control successes.

Examples

```
my.ts.bin.int.dec <- get.ts.bin.int.dec()
my.ts.bin.int.dec.df <- make.ts.bin.int.dec.df(DecisionTable = my.ts.bin.int.dec)
head(my.ts.bin.int.dec.df)
```

`make.ts.bin.int.dec.plot`
Make two-sample binary interim decision plot

Description

Make two-sample binary interim decision plot

Usage

```
make.ts.bin.int.dec.plot(for.plot)
```

Arguments

`for.plot`	the results from get.ts.bin.decision

Value

a ggplot object showing the decision points for the interim analysis.

Examples

```
my.ts.bin.int.dec <- get.ts.bin.int.dec(
a.con = 1, b.con = 1, a.trt = 1, b.trt = 1,
Delta.lrv = 0.15, Delta.tv = .30,
tau.tv = 0.10, tau.lrv = 0.80, tau.ng = 0.65,
go.thresh=.8, ng.thresh=.8,
n.trt = 40, n.con = 40,
n.int.c = c(10, 20, 30), n.int.t = c(10, 20, 30),
runs=500, include_nogo = TRUE)
my.ts.bin.int.dec.plot <- make.ts.bin.int.dec.plot(for.plot=my.ts.bin.int.dec)
my.ts.bin.int.dec.plot
```

`make.ts.bin.int.oc.plot` *Make two-sample binary interim oc plot*

Description

Make two-sample binary interim oc plot

Usage

```
make.ts.bin.int.oc.plot(
  ts.bin.int.oc = get.ts.bin.int.oc(a.con = 1, b.con = 1, a.trt = 1, b.trt = 1, Delta.tv
    = 0.3, Delta.lrv = 0.2, tau.tv = 0.1, tau.lrv = 0.8, tau.ng = 0.65, go.thresh = 0.8,
    ng.thresh = 0.8, n.con = 40, n.trt = 40, n.int.c = c(10, 20, 30), n.int.t = c(10, 20,
    30), DecisionTable = NULL, runs = 5000, ControlRate = 0.2, TreatmentEffect = seq(0,
      0.8, 0.1), include_nogo = TRUE),
   nlines = 25,
   tsize = 4,
   include_nogo = TRUE
)
```

Arguments

`ts.bin.int.oc`	Results from get.ts.bin.int.oc
`nlines`	Control for text spacing
`tsize`	Control for text size
`include_nogo`	logical

Value

A ggplot showing the probability of each out come given a particular difference from control

---

`make.ts.bin.int.oc.table` *Make two-sample binary interim OC table*

Description

Make two-sample binary interim OC table

Usage

```
make.ts.bin.int.oc.table(
  OCsims = get.ts.bin.int.oc(),
  Diff = 0.1,
  interim = "any"
)
```

Arguments

`OCsims`	results from get.ts.bin.int.oc
`Diff`	the difference of interest for the table, must be a difference value included in OCsims
`interim`	either a specific interim analysis or 'any'

Value

returns a 3x3 table showing the probabilities of each outcome for the interim and final analysis. If interm is 'any' the interim analysis is the first non-consider result observed at any interim.

`make.ts.bin.pp.heatmap` *Make two-sample binary predictive probability heatmap*

Description

Make two-sample binary predictive probability heatmap

Usage

```
make.ts.bin.pp.heatmap(my.data = pp.table, go.thresh = 0.8, ng.thresh = 0.8)
```

Arguments

`my.data`	output from return.ts.bin.pp.table
`go.thresh`	go threshold
`ng.thresh`	no-go threshold

Value

A ggplot object is returned.

Examples

```
pp.table <- return.ts.bin.int.predprob.table(goparallel=FALSE)
make.ts.bin.pp.heatmap(my.data = pp.table, go.thresh=0.8, ng.thresh=0.8)
```

---

make.ts.bin.ppp *Make two-sample binary prior/posterior plot*

Description

Make two-sample binary prior/posterior plot

Usage

```
make.ts.bin.ppp(
  a.con = 1,
  b.con = 1,
  n.con = 40,
  x.con = 9,
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  x.trt = 17
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`n.con`	number of trials for control group
`x.con`	number of responses for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior alpha parameter for treatment group
`n.trt`	number of trials for treatment group
`x.trt`	number of responses for treatment group

Value

A list is returned.

Examples

```
my.ts.bin.ppp <- make.ts.bin.ppp()
my.ts.bin.ppp[[1]]
my.ts.bin.ppp[[2]]
```

`make.ts.bin.ria` *Make two-sample binary rule in action plot*

Description

Make two-sample binary rule in action plot

Usage

```
make.ts.bin.ria(
  a.con = 1,
  b.con = 1,
  n.con = 40,
  x.con = 9,
  a.trt = 1,
  b.trt = 1,
  n.trt = 40,
  x.trt = 17,
  Delta.lrv = 0.15,
  Delta.tv = 0.3,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  nlines.ria = 20,
  tsize = 4,
  nlines = 25,
  add.table = TRUE
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`n.con`	sample size for control group
`x.con`	number of responders for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`n.trt`	sample size for control treatment group
`x.trt`	number of responders for treatment group

`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`nlines.ria`	Control for text spacing
`tsize`	Control for text size
`nlines`	Control for text spacing
`add.table`	provides extended output summaries

Value

a ggplot object is returned

Examples

```
my.ts.bin.ria <- make.ts.bin.ria(add.table=TRUE)
plot(my.ts.bin.ria[[1]])
my.ts.bin.ria[[2]]
my.ts.bin.ria[[3]]
my.ts.bin.ria[[4]]
```

`make.ts.bin.ssize.oc` *Make two-sample binary sample size operating characteristics curve*

Description

Make two-sample binary sample size operating characteristics curve

Usage

```
make.ts.bin.ssize.oc(
  for.plot = get.ts.bin.ssize.oc.df(),
  tsize = 4,
  nlines = 25,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ts.bin.ssize.oc.df
`tsize`	Control for text size
`nlines`	Control for text spacing
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ts.bin.ssize.oc.df <- get.ts.bin.ssize.oc.df()
make.ts.bin.ssize.oc(for.plot = my.ts.bin.ssize.oc.df)
```

`make.ts.bin.studyend.GNG.hm` *Make two-sample binary study end GNG heatmap*

Description

Make two-sample binary study end GNG heatmap

Usage

```
make.ts.bin.studyend.GNG.hm(my.df = return.ts.bin.studyend.GNG.hm.df())
```

Arguments

`my.df`	output from return.ts.bin.studyend.GNG.hm.df

Value

A ggplot object is returned

Examples

```
make.ts.bin.studyend.GNG.hm()
```

`make.ts.bin.trt.oc1` *Make two-sample binary treatment OC curve*

Description

Make two-sample binary treatment OC curve

Usage

```
make.ts.bin.trt.oc1(
  for.plot = get.ts.bin.trt.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ts.bin.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned.

Examples

```
my.ts.bin.trt.oc.df <- get.ts.bin.trt.oc.df()
my.ts.bin.trt.oc1 <- make.ts.bin.trt.oc1(for.plot=my.ts.bin.trt.oc.df, add.table=TRUE)
plot(my.ts.bin.trt.oc1[[1]])
my.ts.bin.trt.oc1[[2]]
my.ts.bin.trt.oc1[[3]]
```

`make.ts.bin.trt.oc2` *Make two-sample binary treatment OC curve V2*

Description

Make two-sample binary treatment OC curve V2

Usage

```
make.ts.bin.trt.oc2(
  for.plot = get.ts.bin.trt.oc.df(),
  tsize = 4,
  nlines = 25,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ts.bin.trt.oc.df
`tsize`	Control for text size
`nlines`	Control for text spacing
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ts.bin.trt.oc.df <- get.ts.bin.trt.oc.df()
my.ts.bin.trt.oc2 <- make.ts.bin.trt.oc2(for.plot=my.ts.bin.trt.oc.df, add.table=TRUE)
plot(my.ts.bin.trt.oc2[[1]])
my.ts.bin.trt.oc2[[2]]
my.ts.bin.trt.oc2[[3]]
```

`make.ts.ng.int.oc` *Make two-sample normal-gamma interim OC plot*

Description

Make two-sample normal-gamma interim OC plot

Usage

```
make.ts.ng.int.oc(
  for.plot = get.ts.ng.trt.int.oc.df(go.parallel = FALSE, include_nogo = TRUE),
  nlines = 25,
  tsize = 6,
  include_nogo = TRUE
)
```

Arguments

`for.plot`	call to get.ts.ng.trt.int.oc.df
`nlines`	number of lines
`tsize`	text size
`include_nogo`	logical

Value

A ggplot object is returned

Examples

```
my.ts.ng.trt.int.oc.df <- get.ts.ng.trt.int.oc.df(npointsLookup = 2, npoints=3, n.MC.lookup=5,
n.MC=5, go.parallel=FALSE)
make.ts.ng.int.oc(for.plot = my.ts.ng.trt.int.oc.df)
```

`make.ts.ng.ppp` *Make two-sample normal-gamma prior/posterior plot*

Description

Make two-sample normal-gamma prior/posterior plot

Usage

```
make.ts.ng.ppp(
  mu.0.c = 0,
  alpha.0.c = 2.5,
  beta.0.c = 10,
  n.0.c = 10,
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  n.0.t = 1e-04,
  xbar.c = 0.25,
  s.c = 1.5,
  n.c = 55,
  xbar.t = 2.5,
  s.t = 1.5,
  n.t = 55,
  limits = c(5, 25, 10)
)
```

Arguments

`mu.0.c`	prior mean for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`n.0.c`	prior effective sample size parameter for control group
`mu.0.t`	prior mean for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`n.0.t`	prior effective sample size parameter for treatment group
`xbar.c`	control mean
`s.c`	control sd
`n.c`	control sample size
`xbar.t`	treatment mean
`s.t`	treatment sd
`n.t`	treatment sample size
`limits`	limits for visualizing precision, variance, standard deviation

Value

a ggplot object is returned

Examples

```
my.ts.ng.ppp <- make.ts.ng.ppp()
my.ts.ng.ppp[[1]][[1]]
my.ts.ng.ppp[[1]][[2]]
my.ts.ng.ppp[[1]][[3]]
my.ts.ng.ppp[[1]][[4]]
gridExtra::grid.arrange(my.ts.ng.ppp[[1]][[1]], my.ts.ng.ppp[[1]][[3]],
my.ts.ng.ppp[[1]][[2]], my.ts.ng.ppp[[1]][[4]])
my.ts.ng.ppp[[2]]
my.ts.ng.ppp[[3]]
my.ts.ng.ppp[[4]]
my.ts.ng.ppp[[5]]
my.ts.ng.ppp[[6]]
my.ts.ng.ppp[[7]]
```

`make.ts.ng.ria` *Make two-sample normal-gamma rule in action plot*

Description

Make two-sample normal-gamma rule in action plot

Usage

```
make.ts.ng.ria(
  mu.0.c = 0,
  alpha.0.c = 2.5,
  beta.0.c = 10,
  n.0.c = 10,
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  n.0.t = 1e-04,
  xbar.c = 0.25,
  s.c = 1.5,
  n.c = 40,
  xbar.t = 2.5,
  s.t = 1.5,
  n.t = 40,
  Delta.lrv = 1.5,
  Delta.tv = 3,
  tau.ng = 0.65,
  tau.lrv = 0.8,
  tau.tv = 0.2,
  seed = 1234,
  n.MC = 1000,
  nlines = 25,
  nlines.ria = 20,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`mu.0.c`	prior mean for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`n.0.c`	prior effective sample size parameter for control group
`mu.0.t`	prior mean for treatment group
`alpha.0.t`	prior alpha parameter for treatment group

`beta.0.t`	prior beta parameter for treatment group
`n.0.t`	prior effective sample size parameter for treatment group
`xbar.c`	control mean
`s.c`	control sd
`n.c`	control sample size
`xbar.t`	treatment mean
`s.t`	treatment sd
`n.t`	treatment sample size
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.ng`	threshold associated with No-Go
`tau.lrv`	threshold associated with Min TPP
`tau.tv`	threshold associated with Base TPP
`seed`	random seed
`n.MC`	n for MC sampling
`nlines`	Control for text spacing
`nlines.ria`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ts.ng.ria <- make.ts.ng.ria(add.table=TRUE)
plot(my.ts.ng.ria[[1]])
my.ts.ng.ria[[2]]
my.ts.ng.ria[[3]]
my.ts.ng.ria[[4]]
```

`make.ts.ng.ssize.oc` *Make two-sample normal gamma sample size OC curve*

Description

Make two-sample normal gamma sample size OC curve

Usage

```
make.ts.ng.ssize.oc(
  for.plot = get.ts.ng.ssize.oc.df(),
  tsize = 4,
  nlines = 25,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ts.ng.ssize.oc.df
`tsize`	Control for text size
`nlines`	Control for text spacing
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ts.ng.ssize.oc.df <- get.ts.ng.ssize.oc.df(goparallel=FALSE)
my.ts.ng.ssize.oc <- make.ts.ng.ssize.oc()
my.ts.ng.ssize.oc
```

`make.ts.ng.studyend.criteria` *Create lookup table for study-end*

Description

Create lookup table for study-end

Usage

```
make.ts.ng.studyend.criteria(
  mu.0.c = 0,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  n.0.c = 1,
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  n.0.t = 1,
```

```
    xbar.c.LB = 0,
    xbar.c.UB = 5,
    npoints = 6,
    s.c = 7,
    n.c = 55,
    xbar.t = 0,
    s.t = 7,
    n.t = 55,
    Delta.lrv = 5.5,
    Delta.tv = 6.5,
    tau.ng = 0.65,
    tau.lrv = 0.8,
    tau.tv = 0.1,
    n.MC = 250,
    go.parallel = TRUE,
    cl = cl
  )
```

Arguments

`mu.0.c`	prior parameter, mean, control
`alpha.0.c`	prior parameter, precision, alpha, control
`beta.0.c`	prior parameter, precision, beta, control
`n.0.c`	prior parameter, effective sample size, control
`mu.0.t`	prior parameter, mean, treatment
`alpha.0.t`	prior parameter, precision, alpha, treatment
`beta.0.t`	prior parameter, precision, beta, treatment
`n.0.t`	prior parameter, effective sample size, treatment
`xbar.c.LB`	lowerbound for xbar on control
`xbar.c.UB`	upperbound for xbar on control
`npoints`	number of points
`s.c`	sample sd, control
`n.c`	sample size, control
`xbar.t`	sample mean - treatment
`s.t`	sample sd, treatment
`n.t`	sample size, treatment
`Delta.lrv`	TTP info
`Delta.tv`	TTP info
`tau.ng`	thresholds
`tau.lrv`	thresholds
`tau.tv`	thresholds
`n.MC`	MC size
`go.parallel`	logical to engage parallel computing
`cl`	cluster

Value

A data.frame is returned

Examples

```
make.ts.ng.studyend.criteria(go.parallel=FALSE)
```

`make.ts.ng.studyend.GNG.hm` *Make two-sample normal-gamma studyend GNG Heatmap*

Description

Make two-sample normal-gamma studyend GNG Heatmap

Usage

```
make.ts.ng.studyend.GNG.hm(my.df = return.ts.ng.studyend.GNG.hm.df())
```

Arguments

`my.df`	output from return.ts.ng.studyend.GNG.hm.df

Value

A ggplot object is returned

Author(s)

Greg Cicconetti

Examples

```
make.ts.ng.studyend.GNG.hm()
```

---

make.ts.ng.trt.oc1 *Make two-sample normal-gamma treatment OC curve*

Description

Make two-sample normal-gamma treatment OC curve

Usage

```
make.ts.ng.trt.oc1(
  for.plot = get.ts.ng.trt.oc.df(goparallel = FALSE),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ts.ng.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ts.ng.trt.oc.df <- get.ts.ng.trt.oc.df(goparallel=FALSE)
my.ts.ng.trt.oc1 <- make.ts.ng.trt.oc1(for.plot = my.ts.ng.trt.oc.df, add.table=TRUE)
plot(my.ts.ng.trt.oc1[[1]])
my.ts.ng.trt.oc1[[2]]
my.ts.ng.trt.oc1[[3]]
```

---

make.ts.ng.trt.oc2 *Make two-sample normal-gamma treatment OC curve V2*

Description

Make two-sample normal-gamma treatment OC curve V2

Usage

```
make.ts.ng.trt.oc2(
  for.plot = get.ts.ng.trt.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.ts.ng.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.ts.ng.trt.oc.df <- get.ts.ng.trt.oc.df(goparallel=FALSE)
make.ts.ng.trt.oc2(for.plot=my.ts.ng.trt.oc.df, add.table=TRUE)
```

`make.tte.int.data.plot` *Make time to event predictive probability cutoff plot*

Description

Make time to event predictive probability cutoff plot

Usage

```
make.tte.int.data.plot(
  my.table = make.tte.int.data.table(),
  include_nogo = FALSE
)
```

Arguments

`my.table`	output from make.tte.int.data.table
`include_nogo`	logical

Value

A ggplot object is returned

Examples

```
my.tte.int.data.table <- make.tte.int.data.table()
my.tte.int.data.plot <- make.tte.int.data.plot(my.table = my.tte.int.data.table)
my.tte.int.data.plot
```

`make.tte.int.data.table`

Make TTE predictive probability table

Description

Make TTE predictive probability table

Usage

```
make.tte.int.data.table(my.params = return.tte.int.data.req())
```

Arguments

`my.params`	output from return.tte.int.data.req

Value

A data.frame is returned

Examples

```
my.tte.int.data.req <- return.tte.int.data.req()
my.tte.int.data.table <- make.tte.int.data.table(my.params = my.tte.int.data.req)
my.tte.int.data.table
```

`make.tte.int.oc` *Make TTE interim treatment OC curve*

Description

Make TTE interim treatment OC curve

Usage

```
make.tte.int.oc(
  my.df = get.tte.int.oc.df(npoints = 20, include_nogo = TRUE),
  nlines = 25,
  tsize = 6,
  HR.lower = 0.0025,
  HR.upper = 2,
  include_nogo = TRUE
)
```

Arguments

`my.df`	output from get.tte.int.OC.df
`nlines`	number of lines
`tsize`	text size
`HR.lower`	lower bound for HR
`HR.upper`	upper bound for HR
`include_nogo`	logical

Value

A ggplot object is returned

Examples

```
make.tte.int.oc()
```

`make.tte.ppp` *Make Time to event prior/posterior plot*

Description

Make Time to event prior/posterior plot

Usage

```
make.tte.ppp(
  m.con.prior = 10,
  m.trt.prior = 10,
  HR.prior = 0.845,
  ARatio = 1,
  HR.obs = 0.75,
  m.obs = 50
)
```

Arguments

`m.con.prior`	prior number of control events
`m.trt.prior`	prior number of treatment events
`HR.prior`	HR estimate
`ARatio`	Randomization ratio
`HR.obs`	HR observed
`m.obs`	Number of observed events

Value

A ggplot object is returned

Examples

```
make.tte.ppp()
```

`make.tte.ria` *Make time to event rule in action plot*

Description

Make time to event rule in action plot

Usage

```
make.tte.ria(
  m.con.prior = 1,
  m.trt.prior = 1,
  HR.prior = 1,
  ARatio = 0.5,
  HR.obs = 0.845,
  m.obs = 1500,
  HR.tv = 0.8,
  HR.lrv = 0.9,
  tau.tv = 0.1,
  tau.lrv = 0.7,
  tau.ng = 0.7,
  tsize = 4,
  nlines = 25,
  nlines.ria = 20,
  add.table = TRUE
)
```

Arguments

`m.con.prior`	prior number of control events
`m.trt.prior`	prior number of treatment events
`HR.prior`	prior estimate for HR
`ARatio`	Allocation ratio
`HR.obs`	observed hazard ratio
`m.obs`	observed number of events
`HR.tv`	TPP Target Value aka Base TPP
`HR.lrv`	TPP Lower Reference Value aka Max TPP (large HRs lead to No-Go)
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`tsize`	Control for text size
`nlines`	Control for text spacing
`nlines.ria`	Control for text spacing
`add.table`	provides extended output summaries

Value

A ggplot object is returned.

Examples

```
my.tte.ria <- make.tte.ria(add.table=TRUE)
plot(my.tte.ria[[1]])
my.tte.ria[[2]]
my.tte.ria[[3]]
my.tte.ria[[4]]
```

`make.tte.ssize.oc` *Make time-to-event sample size OC curve*

Description

Make time-to-event sample size OC curve

Usage

```
make.tte.ssize.oc(
  for.plot = get.tte.ssize.oc.df(),
  tsize = 4,
  nlines = 25,
  add.table = TRUE
)
```

Arguments

`for.plot`	output from get.tte.ssize.oc.df
`tsize`	Control for text size
`nlines`	Control for text spacing
`add.table`	provide GNG Rule tables and extensive output

Value

A ggplot object is returned

Examples

```
my.tte.ssize.oc.df <- get.tte.ssize.oc.df()
make.tte.ssize.oc(for.plot = my.tte.ssize.oc.df)
```

`make.tte.trt.oc1` *Make time to event treatment OC curve*

Description

Make time to event treatment OC curve

Usage

```
make.tte.trt.oc1(
  plot.df = get.tte.trt.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`plot.df`	output from get.tte.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.tte.trt.oc.df <- get.tte.trt.oc.df()
make.tte.trt.oc1(plot.df = my.tte.trt.oc.df, add.table=TRUE)
```

`make.tte.trt.oc2` *Make time to event treatment OC curve v2*

Description

Make time to event treatment OC curve v2

Usage

```
make.tte.trt.oc2(
  plot.df = get.tte.trt.oc.df(),
  nlines = 25,
  tsize = 4,
  add.table = TRUE
)
```

Arguments

`plot.df`	output from get.tte.trt.oc.df
`nlines`	Control for text spacing
`tsize`	Control for text size
`add.table`	provides extended output summaries

Value

A ggplot object is returned

Examples

```
my.tte.trt.oc.df <- get.tte.trt.oc.df()
make.tte.trt.oc2(plot.df = my.tte.trt.oc.df, add.table=TRUE)
```

`normal.gamma` *Density function for the Normal-Gamma distribution*

Description

Density function for the Normal-Gamma distribution

Usage

```
dnorgam(mu, tau, mu0, n0, a0, b0)

rnormgam(n = 1e+05, mu.0 = 0, n.0 = 1, alpha.0 = 0.01, beta.0 = 0.01)
```

Arguments

`mu`	likelihood evaluated when mean takes on value mu
`tau`	likelihood evaluated when precision takes on value tau
`mu0`	hyperparameter describing mu
`n0`	hyperparameter describing effective sample size associated with mu0
`a0`	hyperparameter describing shape parameter of precision parameter
`b0`	hyperparameter describing rate parameter of precision parameter
`n`	number of observations
`mu.0`	hyperparameter describing mu
`n.0`	hyperparameter describing effective sample size associated with mu0
`alpha.0`	hyperparameter describing shape parameter of precision parameter
`beta.0`	hyperparameter describing rate parameter of precision parameter

Value

Returns the value of the Normal-gamma density function at the point passed.

Examples

```
dnorgam(100, .25, 0, 10, 0, .25)
rnormgam()
```

`ocTable_multi` *Single sample binary helper functions*

Description

Single sample binary helper functions

Usage

```
ocTable_multi(DesignTable, TargetRate)
```

Arguments

`DesignTable`	DesignTable
`TargetRate`	TargetRate

Value

These functions are used by other functions

These functions are used by other functions

`p2beta_diff` *Probability for difference of x or more between two beta distributions*

Description

Probability for difference of x or more between two beta distributions

Usage

```
p2beta_diff(x, a1, b1, a2, b2)
```

Arguments

`x`	value
`a1`	alpha for first
`b1`	beta for first
`a2`	alpha for second
`b2`	beta for second

Value

returns the probability of a difference of x or more between two beta distributions. This is a simplification of the original P2beta function simplified for vectorization. Therre is a lot of space here for improved run times.

Author(s)

Randall Henner

`p2beta_diff_Vector` *Probability for difference of x or more between two beta distributions*

Description

Probability for difference of x or more between two beta distributions

Usage

```
p2beta_diff_Vector(x, a1, b1, a2, b2)
```

Arguments

`x`	value
`a1`	alpha for first
`b1`	beta for first
`a2`	alpha for second
`b2`	beta for second

Value

a function is returned

`pbetabinom_c` *Replacement for pbetanimom*

Description

Replacement for pbetanimom

Usage

```
pbetabinom_c(q, size, m, s)
```

Arguments

`q`	vector of quarantines
`size`	vector of totals
`m`	vector of probabilities of success
`s`	vector of over dispersion parameters

Value

This is a modification of the pbetanimom function from the rmutil package. The original function would throw an error for non-nonsensical values rather than returning 0 or 1.

`pooled.sd` *Pooled standard deviation*

Description

Computes to the pooled standard deviation.

Usage

```
pooled.sd(s = c(4, 5), n = c(14, 20))
```

Arguments

`s`	expression to be plotted
`n`	number of points to plot

Value

A vector holding the pooled standard deviation is returned.

Author(s)

Greg Cicconetti

Examples

```
pooled.sd()
```

`probmat.func` *probmat.func*

Description

probmat.func

Usage

```
probmat.func(DesignTable, TargetRate)
```

Arguments

`DesignTable`	DesignTable
`TargetRate`	TargetRate

Value

These functions are used by other functions

`Reparameterized.beta` *Reparameterized Beta distribution functions*

Description

Reparameterized Beta distribution functions

Usage

```
dbeta.rp(x, mean = 0.5, effective.ss = 1, ncp = 0, log = FALSE)

pbeta.rp(
  q,
  mean = 0.5,
  effective.ss = 1,
  ncp = 0,
  lower.tail = TRUE,
  log.p = FALSE
)

qbeta.rp(
  p,
```

```
  mean = 0.5,
  effective.ss = 1,
  ncp = 0,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

`x`	likelihood function evaluates at point x
`mean`	mean
`effective.ss`	effective sample size
`ncp`	non-centrality parameter
`log`	as in stats::*beta
`q`	quantile
`lower.tail`	logical; if TRUE (default), probabilities are P(X < x), otherwise, P(X>x).
`log.p`	logical; if TRUE, probabilities p are given as log(p)
`p`	cumulative probability

Value

Returns density values, cumulative probabilities, quantiles and random samples from a Beta distribution.

Examples

```
dbeta.rp(.5, .5, 1)
pbeta.rp(.5, .5, 1)
qbeta.rp(.975, .5, 1)
```

`Reparameterized.gamma` *Reparameterized Gamma distribution functions*

Description

Reparameterized Gamma distribution functions

Usage

```
dgamma.rp(x, mean, var, log = FALSE)

pgamma.rp(q, mean, var, lower.tail = TRUE, log.p = FALSE)

qgamma.rp(p, mean, var, lower.tail = TRUE, log.p = FALSE)

rgamma.rp(n, mean, var)
```

Arguments

`x`	likelihood function evaluates at point x
`mean`	mean
`var`	variance
`log`	as in stats::*gamma
`q`	quantile
`lower.tail`	logical; if TRUE (default), probabilities are P(X < x), otherwise, P(X>x).
`log.p`	logical; if TRUE, probabilities p are given as log(p)
`p`	cumulative probability
`n`	sample size

Value

Returns density values, cumulative probabilities, quantiles and random samples from a gamma distribution.

Examples

```
dgamma.rp(1, 1, 1)
pgamma.rp(1.96, 1, 1)
qgamma.rp(.975, 1, 1)
rgamma.rp(10, 1, 1)
```

`Reparameterized.normal` *Reparameterized normal functions*

Description

Reparameterized normal functions

Usage

```
dnorm.rp(x, mean = 0, tau = 1, log = FALSE)

pnorm.rp(q, mean = 0, tau = 1, lower.tail = TRUE, log.p = FALSE)

qnorm.rp(p, mean = 0, tau = 1, lower.tail = TRUE, log.p = FALSE)

rnorm.rp(n, mean = 0, tau = 1)
```

Arguments

`x`	likelihood function evaluates at point x
`mean`	mean
`tau`	precision parameter
`log`	logical; if TRUE, probabilities p are given as log(p)
`q`	quantile
`lower.tail`	logical; if TRUE (default), probabilities are P(X < x) otherwise, P(X>x).
`log.p`	logical; if TRUE, probabilities p are given as log(p)
`p`	cumulative probability
`n`	number of observations. If length(n) > 1, the length is taken to be the number required.

Value

Returns density values, cumulative probabilities, quantiles and random samples from a normal distribution.

Examples

```
dnorm.rp(x=0)
pnorm.rp(q=.1)
qnorm.rp(p=.975)
rnorm.rp(10)
```

`return.ss.bin.int.req` *Return single sample binary predictive probability*

Description

Return single sample binary predictive probability

Usage

```
return.ss.bin.int.req(
  a.trt = 1,
  b.trt = 1,
  Delta.lrv = 0.2,
  Delta.tv = 0.35,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  interim.n.t = c(0:39),
  final.n.t = 40,
  p.success = 0.5,
  x.ng = NULL,
```

```
  x.go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`a.trt`	prior alpha parameter
`b.trt`	prior beta parameter
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`interim.n.t`	number of trials at interim
`final.n.t`	number of trials at final
`p.success`	probability of success
`x.ng`	responses for go; leave null for standard rule
`x.go`	responses for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability at interim
`ng.thresh`	no-go threshold for predictive probability at interim

Value

Return single sample binary predictive probability

Examples

```
holdit <- return.ss.bin.int.req()
head(holdit)
```

`return.ss.ng.data.req.df` *Get single sample normal-gamma interim treatment effect OC data.frame*

Description

Get single sample normal-gamma interim treatment effect OC data.frame

Usage

```
return.ss.ng.data.req.df(
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = 0.5,
  s.t = 2,
  interim.n.t = 10,
  final.n.t = 40,
  Delta.lrv = 1.25,
  Delta.tv = 1.75,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  xbar_ng = NULL,
  xbar_go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  include_nogo = TRUE
)
```

Arguments

`mu.0.t`	prior mean
`n.0.t`	prior effective sample size
`alpha.0.t`	prior alpha parameter
`beta.0.t`	prior beta parameter
`xbar.t`	sample mean at interim
`s.t`	treatment standard deviation
`interim.n.t`	interim sample sizes
`final.n.t`	final sample size
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`xbar_ng`	Leave NULL to compute or provide user value
`xbar_go`	Leave NULL to compute or provide user value
`go.thresh`	interim go threshold
`ng.thresh`	interim no-go threshold
`include_nogo`	logical

Value

returns data.frame to assist with creating interim oc curve in single sample normal-gamma case

Examples

```
return.ss.ng.data.req.df()
```

---

return.ss.ng.int.data.req

Return single sample normal-gamma predictive probability

Description

Return single sample normal-gamma predictive probability

Usage

```
return.ss.ng.int.data.req(
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = seq(-5, 5, 0.1),
  s.t = 2,
  interim.n.t = c(5, 10, 15, 20, 25, 30, 35),
  final.n.t = 40,
  Delta.lrv = 1.25,
  Delta.tv = 1.75,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  xbar_ng = NULL,
  xbar_go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group

`interim.n.t`	interim sample size
`final.n.t`	final sample size
`Delta.lrv`	Min TPP
`Delta.tv`	Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`xbar_ng`	xbar for no-go; leave null for standard rule
`xbar_go`	xbar for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability
`ng.thresh`	no-go threshold for predictive probability

Value

A data.frame is returned

Examples

```
{
my.ss.ng.int.data.req <- return.ss.ng.int.data.req()
head(my.ss.ng.int.data.req)
}
```

`return.ss.ng.int.req` *Return single sample normal-gamma predictive probability*

Description

Return single sample normal-gamma predictive probability

Usage

```
return.ss.ng.int.req(
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  xbar.t = seq(-5, 5, 0.1),
  s.t = 2,
  interim.n.t = c(5, 10, 15, 20, 25, 30, 35),
  final.n.t = 40,
  Delta.lrv = 1.25,
  Delta.tv = 1.75,
  tau.tv = 0.1,
  tau.lrv = 0.8,
```

```
  tau.ng = 0.65,
  xbar_ng = NULL,
  xbar_go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`xbar.t`	sample mean for treatment group
`s.t`	sample sd for treatment group
`interim.n.t`	interim sample size
`final.n.t`	final sample size
`Delta.lrv`	Min TPP
`Delta.tv`	Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`xbar_ng`	xbar for no-go; leave null for standard rule
`xbar_go`	xbar for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability
`ng.thresh`	no-go threshold for predictive probability

Value

A data.frame is returned

Examples

```
my.ss.ng.int.req <- return.ss.ng.int.req()
head(my.ss.ng.int.req)
```

`return.ts.bin.int.predprob` *Return two-sample binary interim predictive probability*

Description

Return two-sample binary interim predictive probability

Usage

```
return.ts.bin.int.predprob(
  a.con = 1,
  b.con = 1,
  a.trt = 1,
  b.trt = 1,
  Delta.lrv = 0.3,
  Delta.tv = 0.4,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.8,
  n.int.con = 20,
  n.int.trt = 20,
  x.int.con = 17,
  x.int.trt = 17,
  n.final.trt = 40,
  n.final.con = 40
)
```

Arguments

`a.con`	alpha parameter for control
`b.con`	beta parameter for control
`a.trt`	alpha parameter for trt
`b.trt`	beta parameter for trt
`Delta.lrv`	TPP info
`Delta.tv`	TPP info
`tau.tv`	Threshold
`tau.lrv`	Threshold
`tau.ng`	Threshold
`n.int.con`	Sample size, control group at interim
`n.int.trt`	Sample size, treatment group at interim
`x.int.con`	Responders, control group at interim
`x.int.trt`	Responders, treatment group at interim
`n.final.trt`	Final sample size treatment
`n.final.con`	Final sample size control

Value

A data.frame is returned

Examples

```
return.ts.bin.int.predprob()
```

`return.ts.bin.int.predprob.table` *Return two-sample binary predictive probability table*

Description

Return two-sample binary predictive probability table

Usage

```
return.ts.bin.int.predprob.table(
  a.con = 0.5,
  b.con = 0.5,
  a.trt = 0.5,
  b.trt = 0.5,
  Delta.lrv = 0.08,
  Delta.tv = 0.08,
  tau.tv = 0.7,
  tau.lrv = 0.7,
  tau.ng = 0,
  n.int.con = 5,
  n.int.trt = 5,
  n.trt = 10,
  n.con = 10,
  p.con = 0.2,
  p.trt = 0.2 + seq(0, 0.5, 0.1),
  studyend = NULL,
  goparallel = FALSE
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior alpha parameter for treatment group
`Delta.lrv`	TPP Lower Reference Value aka Min TPP

`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`n.int.con`	sample size for control group at interim
`n.int.trt`	sample size for treatment group at interim
`n.trt`	sample size for control group at interim
`n.con`	sample size for treatment group at interim
`p.con`	probability of success for control group
`p.trt`	probability of success for treatment group
`studyend`	keep null
`goparallel`	Option to use parallel computing

Value

A data.frame is returned

Author(s)

Greg Cicconetti

Examples

```
my.ts.bin.int.predprob.table <- return.ts.bin.int.predprob.table()
```

`return.ts.bin.studyend.GNG.hm.df` *Return two-sample binary study end GNG heatmap data.frame*

Description

Return two-sample binary study end GNG heatmap data.frame

Usage

```
return.ts.bin.studyend.GNG.hm.df(
  a.con = 1,
  b.con = 1,
  n.con = 30,
  a.trt = 1,
  b.trt = 1,
  n.trt = 30,
```

```
  Delta.lrv = 0.3,
  Delta.tv = 0.4,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  x_ng = NULL,
  x_go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`a.con`	prior alpha parameter for control group
`b.con`	prior beta parameter for control group
`n.con`	sample size for control group
`a.trt`	prior alpha parameter for treatment group
`b.trt`	prior beta parameter for treatment group
`n.trt`	sample size for control treatment group
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`x_ng`	responses needed for no-go; leave null for standard rule
`x_go`	responses needed for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability
`ng.thresh`	no-go threshold for predictive probability

Value

A data.frame is returned

Examples

```
my.ts.bin.studyend.GNG.hm.df <- return.ts.bin.studyend.GNG.hm.df()
head(my.ts.bin.studyend.GNG.hm.df)
```

`return.ts.ng.int.req.df`
Return two-sample normal-gamma predictive probability

Description

Return two-sample normal-gamma predictive probability

Usage

```
return.ts.ng.int.req.df(
  mu.0.t = 0,
  n.0.t = 1e-04,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  mu.0.c = 0,
  n.0.c = 1e-04,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  xbar.t = c(1.9, 2, 2.1, 2.05),
  s.t = c(2, 2.1, 1.9, 2.04),
  n.t = c(10, 20, 30, 40),
  xbar.c = c(1, 1.1, 1.5, 1.25),
  s.c = c(1.9, 2, 2.5, 2.25),
  n.c = c(10, 20, 30, 40),
  Delta.lrv = 1.25,
  Delta.tv = 1.75,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  xbar_ng = NULL,
  xbar_go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  n.MC = 1000
)
```

Arguments

`mu.0.t`	prior mean for treatment group
`n.0.t`	prior effective sample size parameter for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`mu.0.c`	prior mean for control group
`n.0.c`	prior effective sample size parameter for control group

`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`xbar.t`	treatment mean
`s.t`	treatment sd
`n.t`	treatment sample size
`xbar.c`	control mean
`s.c`	control sd
`n.c`	control sample size
`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`xbar_ng`	Leave NULL to determine what is required or supply a value
`xbar_go`	Leave NULL to determine what is required or supply a value
`go.thresh`	If the predictive probability that study will conclude as 'Go' is larger than this threshold: Declare 'Interim go'.
`ng.thresh`	If the predictive probability that study will conclude as 'No-Go' is larger than this threshold: Declare 'Interim no-go'
`n.MC`	Monte Carlo simulation size

Value

A dataframe is returned

Examples

```
holdit <- return.ts.ng.int.req.df()
head(holdit)
```

`return.ts.ng.studyend.GNG.hm.df` *Return two-sample normal-gamma study end GNG heatmap data.frame*

Description

Return two-sample normal-gamma study end GNG heatmap data.frame

Usage

```
return.ts.ng.studyend.GNG.hm.df(
  mu.0.c = 0,
  alpha.0.c = 0.25,
  beta.0.c = 1,
  n.0.c = 1,
  mu.0.t = 0,
  alpha.0.t = 0.25,
  beta.0.t = 1,
  n.0.t = 1,
  xbar.c.low = -1,
  xbar.c.high = 1,
  s.c = 4,
  n.c = 40,
  s.t = 4,
  n.t = 40,
  npoints = 15,
  Delta.lrv = 1,
  Delta.tv = 1.5,
  tau.ng = 0.65,
  tau.lrv = 0.8,
  tau.tv = 0.1,
  n.MC = 1000,
  x_ng = NULL,
  x_go = NULL
)
```

Arguments

`mu.0.c`	prior mean for control group
`alpha.0.c`	prior alpha parameter for control group
`beta.0.c`	prior beta parameter for control group
`n.0.c`	prior effective sample size parameter for control group
`mu.0.t`	prior mean for treatment group
`alpha.0.t`	prior alpha parameter for treatment group
`beta.0.t`	prior beta parameter for treatment group
`n.0.t`	prior effective sample size parameter for treatment group
`xbar.c.low`	lower bound for control group sample mean grid
`xbar.c.high`	upper bound for control group sample mean grid
`s.c`	control standard deviation assumed
`n.c`	control sample size
`s.t`	treatment standard deviation assumed
`n.t`	treatment sample size
`npoints`	number of points to use in simulation

`Delta.lrv`	TPP Lower Reference Value aka Min TPP
`Delta.tv`	TPP Target Value aka Base TPP
`tau.ng`	threshold associated with No-Go
`tau.lrv`	threshold associated with Min TPP
`tau.tv`	threshold associated with Base TPP
`n.MC`	n for MC sampling
`x_ng`	xbar needed for no-go; leave null for standard rule
`x_go`	xbar needed for go; leave null for standard rule

Value

A data.frame is returned

Author(s)

Greg Cicconetti

Examples

```
my.ts.ng.studyend.GNG.hm.df <- return.ts.ng.studyend.GNG.hm.df()
head(my.ts.ng.studyend.GNG.hm.df)
```

`return.tte.int.data.req` *Title*

Description

Title

Usage

```
return.tte.int.data.req(
  m.con.prior = 0.001,
  m.trt.prior = 0.001,
  HR.prior = 1,
  ARatio = 1,
  interim.HR = c(0.8, 0.7, 0.65, 0.68, 0.7),
  interim.m = c(100, 200, 300, 400, 750),
  final.m = 1000,
  HR.tv = 0.7,
  HR.lrv = 0.9,
  tau.tv = 0.1,
  tau.lrv = 0.8,
```

```
  tau.ng = 0.65,
  HR.ng = NULL,
  HR.go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8,
  include_nogo = TRUE
)
```

Arguments

`m.con.prior`	prior number of control events
`m.trt.prior`	prior number of treatment events
`HR.prior`	prior estimate for HR
`ARatio`	randomization ratio
`interim.HR`	Interim HR
`interim.m`	Interim events
`final.m`	final events
`HR.tv`	TPP Target Value aka Base TPP
`HR.lrv`	TPP Lower Reference Value aka Max TPP (large HRs lead to No-Go)
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`HR.ng`	HR needed for ng; leave null for standard rule
`HR.go`	HR needed for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability
`ng.thresh`	no-go threshold for predictive probability
`include_nogo`	logical

Value

A data.frame is returned

Examples

```
holdit <- return.tte.int.data.req()
head(holdit)
```

`return.tte.pp` *Return time to event preditive probility*

Description

Return time to event preditive probility

Usage

```
return.tte.pp(
  m.con.prior = 50,
  m.trt.prior = 50,
  HR.prior = 0.845,
  ARatio = 1,
  interim.HR = seq(0.0025, 5, 0.0025),
  interim.m = c(428, 750, 1000),
  final.m = 1500,
  HR.tv = 0.139,
  HR.lrv = 0.139001,
  tau.tv = 0.1,
  tau.lrv = 0.8,
  tau.ng = 0.65,
  HR.ng = NULL,
  HR.go = NULL,
  go.thresh = 0.8,
  ng.thresh = 0.8
)
```

Arguments

`m.con.prior`	prior number of control events
`m.trt.prior`	prior number of treatment events
`HR.prior`	prior estimate for HR
`ARatio`	randomization ratio
`interim.HR`	Interim HR
`interim.m`	Interim events
`final.m`	final events
`HR.tv`	TPP Target Value aka Base TPP
`HR.lrv`	TPP Lower Reference Value aka Max TPP (large HRs lead to No-Go)
`tau.tv`	threshold associated with Base TPP
`tau.lrv`	threshold associated with Min TPP
`tau.ng`	threshold associated with No-Go
`HR.ng`	HR needed for ng; leave null for standard rule

`HR.go`	HR needed for go; leave null for standard rule
`go.thresh`	go threshold for predictive probability
`ng.thresh`	no-go threshold for predictive probability

Value

A data.frame is returned

Author(s)

Greg Cicconetti

Examples

```
{
return.tte.pp()
}
```

`SverdlovFunction` *Sverdlov's functions*

Description

Sverdlov's f function

Usage

```
f(a, b1, b2, c, x, y, t)

appel.hypgeom(a, b1, b2, c, x, y)

g(a, b, c, x, t)

gauss.hypgeom(a, b, c, x)

root(fun, left, right, tol)

r2beta(relation, n, a1, b1, a2, b2)

d2beta(relation, x, a1, b1, a2, b2)

p2beta(relation, approach, x, a1, b1, a2, b2, n = 1e+06)

q2beta(relation, a1, b1, a2, b2, alpha, tol = 10^(-5))

ci2beta(relation, method, a1, b1, a2, b2, alpha, left0, right0)
```

Arguments

`a`	a
`b1`	b1
`b2`	b2
`c`	c
`x`	x
`y`	y
`t`	t
`b`	b
`fun`	fun
`left`	left
`right`	right
`tol`	tol
`relation`	relation
`n`	n
`a1`	a1
`a2`	a2
`approach`	approach
`alpha`	alpha
`method`	method
`left0`	left0
`right0`	right0

Value

Various functions are offered.

References

Sverdlov O, Ryeznik Y, Wu S. Exact Bayesian Inference Comparing Binomial Proportions, With Application to Proof-of-Concept Clinical Trials. Therapeutic Innovation & Regulatory Science. 2015;49(1):163-174. doi:10.1177/2168479014547420

Index