Package 'mpower'

May 9, 2026

Type Package

Title Power Analysis via Monte Carlo Simulation for Correlated Data

Version 0.1.0

Description A flexible framework for power analysis using Monte Carlo simulation for settings in which considerations of the correlations between predictors are important. Users can set up a data generative model that preserves dependence structures among predictors given existing data (continuous, binary, or ordinal). Users can also generate power curves to assess the trade-offs between sample size, effect size, and power of a design. This package includes several statistical models common in environmental mixtures studies. For more details and tutorials, see Nguyen et al. (2022) <doi:10.48550/arXiv.2209.08036>.

License LGPL

Encoding UTF-8

LazyData true

Imports abind, boot, dplyr, doSNOW, foreach, ggplot2, MASS, magrittr, parallel, purrr, snow, sbgcop, rlang, reshape2, tibble, tidyr, tidyselect

RoxygenNote 7.2.1

Depends R (>= 3.5.0)

Suggests BMA, bkmr, bws, infinitefactor, knitr, NHANES, qgcomp, rmarkdown, rstan, testthat, openxlsx

NeedsCompilation no

Author Phuc H. Nguyen [aut, cre] (ORCID: <https://orcid.org/0000-0002-6206-0194>)

Maintainer Phuc H. Nguyen <phuc.nguyen.rcran@gmail.com>

Repository CRAN

Date/Publication 2022-09-21 08:50:05 UTC

Contents

`bkmr_wrapper` *Fits a BKMR model with significance criteria: PIP and group-wise PIP*

Description

Fits a BKMR model with significance criteria: PIP and group-wise PIP

Usage

```
bkmr_wrapper(y, x, args = list())
```

Arguments

y	A vector of outcome
x	A matrix of predictors
args	A list of arguments, see R function 'bkmr::kmbayes()'

Value

A list of vectors whose values are between 0 and 1

pip	PIP for component-wise selection or conditional (with-in group) PIP for hierarchical variable selection.
group_pip	PIP for group-specific selection.
time	elapsed time to fit the model.

Reference

Bobb JF, Henn BC, Valeri L, Coull BA (2018). "Statistical software for analyzing the health effects of multiple concurrent exposures via Bayesian kernel machine regression."Environ-mental Health,17(67).doi:10.1186/s12940-018-0413-y.

---

bma_wrapper *Fits a linear model with Bayesian model selection with significance criteria: PIP and posterior probability of nonzero coefficients being on one side of zero.*

Description

Fits a linear model with Bayesian model selection with significance criteria: PIP and posterior probability of nonzero coefficients being on one side of zero.

Usage

```
bma_wrapper(y, x, args = list())
```

Arguments

y	A vector of outcome
x	A matrix of predictors
args	A list of arguments see R function 'BMA::bic.glm()'.

Value

A list of vectors whose values are between 0 and 1

beta	The smaller posterior probability of the coefficients being to one side of zero: min(Pr(beta >0), Pr(beta<0)).
pip	PIP of the effect not being zero.
time	elapsed time to fit the model.

Reference

Raftery A, Hoeting J, Volinsky C, Painter I, Yeung KY (2021).BMA: Bayesian model averaging. R package version 3.18.15.

`bws_wrapper` *Fits a Bayesian weighted sums*

Description

Fits a Bayesian weighted sums

Usage

```
bws_wrapper(y, x, args = list(iter = 2000))
```

Arguments

`y`	A vector of outcome
`x`	A matrix of predictors
`args`	A list of arguments see R 'bws::bws()" function.

Value

A list

`beta`	The smaller posterior probability of the combined overall effect being to one side of zero: min(Pr(beta >0), Pr(beta<0)). The same for all predictor.
`weights`	The 95% CI of the contribution of each predictor to the overall effect. Between 0 and 1.
`time`	elapsed time to fit the model.

Reference

Hamra GB, MacLehose RF, Croen L, Kauffman EM, Newschaffer C (2021). "Bayesian weighted sums: a flexible approach to estimate summed mixture effects." International Journal of Environmental Research and Public Health, 18(4), 1373.

cor2partial *Convert a correlation matrix into a partial correlation matrix*

Description

Convert a correlation matrix into a partial correlation matrix

Usage

```
cor2partial(r)
```

Arguments

r	A correlation matrix

Value

A partial correlation matrix

cvine *Citation: Daniel Lewandowski, Dorota Kurowicka, Harry Joe, Generating random correlation matrices based on vines and extended onion method, Journal of Multivariate Analysis, Volume 100, Issue 9, 2009, Pages 1989-2001, ISSN 0047-259X, https://doi.org/10.1016/j.jmva.2009.04.008.*

Description

Citation: Daniel Lewandowski, Dorota Kurowicka, Harry Joe, Generating random correlation matrices based on vines and extended onion method, Journal of Multivariate Analysis, Volume 100, Issue 9, 2009, Pages 1989-2001, ISSN 0047-259X, https://doi.org/10.1016/j.jmva.2009.04.008.

Usage

```
cvine(d, alpha = 10, beta = 10, S = NULL, m = 100)
```

Arguments

d	Number of dimension
alpha	Parameter for Beta distribution
beta	Parameter for Beta distribution
S	A 'guess' of the correlation matrix
m	A number that indicates how much the random matrices vary from S

Value

A random positive-definite correlation matrix

estimate_snr *Monte Carlo approximation of the SNR*

Description

Monte Carlo approximation of the SNR

Usage

```
estimate_snr(ymod, xmod, m = 5000, R = 100)
```

Arguments

ymod	A OutcomeModel object
xmod	A MixtureModel object
m	Number of MC samples
R	Number of bootstrap replicates

Value

An estimate SNR and 95-percent CI.

fin_wrapper *Fits a Bayesian factor model with interactions*

Description

Fits a Bayesian factor model with interactions

Usage

```
fin_wrapper(y, x, args = list(nrun = 2000))
```

Arguments

y	A vector of outcome
x	A matrix of predictors
args	A list of arguments see R function ‘infinitefactor::interactionDL()‘ in ’infinitefactor’ package.

Value

A list of vectors whose values are between 0 and 1

beta	The smallest posterior probability of the coefficients being to one side of zero for either main effect or interaction: min(Pr(beta >0), Pr(beta<0)).
linear_beta	The smaller of posterior probability of the main effects being to one side of zero.
interact_beta	Same as linear_beta but for pair-wise interactions.
time	elapsed time to fit the model.

Reference

Ferrari F, Dunson DB (2020). "Bayesian factor analysis for inference on interactions."Journal of the American Statistical Association, 0(0), 1–12.

---

fit *Fits the model to given data and gets a list of significance criteria*

Description

Fits the model to given data and gets a list of significance criteria

Usage

```
fit(mod, x, y)
```

Arguments

mod	An InferenceModel object
x	A matrix of predictors
y	A vector of outcome

Value

A list of some of the following significance criteria:

beta	The smaller posterior probability of being to one side of zero for linear term, given either the main effect or interaction is non-zero. Applicable to 'bma', 'bws', 'fin', and 'mixselect' model.
interact_beta	Same as linear_beta but for pair-wise interactions. Applicable to 'fin' model.
pip	Posterior inclusion probability (PIP) of either a linear or non-linear effect. Applicable to 'bma', 'bkmr', and 'mixselect' model.
group_pip	PIP of either a linear or non-linear effect. Applicable to 'bkmr' model.
linear_pip	PIP of a linear effect. Applicable to 'mixselect' model.
gp_pip	PIP of a non-linear effect. Applicable to 'mixselect' model.
pval	The p-value of the combined effect, the same for all predictors. Applicable to 'glm', and 'qgc' model.
time	elapsed time to fit the model.

genx *Generates a matrix of n observations of p predictors*

Description

Generates a matrix of n observations of p predictors

Usage

```
genx(obj, n)
```

Arguments

obj	A MixtureModel object.
n	An integer, number of observations to generate.

Value

A (n x p) dataframe.

geny *Generates a vector of outcomes*

Description

Generates a vector of outcomes

Usage

```
geny(obj, x)
```

Arguments

obj	An OutcomeModel object
x	An (n x p) matrix of predictors

Value

An n-vector of outcomes

glm_wrapper *Fits a generalized linear model*

Description

Fits a generalized linear model

Usage

```
glm_wrapper(y, x, args = list())
```

Arguments

y	A vector of outcome
x	A matrix of predictors
args	A list of arguments see R ‘glm‘ function.

Value

A list

pval	The p-value of the linear main effect.
time	elapsed time to fit the model.

InferenceModel *Statistical model that returns significance criterion*

Description

This function creates a wrapper function for a statistical model and its applicable significance criterion. It finds relationships between a matrix of predictors and a vector of outcomes using the statistical model, and determines if the relationships are 'significant' according to pre-specified criterion for that model.

Usage

```
InferenceModel(model, name = NULL, ...)
```

Arguments

model	A string of the name of a built-in statistical model or a function implements a statistical model and returns a list of significance criteria for each predictor. Built-in options include 'bma', 'bkmr', 'mixselect', 'bws', 'qgc', 'fin', 'glm'.
name	A string, name of the statistical model. Default to string input of model.
...	Additional keyword arguments for the statistical model

Value

An InferenceModel object.

model	a function that takes matrices of predictors and outcomes and returns a list of significance criteria.
model_name	a string.

Examples

```
imod <- mpower::InferenceModel(model = 'glm', family = 'gaussian',
formula = y ~ Poverty*(poly(Age, 2) + HHIncome + HomeOwn + Education))
```

MixtureModel *Correlated predictors generator*

Description

This function creates a generative model for the correlated, mixed-scale predictors.

Usage

```
MixtureModel(
  method = "estimation",
  data = NULL,
  G = NULL,
  m = 100,
  nudge = 1e-09,
  sbg_args = list(nsamp = 1000),
  cvine_marginals = list(),
  cvine_dtypes = list(),
  resamp_prob = NULL
)
```

Arguments

method	A string, one of the three options "resampling", "estimation", or "cvine". Default is "estimation". See Details.
data	A dataframe or matrix, required for resampling" and "estimation" method.
G	A guesstimate pairwise correlation matrix for "cvine" method. See Details.
m	A positive number indicating uncertainty in the guesstimate G, larger means more uncertainty. Default is 100.
nudge	A number, default 10e-10 to add to the diagonal of the covariance matrix for numerical stability.
sbg_args	A list of named arguments, except Y, for function ‘sbgcop.mcmc()‘.
cvine_marginals	A named list describing the univariate distribution of each predictor. See Details.

cvine_dtypes	A named list describing the data type of each variable.
resamp_prob	A vector of sampling probability for each observation in data. Must sums to 1.

Value

A MixtureModel object.

Details

There are three methods to generate data:

1. Resampling: if we have enough data of the predictors, we can resample to get realistic joint distributions and dependence among them.

2. Estimation: if we have a small sample from, for example, a pilot study, we can sample from a semi-parametric copula model (Hoff 2007) after learning the dependence and univariate marginals of the predictors.

3. C-vine: if no pilot data exists, we can still set rough guesstimate of the dependence and univariate marginals. The C-vine algorithm (Joe 2006) generates positive semi-definite correlation matrix given the guesstimate G. The guesstimate G is a symmetric p x p matrix whose ij-th item is between -1 and 1 and is the guesstimate correlation between predictor ith and jth. G doesn't need to be a valid correlation matrix. The method works well when values in G are not extreme (i.e., 0.999, -0.999). Built-in functions for univariate marginals include: 'qbeta' , 'qbinom', 'qcauchy', 'qchisq', 'qexp', 'qf', 'qgamma', 'qgeom', 'qhyper', 'qlogis', 'qlnorm', 'qmultinom', 'qnbinom', 'qnorm', 'qpois', 'qt', 'qunif', 'qweibull'.

References

Hoff P (2007). 'Extending the rank likelihood for semiparametric copula estimation.' Ann. Appl. Stat, 1(1), 265-283.

Joe H (2006). "Generating random correlation matrices based on partial correlations."Journal of Multivariate Analysis, 97, 2177-2189.

Examples

```
data("nhanes1518")
xmod <- mpower::MixtureModel(nhanes1518, method = "resampling")
```

---

mplot *Visualize marginals and Gaussian copula correlations of simulated data*

Description

Visualize marginals and Gaussian copula correlations of simulated data

Usage

```
mplot(obj, split = TRUE)
```

Arguments

obj	A MixtureModel object.
split	A logical, whether to display numbers on half of the covariance matrix.

Value

A 'ggplot2' graphics.

---

mpower *mpower: Power analysis using Monte Carlo for correlated predictors.*

Description

This package provides tools to set up simulations for power calculation

---

nhanes1518 *NHANES data from 2015-2016 and 2017-2018 cycles*

Description

Combined NHANES data from the 2015-2016 and 2017-2018 cycles The weights have been adjusted according to https://wwwn.cdc.gov/nchs/nhanes/tutorials/module3.aspx

Usage

```
nhanes1518
```

Format

Data with the following variables:

SEQN Respondent sequence number

WTINT4YR Full sample 4 year interview weight

WTMEC4YR Full sample 4 year MEC exam weight

WTSB4YR Environmental B 4-year weights

RIDSTATR Interview/Examination status

RIAGENDR Gender of the participant

RIDAGEYR Age in years of the participant at the time of screening. Individuals 80 and over are top-coded at 80 years of age

INDFMPIR A ratio of family income to poverty guidelines

RIDRETH1 Race/Hispanic origin

INDHHIN2 Total household income (reported as a range value in dollars)

BMXBMI Body Mass Index (kg/m**2)

BMXWAIST Waist Circumference (cm)

BMXWT Weight (kg)

BMXHT Standing Height (cm)

URXUCR Creatinine, urine (mg/dL)

URXCNP MCNP Mono(carboxyisononyl) phthalate (ng/mL), LLOD = 0.2

URXCOP MCOP Mono(carboxyisoctyl) phthalate (ng/mL), LLOD = 0.3

URXECP MECPP Mono-2-ethyl-5-carboxypentyl phthalate (ng/mL), LLOD = 0.4

URXHIBP MHIBP phthalate (ng/mL), LLOD = 0.4

URXMBP MnBP Mono-n-butyl phthalate (ng/mL), LLOD = 0.4

URXMC1 MCPP Mono-(3-carboxypropyl) phthalate (ng/mL), LLOD = 0.4

URXMCOH MCOCH phthalate (ng/mL), LLOD = 0.5

URXMEP MEP Mono-ethyl phthalate (ng/mL), LLOD = 1.2

URXMHBP Mono-3-hydroxy-n-butyl phthalate (ng/mL), LLOD = 0.4

URXMHH MEHHP Mono-(2-ethyl-5-hydroxyhexyl) phthalate (ng/mL), LLOD = 0.4

URXMHNC Cyclohexane 1,2-dicarboxylic acid monohydroxy isononyl ester (ng/mL), LLOD = 0.4

URXMHP MEHP Mono-(2-ethyl)-hexyl phthalate (ng/mL), LLOD = 0.8

URXMIB MiBP Mono-isobutyl phthalate (ng/mL), LLOD = 0.8

URXMNP MCNP Mono-isononyl phthalate (ng/mL), LLOD = 0.9

URXMOH MEOHP Mono-(2-ethyl-5-oxohexyl) phthalate (ng/mL), LLOD = 0.2

URXMZP MBzP Mono-benzyl phthalate (ng/mL), LLOD = 0.3

URDCNPLC, URDCOPLC, URDECPLC, URDHIBLC, URDMBPLC, URDMC1LC, URDMCOLC, URDMEPLC, Phthalates comment code for whether the measurement is under the limit of detection

Source

Detailed documentation of the phthalates variables can be found here:

- https://wwwn.cdc.gov/nchs/nhanes/search/default.aspx
- https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/PHTHTE_I.htm
- https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/PHTHTE_J.htm

OutcomeModel *Outcome generator*

Description

This function creates a generative model of the outcome given a matrix of predictors.

Usage

```
OutcomeModel(f, family = "gaussian", sigma = 1, f_args = list())
```

Arguments

f	A string that describes the relationships between the predictors and outcome or a function that takes an input matrix and returns a vector of outcome: $E(y\|x) = g(f(x))$ where g is a link function that depends on the family argument.
family	A string, 'gaussian', 'binomial', or 'poisson' for continuous, binary, or count outcomes.
sigma	A number, Gaussian noise standard deviation if applicable.
f_args	A named list of additional arguments to f

Value

An OutcomeModel object. Attributes:

f	mean function.
sigma	a number for the Gaussian observation noise.
family	a string 'gaussian' or 'binomial'.

Examples

```
# Define BMI as a ratio of weight and height plus random Gaussian error with standard deviation 1.
bmi_model <- mpower::OutcomeModel(f = 'weight/(height^2)', sigma = 1, family = 'gaussian')
```

partial *Partial correlations between elements in x and elements in y*

Description

Partial correlations between elements in x and elements in y

Usage

```
partial(r, x, y)
```

Arguments

r	A correlation matrix
x	A vector of indices
y	A vector of indices

Value

A partial correlation matrix

---

plot_summary *Plot summaries of power simulation*

Description

Plot summaries of power simulation

Usage

```
plot_summary(sim, crit, thres, digits = 3, how = "greater")
```

Arguments

sim	A Sim or a SimCurve object, output from ‘sim_power()‘ or ‘sim_curve()‘.
crit	A string specifying the significance criteria.
thres	A number or vector of numbers specifying the thresholds of "significance".
digits	An integer for the number of decimal points to display.
how	A string, whether to compare the criterion ’greater’ or ’lesser’ than the threshold.

Value

A ’ggplot2’ graphics.

qgcomp_lin_wrapper *Fits a linear Quantile G-Computation model with no interactions*

Description

Fits a linear Quantile G-Computation model with no interactions

Usage

```
qgcomp_lin_wrapper(y, x, args = list())
```

Arguments

y	A vector of outcome
x	A matrix of predictors
args	A list of arguments see R function 'qgcomp::qgcomp.noboot()'.

Value

A list

pval	The p-value of the combined effect, the same for all predictors.
pos_weights	Positive weights. See 'qgcomp' package.
neg_weights	Negative weights. See 'qgcomp' package.
time	elapsed time to fit the model.

Reference

Keil AP, Buckley JP, O'Brien KM, Ferguson KK, Zhao S, White AJ (2020). "A Quantile-based g-computation approach to addressing the effects of exposure mixtures."Environmental Health Perspectives, 128(4).

qmultinom *Quantile function for the multinomial distribution, size = 1*

Description

Quantile function for the multinomial distribution, size = 1

Usage

```
qmultinom(p, probs)
```

Arguments

p	A quantile.
probs	A vector of probabilities for each level.

Value

Gives the quantile function

---

rsq2snr *Convert R-squared value to the SNR*

Description

Convert R-squared value to the SNR

Usage

```
rsq2snr(r)
```

Arguments

r	R-squared value

---

scale_f *Rescale the mean function of an OutcomeModel to meet a given SNR*

Description

Rescale the mean function of an OutcomeModel to meet a given SNR

Usage

```
scale_f(snr, ymod, xmod, m = 5000)
```

Arguments

snr	A SNR
ymod	A OutcomeModel object to modify
xmod	A MixtureModel object
m	Number of MC samples to estimate the SNR of a proposed noise variance

Value

A new OutcomeModel object

`scale_sigma` *Rescale the noise variance of a Gaussian OutcomeModel to meet a given SNR*

Description

Rescale the noise variance of a Gaussian OutcomeModel to meet a given SNR

Usage

```
scale_sigma(snr, ymod, xmod, m = 5000)
```

Arguments

`snr`	A SNR
`ymod`	A OutcomeModel object to modify
`xmod`	A MixtureModel object
`m`	Number of MC samples to estimate the SNR of a proposed noise variance

Value

A new OutcomeModel object

`set_value` *This function updates values in an OutcomeModel object*

Description

This function updates values in an OutcomeModel object

Usage

```
set_value(obj, name, value)
```

Arguments

`obj`	An OutcomeModel object
`name`	A string for name of the attribute to be changed
`value`	An appropriate data type

`sim_curve` *Power curve using Monte Carlo simulation*

Description

This function can be used to create power curves by calling sim_power() on combinations of many sample sizes and signal-to-noise ratio (SNR).

Usage

```
sim_curve(
  xmod,
  ymod,
  imod,
  s = 100,
  n = 100,
  cores = 1,
  file = NULL,
  errorhandling = "stop",
  snr_iter = 10000,
  cluster_export = c()
)
```

Arguments

`xmod`	A MixtureModel object.
`ymod`	One or a list of OutcomeModel object(s).
`imod`	An InferenceModel object.
`s`	An integer for the number of Monte Carlo simulations.
`n`	An integer or a vector of sample sizes.
`cores`	An integer for the number of processing cores. When cores > 1, parallelism is automatically applied.
`file`	A string, a file name with no extension to write samples to periodically. By default write to an RDS file.
`errorhandling`	A string "remove", "stop", or "pass". If an error occurs in any iteration, remove that iteration (remove), return the error message verbatim in the output (pass), or terminate the loop (stop). Default is "remove". See R package 'foreach' for more details.
`snr_iter`	An integer for number of Monte Carlo samples to estimate SNR.
`cluster_export`	A vector of functions to pass to the parallel-processing clusters.

Value

A SimCurve object with the following attributes:

s	a number of simulations.
snr	a real number or array of real numbers for SNR of each OutcomeModel.
n	a number or vector of sample sizes.
xmod	the MixtureModel used.
ymod	the OutcomeModel used.
imod	the InferenceModel used.
sims	a list of simulation output matrices.

Examples

```
data("nhanes1518")
chems <- c("URXCNP", "URXCOP", "URXECP", "URXHIBP", "URXMBP", "URXMC1",
"URXMCOH", "URXMEP","URXMHBP", "URXMHH", "URXMHNC", "URXMHP", "URXMIB",
"URXMNP", "URXMOH", "URXMZP")
chems_mod <- mpower::MixtureModel(nhanes1518[, chems], method = "resampling")
bmi_mod <- mpower::OutcomeModel(f = "0.2*URXCNP + 0.15*URXECP +
0.1*URXCOP*URXECP", family = "binomial")
logit_mod <- mpower::InferenceModel(model = "glm", family = "binomial")
logit_out <- mpower::sim_curve(xmod=chems_mod, ymod=bmi_mod, imod=logit_mod,
s=20, n=c(500, 1000), cores=2, snr_iter=1000)
logit_df <- summary(logit_out, crit="pval", thres=0.05, how="lesser")
plot_summary(logit_out, crit="pval", thres=0.05, how="lesser")
```

---

sim_power *Power analysis using Monte Carlo simulation*

Description

Power analysis using Monte Carlo simulation

Usage

```
sim_power(
  xmod,
  ymod,
  imod,
  s = 100,
  n = 100,
  cores = 1,
  file = NULL,
  errorhandling = "stop",
  snr_iter = 10000,
  cluster_export = c()
)
```

Arguments

xmod	A MixtureModel object.
ymod	An OutcomeModel object.
imod	An InferenceModel object.
s	An integer for number of Monte Carlo simulations.
n	An integer for sample size in each simulation.
cores	An integer for number of processing cores. When cores > 1, parallelism is automatically applied.
file	A string, a file name with no extension to write samples to periodically. By default write to an RDS file.
errorhandling	A string "remove", "stop", or "pass". If an error occurs in any iteration, remove that iteration (remove), return the error message verbatim in the output (pass), or terminate the loop (stop). Default is "remove". See R package 'foreach' for more details.
snr_iter	An integer for number of Monte Carlo samples to estimate SNR
cluster_export	A vector of functions to pass to the parallel-processing clusters

Value

A PowerSim object. Attributes:

s	a number of simulations.
snr	a real number for SNR of the OutcomeModel.
n	a number of sample sizes.
xmod	the MixtureModel used.
ymod	the OutcomeModel used.
imod	the InferenceModel used.
sims	an output matrices.

Examples

```
data("nhanes1518")
chems <- c("URXCNP", "URXCOP", "URXECP", "URXHIBP", "URXMBP", "URXMC1",
"URXMCOH", "URXMEP","URXMHBP", "URXMHH", "URXMHNC", "URXMHP", "URXMIB",
"URXMNP", "URXMOH", "URXMZP")
chems_mod <- mpower::MixtureModel(nhanes1518[, chems], method = "resampling")
bmi_mod <- mpower::OutcomeModel(f = "0.2*URXCNP + 0.15*URXECP +
0.1*URXCOP*URXECP", family = "binomial")
logit_mod <- mpower::InferenceModel(model = "glm", family = "binomial")
logit_out <- mpower::sim_power(xmod=chems_mod, ymod=bmi_mod, imod=logit_mod,
s=100, n=2000, cores=2, snr_iter=2000)
logit_df <- summary(logit_out, crit="pval", thres=0.05, how="lesser")
plot_summary(logit_out, crit="pval", thres=0.05, how="lesser")
```

summary *Tabular summaries of power simulation*

Description

Tabular summaries of power simulation

Usage

```
summary(sim, crit, thres, digits = 3, how = "greater")
```

Arguments

sim	A Sim or a SimCurve object, output from ‘sim_power()‘ or ‘sim_curve()‘.
crit	A string specifying the significance criteria.
thres	A number or vector of numbers specifying the thresholds of "significance".
digits	An integer for the number of decimal points to display.
how	A string, whether to compare the criterion ’greater’ or ’lesser’ than the threshold.

Value

A data.frame summary of power for each predictor for each combination of thresholds, sample size, signal-to-noise ratios.

Index