Package ‘Rgof’

October 30, 2023

**Title** 1d Goodness of Fit Tests

**Version** 1.2.2

**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** 2023-10-30 20:20:02 UTC

### R topics documented:

- check.functions .................................................................................. 2
- gof_power_cont .................................................................................... 2
- gof_power_disc ..................................................................................... 4
- gof_test_cont ....................................................................................... 5
- gof_test_disc ....................................................................................... 6
- plot_power ............................................................................................. 8

**Index** 9
check.functions

This function checks whether the inputs have the correct format

Usage

    check.functions(pnull, rnull, qnull, phat, vals, x)

Arguments

    pnull    cdf under the null hypothesis
    rnull    routine to generate data under the null hypothesis
    qnull    routine to calculate quantiles under null hypothesis
    phat     function to estimate parameters from the data
    vals     vector of discrete values
    x        data

gof_power_cont

Find the power of various gof tests for continuous data.

Usage

    gof_power_cont(
        pnull, rnull, qnull, ralt, param_alt, phat, TS,
        alpha = 0.05, Range = c(-Inf, Inf), B = c(1000, 1000),
        nbins = c(100, 10), rate = 0, maxProcessors, minexpcount = 2
    )
Arguments

- `pnull` function to find cdf under null hypothesis
- `rnull` function to generate data under null hypothesis
- `qnull` quantile function (inverse cdf). If missing Wasserstein test can not be done.
- `ralt` function to generate data under alternative hypothesis
- `param_alt` vector of parameter values for distribution under alternative hypothesis
- `phat` function to estimate parameters from the data
- `TS` user supplied function to find test statistics
- `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` = 2 minimal expected bin count required

Value

A numeric matrix of power values.

Examples

```r
# 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)
qnull = function(x) qnorm(x)
rnull = function() rnorm(50)
ralt = function(mu) rnorm(50, mu)
gof_power_cont(pnull, rnull, qnull, ralt, c(0.25, 0.5), 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))
qnull = function(x, p=c(0, 1)) qnorm(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))
gof_power_cont(pnull, rnull, qnull, ralt, c(0, 1), phat, B=c(200, 200), maxProcessor=2)
```
Find the power of various gof tests for discrete data.

Usage

```r
gof_power_disc(
  pnull,
  rnull,
  vals,
  ralt,
  param_alt,
  phat,
  TS,
  alpha = 0.05,
  B = c(1000, 1000),
  nbins = c(100, 10),
  rate = 0,
  maxProcessors,
  minexpcount = 2
)
```

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 to estimate parameters from the data
- `TS`: user supplied function to find test statistics
- `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 chisquare 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`: =2 minimal expected bin count required
Value

A numeric matrix of power values.

Examples

# Power of tests when null hypothesis specifies a binomial N=10, p=0.5 distribution but
# true data comes from a binomial distribution with success probability 0.55 or 0.6.
vals=0:10
pnull = function() pbinom(0:10, 10, 0.5)
rnull = function() table(c(0:10, rbinom(1000, 10, 0.5)))-1
ralt = function(p) table(c(0:10, rbinom(1000, 10, p)))-1
gof_power_disc(pnull, rnull, vals, ralt, c(0.515, 0.53), B=c(500, 500))

# Power of tests when null hypothesis specifies a binomial N=10 distribution and
# p is estimated from the data.
pnull=function(p=0.5) pbinom(0:10, 10, p)
rnull = function(p=0.5) table(c(0:10, rbinom(1000, 10, p)))-1
ralt = function(p=0.5) table(c(0:10, rbinom(1000, 10, p)))-1
phat = function(x) mean(rep(0:10, x))/10
gof_power_disc(pnull, rnull, vals, ralt, phat, param_alt=0.6, B=c(100, 100), maxProcessors = 2)

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,
  qnull,
  phat,
  TS,
  nbins = c(100, 10),
  rate = 0,
  Range = c(-Inf, Inf),
  B = 5000,
  minexpcount = 2,
  maxProcessors = 1,
  doMethod = "Default"
)
Arguments

- x: data set
- pnull: cdf under the null hypothesis
- rnull: routine to generate data under the null hypothesis
- qnull: routine to calculate quantiles under null hypothesis
- phat: function to estimate parameters from the data
- TS: user supplied function to find test statistics
- 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: =2 minimal expected bin count required
- maxProcessors: =1 number of processors to use in parallel processing. If missing single processor is used.
- doMethod: 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)
qnull = function(x) qnorm(x)
rnull = function() rnorm(100)
x = rnorm(100)
gof_test_cont(x, pnull, rnull, qnull, doMethod="all")
# Tests to see whether data comes from a normal distribution with
# mean and standard deviation estimated from the data.
pnull = function(x, p=c(0, 1)) pnorm(x, p[1], ifelse(p[2]>0.001, p[2], 0.001))
qnull = function(x, p=c(0, 1)) qnorm(x, p[1], ifelse(p[2]>0.001, p[2], 0.001))
rnull = function(p=c(0, 1)) rnorm(100, p[1], ifelse(p[2]>0.001, p[2], 0.001))
phat = function(x) c(mean(x), sd(x))
gof_test_cont(x, pnull, rnull, qnull, phat)

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

Description

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

Usage

gof_test_disc(
  x,
  pnull,
  rnull,
  vals,
  phat,
  TS,
  nbins = c(100, 10),
  rate = 0,
  B = 5000,
  minexpcount = 2,
  maxProcessors = 1,
  doMethod = "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  a function to estimate parameters from the data
TS    user supplied function to find test statistics
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
B     =5000 number of simulation runs
minexpcount =2 minimal expected bin count required
maxProcessors =1 number of processors to use in parallel processing. If missing single processor is used.
doMethod Methods to include in tests

Value

A numeric matrix of test statistics and p values

Examples

# 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_disc(x, pnull, rnull, vals, doMethod="all")
# Tests to see whether data comes from a binomial distribution with the success probability
# estimated from the data.
plot_power

$pnull = function(p=0.5) pbinom(0:10, 10, ifelse(p>0&&p<1,p,0.001))$
$rnull = function(p=0.5) table(c(0:10, rbinom(1000, 10, ifelse(p>0&&p<1,p,0.001))))[-1]
$phat = function(x) mean(0:10*x)/1000$
$gof_test_disc(x, pnull, rnull, vals, phat)$

plot_power

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

Description

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

Usage

plot_power(pwr, xname = "", 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
Smooth =TRUE lines are smoothed for easier reading
span =0.25 bandwidth of smoothing method

Value

plt, an object of class ggplot.
Index

check.functions, 2

gof_power_cont, 2
gof_power_disc, 4
gof_test_cont, 5
gof_test_disc, 6

plot_power, 8