

Package ‘Rgof’

October 8, 2024

Title 1d Goodness of Fit Tests

Version 2.1.0

Description Routines that allow the user to run a large number of goodness-of-fit tests. It allows for data to be continuous or discrete. It includes routines to estimate the power of the tests and display them as a power graph.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.1

LinkingTo Rcpp

Imports Rcpp, parallel, ggplot2, stats

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation yes

Author Wolfgang Rolke [aut, cre] (<<https://orcid.org/0000-0002-3514-726X>>)

Maintainer Wolfgang Rolke <wolfgang.rolke@upr.edu>

Repository CRAN

Date/Publication 2024-10-08 19:20:02 UTC

Contents

check.functions	2
chi_power_cont	2
chi_power_disc	4
chi_test_cont	5
chi_test_disc	6
gof_power	7
gof_power_cont	9
gof_power_disc	10
gof_test	11
gof_test_adjusted_pvalue	13
gof_test_cont	15

gof_test_cont_adj	16
gof_test_disc	17
gof_test_disc_adj	18
make_bins_cont	19
make_bins_disc	20
plot_power	21
signif.digits	21

Index	22
--------------	-----------

check.functions	<i>This function checks whether the inputs have the correct format</i>
-----------------	--

Description

This function checks whether the inputs have the correct format

Usage

```
check.functions(pnull, rnull, phat = function(x) -99, vals, x)
```

Arguments

pnull	cdf under the null hypothesis
rnull	routine to generate data under the null hypothesis
phat	=function(x) -99, function to estimate parameters from the data, or -99
vals	vector of discrete values
x	data

chi_power_cont	<i>This function finds the power of various chi-square tests for continuous data</i>
----------------	--

Description

This function finds the power of various chi-square tests for continuous data

Usage

```

chi_power_cont(
  pnull,
  ralt,
  param_alt,
  qnull = NA,
  phat = function(x) -99,
  w = function(x) -99,
  alpha = 0.05,
  Range = c(-99999, 99999),
  B = 1000,
  nbins = c(50, 10),
  rate = 0,
  minexpcount = 5,
  ChiUsePhat = TRUE
)

```

Arguments

pnull	function to find cdf under null hypothesis
ralt	function to generate data under alternative hypothesis
param_alt	vector of parameter values for distribution under alternative hypothesis
qnull	=NA function to find quantiles under null hypothesis, if available
phat	=function(x) -99, function to estimate parameters
w	=function(x) -99, optional weight function
alpha	=0.05, the level of the hypothesis test
Range	=c(-99999, 99999) limits of possible observations, if any
B	=1000 number of simulation runs to find power
nbins	=c(50,10), number of bins for chi square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
minexpcount	=5 minimal expected bin count required
ChiUsePhat	=TRUE, if TRUE param is estimated parameters and no minimization is used

Value

A numeric matrix of power values.

chi_power_disc	<i>This function finds the power of various chi-square tests for continuous data</i>
----------------	--

Description

This function finds the power of various chi-square tests for continuous data

Usage

```
chi_power_disc(
  pnull,
  ralt,
  param_alt,
  phat = function(x) -99,
  alpha = 0.05,
  B = 1000,
  nbins = c(50, 10),
  rate = 0,
  minexpcount = 5,
  ChiUsePhat = TRUE
)
```

Arguments

pnull	function to find cdf under null hypothesis
ralt	function to generate data under alternative hypothesis
param_alt	vector of parameter values for distribution under alternative hypothesis
phat	=function(x) -99, routine to estimate parameters
alpha	=0.05, the level of the hypothesis test
B	=1000 number of simulation runs to find power
nbins	=c(50,10), number of bins for chi square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
minexpcount	=5 minimal expected bin count required
ChiUsePhat	= TRUE, should chi square use minimum chi square method?

Value

A numeric matrix of power values.

chi_test_cont	<i>This function performs a number of chi-square gof tests for continuous data</i>
---------------	--

Description

This function performs a number of chi-square gof tests for continuous data

Usage

```
chi_test_cont(
  x,
  pnull,
  w = function(x) -99,
  phat = function(x) -99,
  qnull = NA,
  nbins = c(50, 10),
  rate = 0,
  Range = c(-99999, 99999),
  minexpcount = 5,
  ChiUsePhat = TRUE,
  allbins
)
```

Arguments

x	data set
pnull	cdf under the null hypothesis
w	function to find weights of observations, returns -99 if data is unweighted
phat	=function(x) -99, estimated parameters, or starting values of multi-D minimum chi square minimization, or -99 if no estimation is done
qnull	=NA quantile function, if available
nbins	=c(50, 10) number of bins for chi-square tests
rate	=0, rate of Poisson if sample size is random
Range	=c(-99999, 99999) limits of possible observations, if any
minexpcount	=5 minimal expected bin count required
ChiUsePhat	=TRUE, if TRUE param is estimated parameters and no minimization is used
allbins	set of bins to use

Value

A numeric matrix of test statistics, degrees of freedom and p.values

chi_test_disc	<i>This function performs a number of chi-square gof tests for continuous data</i>
---------------	--

Description

This function performs a number of chi-square gof tests for continuous data

Usage

```
chi_test_disc(
  x,
  pnull,
  phat = function(x) -99,
  nbins = c(50, 10),
  rate = 0,
  minexpcount = 5,
  ChiUsePhat = TRUE,
  allbins
)
```

Arguments

x	data set
pnull	cdf under the null hypothesis
phat	=function(x) -99, function to estimate parameters, or starting values of multi-D minimum chi square minimization, or -99 if no parameters are estimated
nbins	=c(50, 10) number of bins for chi-square tests
rate	=0, rate of Poisson if sample size is random
minexpcount	=5 minimal expected bin count required
ChiUsePhat	= TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.
allbins	set of bins to use

Value

A numeric matrix of test statistics, degrees of freedom and p.values

gof_power

*Find the power of various gof tests for continuous data.***Description**

Find the power of various gof tests for continuous data.

Usage

```
gof_power(
  pnull,
  vals = NA,
  rnull,
  ralt,
  param_alt,
  w = function(x) -99,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  alpha = 0.05,
  Range = c(-Inf, Inf),
  B = c(1000, 1000),
  nbins = c(50, 10),
  rate = 0,
  maxProcessors,
  minexpcount = 5,
  ChiUsePhat = TRUE
)
```

Arguments

pnull	function to find cdf under null hypothesis
vals	=NA, values of rv, if data is discrete, NA if data is continuous
rnull	function to generate data under null hypothesis
ralt	function to generate data under alternative hypothesis
param_alt	vector of parameter values for distribution under alternative hypothesis
w	(Optional) function to calculate weights, returns -99 if no weights
phat	=function(x) -99 function to estimate parameters from the data, or -99
TS	user supplied function to find test statistics
TSextra	=NA, list provided to TS
alpha	=0.05, the level of the hypothesis test
Range	=c(-Inf, Inf) limits of possible observations, if any
B	=c(1000, 1000), number of simulation runs to find power and null distribution

nbins =c(100,10), number of bins for chi square tests.
rate =0 rate of Poisson if sample size is random, 0 if sample size is fixed
maxProcessors maximum of number of processors to use, 1 if no parallel processing is needed or number of cores-1 if missing
minexpcount =5 minimal expected bin count required
ChiUsePhat = TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.

Value

A numeric matrix of power values.

Examples

```

# Power of tests when null hypothesis specifies the standard normal distribution but
# true data comes from a normal distribution with mean different from 0.
pnull = function(x) pnorm(x)
rnull = function() rnorm(50)
ralt = function(mu) rnorm(50, mu)
TSextra = list(qnull=function(x) qnorm(x))
gof_power(pnull, NA, rnull, ralt, c(0.25, 0.5), TSextra=TSextra, B=c(500, 500))
# Power of tests when null hypothesis specifies normal distribution and
# mean and standard deviation are estimated from the data.
# Example is not run because it takes several minutes.
# true data comes from a normal distribution with mean different from 0.
pnull = function(x, p=c(0, 1)) pnorm(x, p[1], ifelse(p[2]>0.001, p[2], 0.001))
rnull = function(p=c(0, 1)) rnorm(50, p[1], ifelse(p[2]>0.001, p[2], 0.001))
phat = function(x) c(mean(x), sd(x))
TSextra = list(qnull = function(x, p=c(0, 1)) qnorm(x, p[1],
  ifelse(p[2]>0.001, p[2], 0.001)))
gof_power(pnull, NA, rnull, ralt, c(0, 1), phat=phat, TSextra=TSextra,
  B=c(200, 200), maxProcessor=2)
# Power of tests when null hypothesis specifies Poisson rv with rate 100 and
# true rate is 100.5
vals = 0:250
pnull = function() ppois(0:250, 100)
rnull =function () table(c(0:250, rpois(1000, 100)))-1
ralt =function (p) table(c(0:250, rpois(1000, p)))-1
gof_power(pnull, vals, rnull, ralt, param_alt=100.5, B=c(500,500))
# Power of tests when null hypothesis specifies a Binomial n=10 distribution
# with the success probability estimated
vals = 0:10
pnull=function(p) pbinom(0:10, 10, ifelse(0<p&p<1, p, 0.001))
rnull=function(p) table(c(0:10, rbinom(1000, 10, ifelse(0<p&p<1, p, 0.001))))-1
ralt=function(p) table(c(0:10, rbinom(1000, 10, p)))-1
phat=function(x) mean(rep(0:10,x))/10
gof_power(pnull, vals, rnull, ralt, c(0.5, 0.6), phat=phat,
  B=c(200, 200), maxProcessor=2)

```

gof_power_cont	<i>Find the power of various gof tests for continuous data.</i>
----------------	---

Description

Find the power of various gof tests for continuous data.

Usage

```
gof_power_cont(
  pnull,
  rnull,
  ralt,
  param_alt,
  w = function(x) -99,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  alpha = 0.05,
  Range = c(-Inf, Inf),
  B = c(1000, 1000),
  nbins = c(100, 10),
  rate = 0,
  maxProcessors,
  minexpcount = 5,
  ChiUsePhat = TRUE
)
```

Arguments

pnull	function to find cdf under null hypothesis
rnull	function to generate data under null hypothesis
ralt	function to generate data under alternative hypothesis
param_alt	vector of parameter values for distribution under alternative hypothesis
w	(Optional) function to calculate weights, returns -99 if no weights
phat	=function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated
TS	user supplied function to find test statistics, if any
TSextra	=NA, list provided to TS
alpha	=0.05, the level of the hypothesis test
Range	=c(-Inf, Inf) limits of possible observations, if any
B	=c(1000, 1000), number of simulation runs to find power and null distribution
nbins	=c(100,10), number of bins for chi square tests.

rate =0 rate of Poisson if sample size is random, 0 if sample size is fixed

maxProcessors maximum of number of processors to use, 1 if no parallel processing is needed or number of cores-1 if missing

minexpcount =5 minimal expected bin count required

ChiUsePhat =TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.

Value

A numeric matrix of power values.

gof_power_disc *Find the power of various gof tests for discrete data.*

Description

Find the power of various gof tests for discrete data.

Usage

```
gof_power_disc(
  pnull,
  rnull,
  vals,
  ralt,
  param_alt,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  alpha = 0.05,
  B = c(1000, 1000),
  nbins = c(100, 10),
  rate = 0,
  maxProcessors,
  minexpcount = 5,
  ChiUsePhat = TRUE
)
```

Arguments

pnull cumulative distribution function under the null hypothesis

rnull a function to generate data under null hypothesis

vals values of discrete rv.

ralt function to generate data under alternative hypothesis

param_alt vector of parameter values for distribution under alternative hypothesis

phat	=function(x) -99, function to estimate parameters from the data, -99 if no parameters are estimated
TS	user supplied function to find test statistics, if any
TSextra	=NA, list passed to TS, if desired
alpha	=0.05, the level of the hypothesis test
B	=c(1000, 1000), number of simulation runs to find power and null distribution
nbins	=c(100, 10) number of bins for chi square tests
rate	rate of Poisson if sample size is random
maxProcessors	maximum of number of processors to use, 1 if no parallel processing is needed or number of cores-1 if missing
minexpcount	=5 minimal number of expected counts in each bin for chi square tests
ChiUsePhat	= TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.

Value

A numeric matrix of power values.

gof_test	<i>This function performs a number of gof tests</i>
----------	---

Description

This function performs a number of gof tests

Usage

```
gof_test(
  x,
  vals = NA,
  pnull,
  rnull,
  w = function(x) -99,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  nbins = c(50, 10),
  rate = 0,
  Range = c(-Inf, Inf),
  B = 5000,
  minexpcount = 5,
  ChiUsePhat = TRUE,
  maxProcessors = 1,
  doMethods = "all"
)
```

Arguments

x	data set
vals	=NA, values of discrete RV, or NA if data is continuous
pnull	cdf under the null hypothesis
rnull	routine to generate data under the null hypothesis
w	(Optional) function to calculate weights, returns -99 if no weights
phat	=function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated
TS	user supplied function to find test statistics, if any
TSextra	=NA, list passed to TS, if desired, or NA
nbins	=c(100, 10) number of bins for chi-square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
Range	=c(-Inf, Inf) limits of possible observations, if any, for chi-square tests
B	=5000 number of simulation runs
minexpcount	=5 minimal expected bin count required
ChiUsePhat	= TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.
maxProcessors	=1, number of processors to use in parallel processing.
doMethods	Methods to include in tests

Value

A list with vectors of test statistics and p.values

Examples

```
# Tests to see whether data comes from a standard normal distribution.
pnull = function(x) pnorm(x)
rnull = function() rnorm(100)
x = rnorm(100)
gof_test(x, NA, pnull, rnull)
# Tests to see whether data comes from a normal distribution with standard deviation 1
# and the mean estimated.
pnull=function(x, m) pnorm(x, m)
rnull=function(m) rnorm(100, m)
TSextra = list(qnull=function(x, m=0) qnorm(x, m),
              pnull=function(x, m=0) pnorm(x, m), phat=function(x) mean(x))
phat=function(x) mean(x)
x = rnorm(100, 1, 2)
gof_test(x, NA, pnull, rnull, phat=phat, TSextra=TSextra)
# Tests to see whether data comes from a binomial (10, 0.5) distribution.
vals=0:10
pnull = function() pbinom(0:10, 10, 0.5)
rnull = function() table(c(0:10, rbinom(1000, 10, 0.5)))-1
x = rnull()
```

```

gof_test(x, vals, pnull, rnull, doMethods="all")
# Tests to see whether data comes from a binomial distribution with
# the success probability estimated from the data.
pnull = function(p=0.5) pbinom(0:10, 10, ifelse(p>0&&p<1, p, 0.001))
rnull = function(p=0.5) table(c(0:10, rbinom(1000, 10,
      ifelse(p>0&&p<1, p, 0.001))))-1
phat=function(x) mean(rep(0:10,x))/10
gof_test(x, vals, pnull, rnull, phat=phat)

```

gof_test_adjusted_pvalue

This function performs a number of gof tests and finds the adjusted p value for the combined test

Description

This function performs a number of gof tests and finds the adjusted p value for the combined test

Usage

```

gof_test_adjusted_pvalue(
  x,
  vals = NA,
  pnull,
  rnull,
  w = function(x) -99,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  nbins = c(50, 10),
  rate = 0,
  Range = c(-Inf, Inf),
  B = 5000,
  minexpcount = 5,
  ChiUsePhat = TRUE,
  doMethods
)

```

Arguments

x	data set
vals	=NA, values of discrete RV, or NA if data is continuous
pnull	cdf under the null hypothesis
rnull	routine to generate data under the null hypothesis
w	(Optional) function to calculate weights, returns -99 if no weights

phat	=function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated
TS	user supplied function to find test statistics, if any
TSextra	=NA, list passed to TS, if desired, or NA
nbins	=c(100, 10) number of bins for chi-square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
Range	=c(-Inf, Inf) limits of possible observations, if any, for chi-square tests
B	=c(5000,1000) number of simulation runs for individual and for adjusted p values
minexpcount	=5 minimal expected bin count required
ChiUsePhat	= TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.
doMethods	Methods to include in tests

Value

None

Examples

```
# Tests to see whether data comes from a standard normal distribution.
pnull = function(x) pnorm(x)
rnull = function() rnorm(100)
x = rnorm(100)
gof_test_adjusted_pvalue(x, NA, pnull, rnull, B=c(1000, 200))
# Tests to see whether data comes from a normal distribution with standard deviation 1
# and the mean estimated.
pnull=function(x, m) pnorm(x, m)
rnull=function(m) rnorm(100, m)
TSextra = list(qnull=function(x, m=0) qnorm(x, m),
              pnull=function(x, m=0) pnorm(x, m), phat=function(x) mean(x))
phat=function(x) mean(x)
x = rnorm(100, 1, 2)
gof_test_adjusted_pvalue(x, NA, pnull, rnull, phat=phat, TSextra=TSextra, B=c(1000, 200))
# Tests to see whether data comes from a binomial (10, 0.5) distribution.
vals=0:10
pnull = function() pbinom(0:10, 10, 0.5)
rnull = function() table(c(0:10, rbinom(1000, 10, 0.5)))-1
x = rnull()
gof_test_adjusted_pvalue(x, vals, pnull, rnull, B=c(1000, 200))
# Tests to see whether data comes from a binomial distribution with
# the success probability estimated from the data.
pnull = function(p=0.5) pbinom(0:10, 10, ifelse(p>0&&p<1, p, 0.001))
rnull = function(p=0.5) table(c(0:10, rbinom(1000, 10,
              ifelse(p>0&&p<1, p, 0.001))))-1
phat=function(x) mean(rep(0:10,x))/10
gof_test_adjusted_pvalue(x, vals, pnull, rnull, phat=phat, B=c(1000, 200))
```

gof_test_cont

*This function performs a number of gof tests for continuous data***Description**

This function performs a number of gof tests for continuous data

Usage

```
gof_test_cont(
  x,
  pnull,
  rnull,
  w = function(x) -99,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  nbins = c(50, 10),
  rate = 0,
  Range = c(-Inf, Inf),
  B = 5000,
  minexpcount = 5,
  ChiUsePhat = TRUE,
  maxProcessors = 1,
  doMethods = "all"
)
```

Arguments

x	data set
pnull	cdf under the null hypothesis
rnull	routine to generate data under the null hypothesis
w	(Optional) function to calculate weights, returns -99 if no weights
phat	=function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated
TS	user supplied function to find test statistics, if any
TSextra	=NA, list passed to TS, if desired
nbins	=c(50, 10) number of bins for chi-square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
Range	=c(-Inf, Inf) limits of possible observations, if any, for chi-square tests
B	=5000 number of simulation runs
minexpcount	=5 minimal expected bin count required
ChiUsePhat	=TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.

maxProcessors =1, number of processors to use in parallel processing. If missing single processor is used.

doMethods Methods to include in tests

Value

A list with vectors of test statistics and p.values

gof_test_cont_adj *This function performs a number of gof tests for continuous data and finds the adjusted p value*

Description

This function performs a number of gof tests for continuous data and finds the adjusted p value

Usage

```
gof_test_cont_adj(
  x,
  pnull,
  rnull,
  w = function(x) -99,
  phat = function(x) 0,
  TS,
  TSextra = NA,
  nbins = c(50, 10),
  rate = 0,
  Range = c(-Inf, Inf),
  B = c(5000, 1000),
  minexpcount = 5,
  ChiUsePhat = TRUE,
  doMethods = c("W", "ZC", "AD", "ES-s-P")
)
```

Arguments

x data set

pnull cdf under the null hypothesis

rnull routine to generate data under the null hypothesis

w (Optional) function to calculate weights, returns -99 if no weights

phat =function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated

TS user supplied function to find test statistics, if any

TSextra =NA, list passed to TS, if desired

nbins	=c(50, 10) number of bins for chi-square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
Range	=c(-Inf, Inf) limits of possible observations, if any, for chi-square tests
B	=c(5000,1000) number of simulation runs for p values and for p value distribution
minexpcount	=5 minimal expected bin count required
ChiUsePhat	=TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.
doMethods	Methods to include in tests

Value

None

gof_test_disc	<i>This function performs a number of gof tests for discrete data.</i>
---------------	--

Description

This function performs a number of gof tests for discrete data.

Usage

```
gof_test_disc(
  x,
  pnull,
  rnull,
  vals,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  nbins = c(50, 10),
  rate = 0,
  B = 5000,
  minexpcount = 5,
  ChiUsePhat = TRUE,
  maxProcessors = 1,
  doMethods = "Default"
)
```

Arguments

x	data set (the counts)
pnull	cumulative distribution function under the null hypothesis
rnull	routine to generate data under the null hypothesis

vals	a vector of values of discrete random variables
phat	=function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated
TS	=NA, user supplied function to find test statistics
TSextra	=NA, list passed to TS, if desired
nbins	=c(50, 10) number of bins for chi-square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
B	=5000 number of simulation runs
minexpcount	=5 minimal expected bin count required
ChiUsePhat	= TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.
maxProcessors	=1, number of processors to use in parallel processing. If missing single processor is used.
doMethods	Methods to include in tests

Value

A numeric matrix of test statistics and p.values

gof_test_disc_adj	<i>This function performs a number of gof tests for discrete data and finds the adjusted p value</i>
-------------------	--

Description

This function performs a number of gof tests for discrete data and finds the adjusted p value

Usage

```
gof_test_disc_adj(
  x,
  pnull,
  rnull,
  vals,
  phat = function(x) -99,
  TS,
  TSextra = NA,
  nbins = c(50, 10),
  rate = 0,
  B = 5000,
  minexpcount = 5,
  ChiUsePhat = TRUE,
  doMethods = c("Wassp1", "W", "AD", "s-P")
)
```

Arguments

x	data set (the counts)
pnull	cumulative distribution function under the null hypothesis
rnull	routine to generate data under the null hypothesis
vals	a vector of values of discrete random variables
phat	=function(x) -99, function to estimate parameters from the data, or -99 if no parameters are estimated
TS	=NA, user supplied function to find test statistics
TSextra	=NA, list passed to TS, if desired
nbins	=c(50, 10) number of bins for chi-square tests
rate	=0 rate of Poisson if sample size is random, 0 if sample size is fixed
B	=c(5000, 1000) number of simulation runs for p values and for adjusted p value
minexpcount	=5 minimal expected bin count required
ChiUsePhat	= TRUE, if TRUE param is estimated parameter, otherwise minimum chi square method is used.
doMethods	Methods to include in tests

Value

A numeric matrix of test statistics and p.values

make_bins_cont	<i>This function creates several type of bins for continuous data</i>
----------------	---

Description

This function creates several type of bins for continuous data

Usage

```
make_bins_cont(
  x,
  pnull,
  qnull = NA,
  phat = function(x) -99,
  DataBased = FALSE,
  nbins = c(50, 10),
  minexpcount = 5,
  Range = c(-99999, 99999)
)
```

Arguments

x	data set
pnull	cdf under the null hypothesis
qnull	=NA quantile function, if available
phat	=function(x) -99 parameters for pnull
DataBased	=FALSE bins based on data, not expected counts
nbins	=c(50, 10) number of bins
minexpcount	=5 smallest expected count per bin
Range	=c(-99999, 99999) limits of possible observations, if any

Value

A list of bins and bin probabilities

make_bins_disc	<i>This function creates several types of bins for discrete data</i>
----------------	--

Description

This function creates several types of bins for discrete data

Usage

```
make_bins_disc(
  x,
  pnull,
  phat = function(x) -99,
  nbins = c(50, 10),
  minexpcount = 5
)
```

Arguments

x	counts
pnull	cumulative distribution function
phat	=function(x) -99, function to estimated parameters, or -99
nbins	=c(50, 10) number of bins
minexpcount	=5 smallest expected count per bin

Value

A list of indices

plot_power	<i>This function draws the power graph, with curves sorted by the mean power and smoothed for easier reading.</i>
------------	---

Description

This function draws the power graph, with curves sorted by the mean power and smoothed for easier reading.

Usage

```
plot_power(pwr, xname = " ", title, Smooth = TRUE, span = 0.25)
```

Arguments

pwr	a matrix of power values, usually from the twosample_power command
xname	Name of variable on x axis
title	(Optional) title of graph
Smooth	=TRUE lines are smoothed for easier reading
span	=0.25bandwidth of smoothing method

Value

plt, an object of class ggplot.

signif.digits	<i>This function does some rounding to nice numbers</i>
---------------	---

Description

This function does some rounding to nice numbers

Usage

```
## S3 method for class 'digits'
signif(x, d = 4)
```

Arguments

x	a list of two vectors
d	=4 number of digits to round to

Value

A list with rounded vectors

Index

check.functions, 2
chi_power_cont, 2
chi_power_disc, 4
chi_test_cont, 5
chi_test_disc, 6

gof_power, 7
gof_power_cont, 9
gof_power_disc, 10
gof_test, 11
gof_test_adjusted_pvalue, 13
gof_test_cont, 15
gof_test_cont_adj, 16
gof_test_disc, 17
gof_test_disc_adj, 18

make_bins_cont, 19
make_bins_disc, 20

plot_power, 21

signif.digits, 21