# Package 'PoweR'

July 22, 2025

Version	<b>n</b> 1.1.4
Date 2	2025-07-14
Title (	Computation of Power and Level Tables for Hypothesis Tests
V A F	r Pierre Lafaye De Micheaux [aut, cre], Viet Anh Tran [aut], Alain Desgagne [aut], Frederic Ouimet [aut], Steven G. Johnson [aut]
Mainta	ainer Pierre Lafaye De Micheaux <lafaye@unsw.edu.au></lafaye@unsw.edu.au>
Depen	<b>ds</b> R (>= 4.4.0), parallel, Rcpp
f i	ption Computes power and level tables for goodness-of-fit tests for the normal, Laplace, and uniform distributions. Generates output in 'LaTeX' format to facilitate reporting and reproducibility. Explanatory graphs help visualize the statistical power of test statistics under various alternatives. For more details, see Lafaye De Micheaux and Tran (2016) <doi:10.18637 jss.v069.i03="">.</doi:10.18637>
License	<b>e</b> GPL (>= 2)
Linkin	gTo Rcpp, RcppArmadillo
	Compilation yes
	itory CRAN
•	Publication 2025-07-22 10:20:13 UTC
Dawi	ubileation 2023-07-22 10.20.13 0 1C
Cont	tents
	calcFx checklaw compquant create.alter densities Distributions gensample getindex getnbparlaws 1

2 Contents

getnbparstats					
help.law					•
help.stat					
Laplace.tests					
law.cstr					
law0001.Laplace	 	 			
law0002.Normal	 	 			
law0003.Cauchy	 	 			
law0004.Logistic	 	 			
law0005.Gamma	 	 			
law0006.Beta	 	 			
law0007.Uniform					
law0008.Student					
law0009.Chisquared					
1					
law0010.LogNormal					
law0011.Weibull					
law0012.ShiftedExp					
law0013.PowerUnif					
law0014.AverageUnif					
law0015.UUnif	 	 			
law0016.VUnif	 	 			
law0017.JohnsonSU	 	 			
law0018.Tukey	 	 			
law0019.LocationCont	 	 			
law0020.JohnsonSB					
law0021.SkewNormal					
law0022.ScaleCont					
law0023.GeneralizedPareto					
law0024.GeneralizedError					
law0025.Stable					
law0026.Gumbel					
law0027.Frechet					
law0028.GeneralizedExtValue					
law0029.GeneralizedArcsine					
$law 0030. Folded Normal \ \dots \ \dots \ \dots \ \dots$					
law0031.MixtureNormal	 	 			
law0032.TruncatedNormal	 	 			
law0033.Nout	 	 			
law0034.GeneralizedExpPower	 	 			
law0035.Exponential					
law0036.AsymmetricLaplace					
law0037.NormalInvGaussian					
law0038.AsymmetricPowerDistribution					
law0039.modifiedAsymmetricPowerDistribution					
law0040.Log-Pareto-tail-normal					
many.crit	 	 	 •	•	

Contents 3

moments	
Normality.tests	
plot.discrepancy	. 67
plot.pvalue	. 68
plot.sizepower	. 69
powcomp.easy	. 70
powcomp.fast	. 72
power.gui	. 75
print.critvalues	. 76
print.power	. 78
pvalueMC	. 79
stat.cstr	. 80
stat0001.Lilliefors	. 81
stat0002.AndersonDarling	. 82
stat0003.ZhangWu1	. 82
stat0004.ZhangWu2	. 83
stat0005.GlenLeemisBarr	. 84
stat0006.DAgostinoPearson	
stat0007.JarqueBera	. 85
stat0008.DoornikHansen	. 86
stat0009.GelGastwirth	
stat0010.Hosking1	
stat0011.Hosking2	
stat0012.Hosking3	
stat0013.Hosking4	
stat0014.BontempsMeddahi1	
stat0015.BontempsMeddahi2	. 90
stat0016.BrysHubertStruyf	. 91
stat0017.BonettSeier	. 91
stat0018.BrysHubertStruyf-BonettSeier	. 92
stat0019.CabanaCabana1	
stat0020.CabanaCabana2	
stat0021.ShapiroWilk	
stat0022.ShapiroFrancia	
stat0023.ShapiroWilk-RG	
stat0024.DAgostino	. 96
stat0025.Filliben	. 97
stat0026.ChenShapiro	
stat0027.ZhangQ	
stat0028.ZhangQQstar	
stat0029.BarrioCuestaMatranRodriguez	
stat0030.Coin	
stat0031.EppsPulley	
stat0032.MartinezIglewicz	
stat0033.GelMiaoGastwirth	
stat0034.ZhangQstar	
stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn	
stat0036.DesgagneLafayeDeMicheaux-XAPD	. 104

4 Contents

stat0037.DesgagneLafayeDeMicheaux-ZEPD	105
stat0038.Glen	105
stat0039.Rayner1	106
stat0040.Rayner2	107
stat0041.Spiegelhalter	107
stat0042.AndersonDarling	108
stat0043.CramervonMises	109
stat0044.Watson	109
stat0045.KolmogorovSmirnov	110
stat0046.Kuiper	111
stat0047.Meintanis1MO	111
stat0048.Meintanis1ML	112
stat0049.Meintanis2MO	113
stat0050.Meintanis2ML	113
stat0051.ChoiKim1	114
stat0052.ChoiKim2	115
stat0053.ChoiKim3	116
stat0054.DesgagneMicheauxLeblanc-Gn	116
stat0055.RaynerBest1	
stat0056.RaynerBest2	
stat0057.LangholzKronmal	
stat0058.Kundu	
stat0059.Gulati	
stat0060.Gel	
stat0061.DesgagneMicheauxLeblanc-Lap1	
stat0062.DesgagneMicheauxLeblanc-Lap2	
stat0063.Kolmogorov	
stat0064.CramervonMises	
stat0065.AndersonDarling	
stat0066.Durbin	
stat0067.Kuiper	
stat0068.HegazyGreen1	
stat0069.HegazyGreen2	
stat0070.Greenwood	
stat0071.QuesenberryMiller	
stat0072.ReadCressie	
stat0073.Moran	
stat0074.Cressie1	130
stat0075.Cressie2	
stat0076.Vasicek	
stat0077.Swartz	
stat0078.Morales	
stat0079.Pardo	
stat0080.Marhuenda	
stat0081.Zhang1	
stat0082.Zhang2	
stat0083.ttests	
stat0085.DesgagneMicheauxLeblanc-Lap3	

calcFx 5

stat0086.Lafaye
stat0087.Lafaye2
stat0088.Lafaye3
stat0089.Lafaye4
stat0090.Lafaye5
stat0091.Gonzales1
stat0092.Gonzales2
stat0093.Hogg1
stat0094.Hogg2
stat0095.Hogg3
stat0096.Hogg4
stat0097.Rizzo
statcompute
testsPureR
Uniformity.tests

Index 149

calcFx

Empirical distribution function of p-values.

# **Description**

This function computes, at given points, the value of the empirical distribution function of a sample of p-values.

# Usage

```
calcFx(pval.mat, x = c(seq(0.001, 0.009, by = 0.001), seq(0.01, 0.985, by = 0.005),
       seq(0.99, 0.999, by = 0.001)))
```

# **Arguments**

pval.mat

matrix whose each column contains a vector of p-values for a given test statistic. The column names of this matrix should be set to the names of the various test statistics considered, whereas the rownames should all be set to the name of the distribution under which the p-values have been computed. This matrix can be obtained using function many.pval.

Х

vector of points at which to evaluate the empirical distribution function.

## **Details**

See equation (2) in Lafaye de Micheaux and Tran (2014).

6 checklaw

#### Value

An object of class Fx is returned, which contains a list whose components are:

Fx.mat matrix whose ith column contains the values of the empirical distribution function (evaluated at the points in vector x) of the *p*-values of the ith test statistic.

x same vector x as input.

law name of the distribution under which the p-values have been computed. Should correspond to the row names of pval.mat.

statnames names of the test statistics. Should correspond to the column names of pval.mat.

N number of p-values used.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

```
many.pval, plot.pvalue, plot.discrepancy, plot.sizepower
```

## **Examples**

checklaw

Check proper behaviour of a random generator

# **Description**

It is desirable to check if a newly added random generator coded in C behaves correctly. To perform this operation, one can superimpose the theoretical density on a histogram of the generated values.

#### Usage

```
checklaw(law.index, sample.size = 50000, law.pars = NULL, density =
NULL, trunc = c(-Inf, Inf), center = FALSE, scale = FALSE)
```

checklaw 7

# Arguments

law.index	index of the desired law, as given by getindex.
sample.size	number of observations to generate.
law.pars	vector of parameters for the law. The length of this parameter should not exceed 4. If not provided, the default values are used by means of getindex function.
density	a function of two arguments x and pars. Can also be a function of the arguments x and pars[1],, pars[k]. See the two examples below.
trunc	vector of left and right truncation thresholds for the generated sample values. Only those values in between will be kept to build the histogram. This can be useful for a distribution with extreme values.
center	Logical. Should we center the data.
scale	Should we scale the data.

## Value

Returns invisibly the data generated and make a plot showing histogram and density superimposed.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

gensample

```
dlaplace1 <- function(x, mu, b) {dexp(abs(x - mu), 1 / b) / 2}
checklaw(1, density = dlaplace1)
dlaplace2 <- function(x, pars) {dexp(abs(x - pars[1]), 1 / pars[2]) / 2}
checklaw(1, density = dlaplace2)

checklaw(law.index = 2, sample.size = 50000, law.pars = c(2, 3), density = dnorm)

## We use the 'trunc' argument to display the density in a region where
## no extreme values are present.
checklaw(27, density = dlaw27, trunc = c(-Inf,10))

# This one (Tukey) does not have a closed form expression for
# the density. But we can use the stats::density() function as</pre>
```

8 compquant

```
# follows.
res <- checklaw(18)
lines(density(res$sample), col = "blue")</pre>
```

compquant

Computation of the quantile values only for one test statistic.

# Description

Functions for the computation of the quantile values only for one test statistic at a time and also one n value.

# Usage

# Arguments

n	number of observations for each sample to be generated; length(n)=1. This can also be set to 0 if you want to use your own function using the 'Rstat' argument (see below).
law.index	law index as given by getindex; length(law.index)=1.
stat.index	stat index as given by getindex; length(stat.index)=1.
probs	If not NULL, should be a vector of levels from which to compute the quantile values. If NULL, the levels 0.025,0.05,0.1,0.9,0.95,0.975 will be used.
М	Number of Monte Carlo repetitions to use.
law.pars	NULL or a vector of length at most 4 containing 4 possible parameters to generate random values from distribution law(law.pars[ $j$ ], $j <= 4$ ). If NULL, the default parameter values for the law specified by law.index will be used.
stat.pars	A vector of parameters. If NULL, the default parameter values for the statistic specified by this stat.index will be used.
model	NOT YET IMPLEMENTED. If NULL, no model is used. If an integer $i>0$ , the model coded in the C function modele $i$ is used. Else this should be an R function that takes three arguments: eps (vector of $\epsilon$ values), thetavec (vector of $\theta$ values) and xvec (vector or matrix of $x$ values). This function should take a vector of errors, generate observations from a model (with parameters thetavec and values xvec) based on these errors, then compute and return the residuals from the model. See file modele1.R in directory inst/doc/ for an example in multiple linear regression.
Rlaw	The user can provide its own (random generating) R function using this param-

eter. In this case, 'law.index' should be set to 0.

create.alter 9

Rstat	If 'stat.index' is set to 0, an R function that outputs a list with components 'statis-
	tic' (value of the test statistic), 'pvalue' (pvalue of the test; if not computable
	should be set to 0), 'decision' (1 if we reject the null, 0 otherwise), 'alter' (see
	above), 'stat.pars' (see above), 'pvalcomp' (1L if the pvalue can be computed,

OL otherwise), 'nbparstat' (length of stat.pars).

center Logical. Should we center the data generated scale Logical. Should we center the data generated

#### Value

A list with M statistic values and also some quantiles (with levels 0.025,0.05,0.1,0.9,0.95,0.975), as well as the name of the law and the name of the test statistic used (just to be sure!).

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## **Examples**

```
compquant(n=50,law.index=2,stat.index=10,M=10^3)$quant
compquant(n=50,law.index=0,stat.index=10,M=10^3,Rlaw=rnorm)$quant
```

create.alter

Create a list giving the type of test statistics.

# Description

Create a list giving the type of each test statistic for a given vector of indices of these test statistics.

#### Usage

```
create.alter(stat.indices = c(42, 51, 61), values.alter = NULL)
```

## **Arguments**

stat.indices vector of indices of test statistics, as given by function getindex.

values.alter vector of the type of each test statistic in stat.indices. If NULL, the default

value will be returned.

## Details

See Section 3.3 in Lafaye de Micheaux, P. and Tran, V. A. (2014).

10 densities

## Value

A named list. Each component of the list has the name of the corresponding index in stat.indices (e.g. statxxx) and has the value (in  $\{0,1,2,3,4\}$ ) of the type of test (see Details above).

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See getindex.

## **Examples**

create.alter()

densities

Density function

# Description

Evaluate the density function at a vector points.

#### **Details**

Use the function by typing:

dlawj(x,par1,par2,etc.)

where j is the index of the law and par1, par2, etc. are the parameters of law j.

The indicator function takes a vector x of length n as first argument and two real values a < b. It returns a vector of length n which contains only 0s and 1s (1 if the corresponding value in x is strictly between a and b).

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Distributions 11

Distributions

Distributions in the PoweR package

# **Description**

Random variate generation for many standard probability distributions are available in the PoweR package.

#### **Details**

The functions for the random variate generation are named in the form lawxxxx.

For the Laplace distribution see law0001.Laplace.

For the Normal distribution see law0002.Normal.

For the Cauchy distribution see law0003. Cauchy.

For the Logistic distribution see law0004.Logistic.

For the Gamma distribution see law0005. Gamma.

For the Beta distribution see law0006.Beta.

For the Uniform distribution see law0007. Uniform.

For the Student distribution see law0008. Student.

For the Chi-Squared distribution see law0009. Chisquared.

For the Log Normal distribution see law0010.LogNormal.

For the Weibull distribution see law0011. Weibull.

For the Shifted Exponential distribution see law0012. ShiftedExp.

For the Power Uniform distribution see law0013. PowerUnif.

For the Average Uniform distribution see law0014. AverageUnif.

For the UUniform distribution see law0015.UUnif.

For the VUniform distribution see law0016. VUnif.

For the Johnson SU distribution see law0017. JohnsonSU.

For the Tukey distribution see law0018. Tukey.

For the Location Contaminated distribution see law0019.LocationCont.

For the Johnson SB distribution see law0020. JohnsonSB.

For the Skew Normal distribution see law0021. SkewNormal.

For the Scale Contaminated distribution see law0022. ScaleCont.

For the Generalized Pareto distribution see law0023. GeneralizedPareto.

For the Generalized Error distribution see law0024. GeneralizedError.

For the Stable distribution see law0025. Stable.

For the Gumbel distribution see law0026. Gumbel.

For the Frechet distribution see law0027.Frechet.

12 gensample

For the Generalized Extreme Value distribution see law0028. GeneralizedExtValue.

For the Generalized Arcsine distribution see law0029. GeneralizedArcsine.

For the Folded Normal distribution see law0030.FoldedNormal.

For the Mixture Normal distribution see law0031.MixtureNormal.

For the Truncated Normal distribution see law0032. TruncatedNormal.

For the Normal with outliers distribution see law0033. Nout.

For the Generalized Exponential Power distribution see law0034.GeneralizedExpPower.

For the Exponential distribution see law0035. Exponential.

For the Asymmetric Laplace distribution see law0036. AsymmetricLaplace.

For the Normal-inverse Gaussian distribution see law0037.NormalInvGaussian.

For the Asymmetric Power Distribution see law0038. AsymmetricPowerDistribution.

For the modified Asymmetric Power Distribution see law0039.modifiedAsymmetricPowerDistribution.

For the Log-Pareto-tail-normal distribution see law0040.Log-Pareto-tail-normal.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

The CRAN task view on distributions, https://CRAN.R-project.org/view=Distributions, mentioning several CRAN packages for additional distributions.

gensample

Generate random samples from a law added in the package.

#### **Description**

Generate random samples from a law added in the package as a C function.

## Usage

```
gensample(law.index,n,law.pars = NULL,check = TRUE, center=FALSE, scale=FALSE)
```

gensample 13

# **Arguments**

law.index	law index as given by function getindex.
n	number of observations to generate.
law.pars	vector of parameters for the law. The length of this parameter should not exceed 4.
check	logical. If TRUE, we check if law.index belongs to the list of laws. If FALSE, we pass on this verification, this will reduce the simulation time.
center	Logical. Should we center the data generated
scale	Logical. Should we center the data generated

## Value

A list containing the random sample and the vector of parameters used for the chosen law.

## Author(s)

```
P. Lafaye de Micheaux, V. A. Tran
```

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See checklaw

```
# This is good to check if the generator of the given law has been well coded.

res <- gensample(2,10000,law.pars=c(-5,2),check=TRUE)
res$law
res$law.pars
mean(res$sample)

# See function checklaw() in this package.
hist(gensample(2,10000,law.pars=c(0,1),check=TRUE)$sample,prob=TRUE,breaks=100,main="Density histogram of the N(0,1) distribution")
curve(dnorm(x),add=TRUE,col="blue")</pre>
```

14 getindex

getindex

Get indices of laws and statistics functions.

# **Description**

Print two correspondence tables between indices and random generators functions or test statistics functions programmed in C in this package. The first table gives indices/laws and the second one gives indices/statistics. These indices can be used in the functions powcomp.easy, powcomp.fast, compquant, gensample, statcompute, checklaw.

# Usage

```
getindex(law.indices = NULL, stat.indices = NULL)
```

# **Arguments**

law.indices if not NULL, select only the laws corresponding to this vector of indices. stat.indices if not NULL, select only the stats corresponding to this vector of indices.

#### Value

A list with two matrices. The first one gives the correspondence between the indices and the laws (with also the number of parameters for each law as well as the default values). The second one gives the correspondence between the indices and the test statistics. Note that you can use the law.indices or stat.indices parameters of this function to obtain only some part of these tables of correspondence.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

```
See getnbparlaws, getnbparstats, stat.cstr,law.cstr.
```

```
getindex(1,c(4,3))
```

getnbparlaws 15

getnbparlaws

Retrieve the default number of parameters of some laws.

## **Description**

Retrieve the default number of parameters of the distributions in the package.

## Usage

```
getnbparlaws(law.indices = NULL)
```

# Arguments

law.indices

vector of the indices of the distributions from which to retrieve the default number of parameters. If NULL, all the distributions will be considered.

#### Value

The default number of parameters for the laws specified in law.indices.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See getnbparstats, getindex, law.cstr, stat.cstr.

```
## Default numbers of parameters for all the distributions in the package:
getnbparlaws()
## The Gaussian distribution has two parameters:
getnbparlaws(2)
```

16 getnbparstats

getnbparstats

Get numbers of parameters of test statistics.

# **Description**

Return the default numbers of parameters of the test statistics in the package.

# Usage

```
getnbparstats(stat.indices = NULL)
```

## **Arguments**

stat.indices if not NULL, select only the statistics corresponding to this vector of indices.

## Value

A vector giving the numbers of parameters of test statistics corresponding to the vector of indices.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

See getnbparlaws, getindex, law.cstr, stat.cstr.

```
getnbparstats(c(42:53))
```

graph 17

graph

p-value plot, p-value discrepancy plot and size-power curves.

## **Description**

This function draws a p-value plot, a p-value discrepancy plot or a size-power curves plot.

#### **Usage**

```
graph(matrix.pval, xi = c(seq(0.001, 0.009, by = 0.001),
    seq(0.01, 0.985, by = 0.005), seq(0.99, 0.999, by = 0.001)),
    type = c("pvalue.plot", "pvalue.discrepancy", "size.power"),
    center = FALSE, scale = FALSE)
```

# Arguments

matrix.pval a matrix of p-values as returned by function many.pval.

xi a vector of values at which to compute the empirical distribution of the p-values.

type character. Indicate the type of plot desired.

center Logical. Should we center the data generated

scale Logical. Should we center the data generated

#### **Details**

See Section 2.3 in Lafaye de Micheaux, P. and Tran, V. A. (2014).

#### Value

No return value. Displays a graph.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See plot.pvalue, plot.discrepancy, plot.sizepower.

18 help.law

## **Examples**

help.law

Help Law

# Description

Open directly the documentation for a specified law using its index.

# Usage

```
help.law(law.index)
```

## **Arguments**

law.index

law index as given by function getindex.

## Value

No return value. The function opens the help page for the law corresponding to the given index.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

Distributions for other standard distributions.

help.stat 19

help.stat

Help Stat

# Description

Open directly the documentation for a specified goodness-of-fit using its index.

## Usage

```
help.stat(stat.index)
```

#### **Arguments**

stat.index

statistic index as given by function getindex.

## Value

No return value. The function opens the help page for the test corresponding to the given index.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See Normality. tests for goodness-of-fit tests for normality. See Laplace. tests for goodness-of-fit tests for the Laplace distribution. See Uniformity. tests for goodness-of-fit tests for uniformity.

Laplace.tests

Goodness-of-fit tests for the Laplace distribution

## **Description**

List of goodness-of-fit tests for the Laplace distribution.

20 Laplace.tests

#### **Details**

```
The statistic tests for the Laplace distribution are named in the form statxxxx.
For the Glen-Leemis-Barr test see stat0038. Glen.
For the 1st Rayner-Best statistic test see stat0039.Rayner1.
For the 2nd Rayner-Best statistic test see stat0040.Rayner2.
For the Anderson-Darling statistic see stat0042. AndersonDarling.
For the Cramer-von Mises statistic see stat0043. CramervonMises.
For the Watson statistic see stat0044. Watson.
For the Kolmogorov-Smirnov statistic see stat0045.KolmogorovSmirnov.
For the Kuiper statistic see stat0046. Kuiper.
For the 1st Meintanis statistic with moment estimators see stat0047. Meintanis1MO.
For the 1st Meintanis statistic with maximum likelihood estimators see stat0048. Meintanis1ML.
For the 2nd Meintanis statistic with moment estimators see stat0049. Meintanis 2Mo.
For the 2nd Meintanis statistic with maximum likelihood estimators see stat0050. Meintanis 2ML.
For the 1st Choi-Kim statistic see stat0051. ChoiKim1.
For the 2nd Choi-Kim statistic see stat0052. ChoiKim2.
For the 3rd Choi-Kim statistic see stat0053. ChoiKim3.
For the Desgagne-Micheaux-Leblanc statistic see stat0054.DesgagneMicheauxLeblanc-Gn.
For the 1st Rayner-Best statistic see stat0055. RaynerBest1.
For the 2nd Rayner-Best statistic see stat0056. RaynerBest2.
For the Langholz-Kronmal statistic see stat0057. LangholzKronmal.
For the Kundu statistic see stat0058.Kundu.
For the Gulati statistic see stat0059.Gulati.
For the Gel statistic see stat0060.Gel.
For the 1st Gonzalez-Estrada and Villasenor test see stat0091. Gonzales1.
For the 2nd Gonzalez-Estrada and Villasenor test see stat0092.Gonzales2.
For the 1st Hogg test see stat0093. Hogg1.
For the 2nd Hogg test see stat0094. Hogg2.
For the 3rd Hogg test see stat0095. Hogg3.
For the 4th Hogg test see stat0096. Hogg4.
For the Rizzo and Haman test see stat0097.Rizzo.
```

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

law.cstr 21

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See Normality. tests for goodness-of-fit tests for normality. See Uniformity. tests for goodness-of-fit tests for uniformity.

law.cstr

Gives information about a given law.

## **Description**

To obtain the name of a law as well as its default number of parameters and default parameter values.

# Usage

```
law.cstr(law.index, law.pars = NULL)
```

## **Arguments**

law.index a single integer value corresponding to the index of a distribution as given by

function getindex.

law.pars vector of the values of the parameters of the law specified in law.index. If

NULL, the default values are used.

## Details

This function can be useful to construct a title for a graph for example.

#### Value

name name of the distribution with its parameters and the values they take.

nbparams default number of parameters of the law.

law.pars values of the parameters.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

22 law0001.Laplace

## See Also

See stat.cstr, getindex, getnbparlaws, getnbparstats.

## **Examples**

```
law.cstr(2)
```

law0001.Laplace

The Laplace Distribution

#### **Description**

Random generation for the Laplace distribution with parameters mu and b.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu or b are not specified they assume the default values of 0 and 1, respectively.

The Laplace distribution has density:

$$\frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$$

where  $\mu$  is a location parameter and b > 0, which is sometimes referred to as the diversity, is a scale parameter.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See function urlaplace() from Runuran package. See Distributions for other standard distributions.

```
res <- gensample(1,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0002.Normal

law0002.Normal

The Normal Distribution

# **Description**

Random generation for the Normal distribution with parameters mu and sigma.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If mu or sigma are not specified they assume the default values of 0 and 1, respectively.

The Normal distribution has density:

$$(\sqrt{2\pi}\sigma)^{-1}\exp^{-\frac{x^2}{2\sigma^2}}$$

where  $\mu$  is the mean of the distribution and  $\sigma$  is the standard deviation.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See function link{rnorm} from stats package. See Distributions for other standard distributions.

```
res <- gensample(2,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

24 law0003.Cauchy

law0003.Cauchy

The Cauchy Distribution

# Description

Random generation for the Cauchy distribution with parameters location and scale.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If location or scale are not specified, they assume the default values of 0 and 1 respectively.

The Cauchy distribution has density:

$$\frac{1}{\pi s(1+(\frac{x-l}{s})^2)}$$

where l is the location parameter and s is the scale parameter, for all x.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See function reauchy from package stats. See Distributions for other standard distributions.

```
res <- gensample(3,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0004.Logistic 25

law0004.Logistic

The Logistic Distribution

# **Description**

Random generation for the Logistic distribution with parameters location and scale.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If location or scale are omitted, they assume the default values of 0 and 1 respectively.

The Logistic distribution with location =  $\mu$  and scale = s has distribution function

$$\frac{1}{1 + exp^{-\frac{(x-\mu)}{s}}}$$

and density

$$\frac{exp^{-\frac{(x-\mu)}{s}}}{s(1+exp^{-\frac{(x-\mu)}{s}})^2}$$

It is a long-tailed distribution with mean  $\mu$  and variance  $(\pi^2)/3s^2$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See function rlogis from package stats. See Distributions for other standard distributions.

```
res <- gensample(4,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

26 law0005.Gamma

law0005.Gamma

The Gamma Distribution

# **Description**

Random generation for the Gamma distribution with parameters shape and rate.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If shape or rate are not specified they assume the default values of 2 and 1, respectively.

The Gamma distribution has density:

$$\frac{1}{b^a\Gamma(a)}x^{a-1}exp^{-x/b}$$

for  $x \ge 0$ , a > 0 and b > 0; where a is the shape parameter and b is the rate parameter.

Here  $\Gamma(a)$  is the gamma function implemented by R and defined in its help.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

See function rgamma from package stats. See. Distributions for other standard distributions. Type help(gamma) for additional information about the gamma function.

```
res <- gensample(5,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0006.Beta 27

law0006.Beta

The Beta Distribution

# Description

Random generation for the Beta distribution with parameters shape1 and shape2.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If shape1 or shape2 are not specified they assume the default values of 1 and 1, respectively.

The Beta distribution with parameters shape1 = a and shape2 = b has density:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}x^{\alpha-1}(1-x)^{\beta-1}$$

for a>0, b>0 and  $0\leq x\leq 1$  where the boundary values at x=0 or x=1 are defined as by continuity (as limits).

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

See function rbeta from package stats. See Distributions for other standard distributions.

```
res <- gensample(6,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

28 law0007.Uniform

law0007.Uniform

The Uniform Distribution

# **Description**

Random generation for the Uniform distribution with parameters min and max.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If min or max are not specified they assume the default values of 0 and 1, respectively.

The Uniform distribution has density:

$$\frac{1}{max-min}$$

for  $min \le x \le max$ .

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See function runif from package stats. Distributions for other standard distributions.

```
res <- gensample(7,10000,law.pars=c(2,9))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0008.Student 29

law0008.Student

The Student t Distribution

# Description

Random generation for the Student t distribution with df degrees of freedom.

This generator is called by function gensample to create random variables based on its parameter.

#### **Details**

If df is not specified it assumes the default value of 1.

The t distribution with df = k degrees of freedom has density:

$$(\sqrt{k\pi})^{-1} \frac{\Gamma\left(\frac{k+1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \left(1 + \frac{t^2}{k}\right)^{-\frac{k+1}{2}}$$

for all real x. It has mean 0 (for k > 1) and variance k/(k-2) (for k > 2).

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

Distributions for other standard distributions.

```
res <- gensample(8,10000,law.pars=8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

30 law0009.Chisquared

law0009.Chisquared

The Chi-Squared Distribution

# Description

Random generation for the Chi-squared distribution with df degrees of freedom.

This generator is called by function gensample to create random variables based on its parameter.

## **Details**

If df is not specified it assumes the default value of 1.

The Chi-squared distribution with df = k degrees of freedom has density:

$$2^{-k/2}\Gamma(k/2)^{-1}x^{k/2-1}e^{-x/2}$$

for x > 0 and  $k \ge 1$ . The mean and variance are n and 2n.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

Distributions for other standard distributions.

```
res <- gensample(9,10000,law.pars=8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0010.LogNormal 31

law0010.LogNormal

The Log Normal Distribution

## **Description**

Random generation for the Log Normal distribution whose logarithm has mean equal to meanlog and standard deviation equal to sdlog.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If meanlog or sdlog are not specified they assume the default values of 0 and 1, respectively.

The Log Normal distribution has density:

$$\frac{1}{x\sigma\sqrt{2\pi}}e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$$

where  $\mu$  and  $\sigma$  are the mean and standard deviation of the logarithm. The mean is  $E(X) = exp(\mu + 1/2\sigma^2)$  and the variance is  $Var(X) = exp(2*\mu + \sigma^2)*(exp(\sigma^2) - 1)$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

Distributions for other standard distributions.

```
res <- gensample(10,10000,law.pars=c(8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

32 law0011.Weibull

law0011.Weibull

The Weibull Distribution

# Description

Random generation for the Weibull distribution with parameters shape and scale.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If shape or scale are not specified they assume the default values of 1 and 1, respectively.

The Weibull distribution with shape parameter k and scale parameter  $\lambda$  has density given by

$$\frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$$

for x > 0. The cumulative distribution function is  $F(x) = 1 - e^{(-(x/\lambda)^k)}$  on x > 0, the mean is  $E(X) = \lambda \Gamma(1 + 1/k)$ , and the  $Var(X) = \lambda^2 * (\Gamma(1 + 2/k) - (\Gamma(1 + 1/k))^2)$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

Distributions for other standard distributions.

```
res <- gensample(11,10000,law.pars=c(8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0012.ShiftedExp 33

law0012.ShiftedExp

The Shifted Exponential Distribution

# Description

Random generation for the Shifted Exponential distribution with parameters 1 and rate.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If 1 or rate are not specified they assume the default values of 0 and 1, respectively.

The Shifted Exponential distribution has density

$$b\exp\{-(x-l)b\}$$

for  $x \le 1$ , where rate = b. The mean is E(X) = l + 1/b, and the  $Var(X) = 1/(b^2)$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

Distributions for other standard distributions.

```
res <- gensample(12,10000,law.pars=c(8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

34 law0013.PowerUnif

law0013.PowerUnif

The Power Uniform Distribution

# Description

Random generation for the Power Uniform distribution with parameter power.

This generator is called by function gensample to create random variables based on its parameter.

## **Details**

If power is not specified it assumes the default value of 1.

The Power Uniform distribution has density:

$$\frac{1}{1+j}x^{-\frac{j}{j+1}}$$

where power = j.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Quesenberry and Miller (1977), Power studies of some tests for uniformity, *Journal of Statistical Computation and Simulation*, **5**(3), 169–191 (see p. 178)

## See Also

See law0007.Uniform for the Uniform distribution. See Distributions for other standard distributions.

```
res <- gensample(13,10000,law.pars=8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0014.AverageUnif The Average Uniform Distribution

## Description

Random generation for the Average Uniform distribution with parameters size, a and b.

This generator is called by function gensample to create random variables based on its parameter.

## **Details**

If size, a and b are not specified they assume the default values of 2, 0 and 1.

The Average Uniform distribution has density:

$$\frac{k^k}{(k-1)!} \sum_{j=0}^{\lfloor k \frac{x-a}{b-a} \rfloor} (-1)^j \binom{k}{j} (\frac{x-a}{b-a} - \frac{j}{k})^{k-1}$$

where size = k and for  $a \le x \le b$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Quesenberry and Miller (1977), Power studies of some tests for uniformity, *Journal of Statistical Computation and Simulation*, **5**(3), 169–191 (see p. 179)

## See Also

law0007. Uniform for the Uniform distribution.

Distributions for other standard distributions.

```
res <- gensample(14,10000,law.pars=c(9,2,3))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

36 law0015.UUnif

law0015.UUnif

The UUniform Distribution

# Description

Random generation for the UUniform distribution with parameter power.

This generator is called by function gensample to create random variables based on its parameter.

## **Details**

If power is not specified it assumes the default value of 1.

The UUniform distribution has density:

$$(2(1+j))^{-1}(x^{-j/(1+j)} + (1-x)^{-j/(1+j)})$$

where power = j.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Quesenberry and Miller (1977), Power studies of some tests for uniformity, *Journal of Statistical Computation and Simulation*, **5**(3), 169–191 (see p. 179)

#### See Also

law0007. Uniform for the Uniform distribution.

Distributions for other standard distributions.

```
res <- gensample(15,10000,law.pars=9)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0016.VUnif 37

law0016.VUnif

The VUniform Distribution

## **Description**

Random generation for the VUniform distribution with parameter size.

This generator is called by function gensample to create random variables based on its parameter.

### **Details**

If size is not specified it assumes the default value of 1.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Quesenberry and Miller (1977), Power studies of some tests for uniformity, *Journal of Statistical Computation and Simulation*, **5**(3), 169–191 (see p. 179)

#### See Also

See law0007.Uniform for the Uniform distribution. See Distributions for other standard distributions.

```
res <- gensample(16,10000,law.pars=8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

38 law0017.JohnsonSU

law0017.JohnsonSU

The Johnson SU Distribution

# Description

Random generation for the Johnson SU distribution with parameters mu, sigma, nu and tau.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu, sigma, nu and tau are not specified they assume the default values of 0, 1, 0 and 0.5, respectively.

The Johnson SU distribution with parameters mu =  $\mu$ , sigma =  $\sigma$ , nu =  $\nu$  and tau =  $\tau$  has density:

$$\frac{1}{c\sigma\tau}\frac{1}{\sqrt{z^2+1}}\frac{1}{\sqrt{2\pi}}e^{-r^2/2}$$

where  $r = -\nu + (1/\tau)sinh^-1(z), z = (x - (\mu + c * \sigma(\sqrt(\omega))sinh(w)))/(c * \sigma), c = ((w - 1)(wcosh(2\omega) + 1)/2)^-1/2, w = e^(\tau^2)$  and  $\omega = -\nu\tau$ .

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Distributions for other standard distributions.

```
res <- gensample(17,10000,law.pars=c(9,8,6,0.5))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

*law0018.Tukey* 39

law0018.Tukey

The Tukey Distribution

# Description

Random generation for the Tukey distribution with parameter lambda.

This generator is called by function gensample to create random variables based on its parameter.

#### **Details**

If lambda is not specified it assumes the default value of 1.

```
The Tukey distribution with lambda = \lambda has E[X]=0 and Var[X]=2/(\lambda^2)(1/(2\lambda+1)-\Gamma^2(\lambda+1)/\Gamma(2\lambda+2)).
```

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

Distributions for other standard distributions.

## **Examples**

```
res <- gensample(18,10000,law.pars=8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0019.LocationCont

The Location Contaminated Distribution

### **Description**

Random generation for the Location Contaminated distribution with parameters p and m.

This generator is called by function gensample to create random variables based on its parameters.

40 law0020.JohnsonSB

### **Details**

If p or m are not specified they assume the default values of 0.5 and 0, respectively.

The Location Contaminated distribution has density:

$$\frac{1}{\sqrt{2\pi}} \left[ pe^{-\frac{(x-m)^2}{2}} + (1-p)e^{-\frac{x^2}{2}} \right]$$

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

Distributions for other standard distributions.

### **Examples**

```
res <- gensample(19,10000,law.pars=c(0.8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0020.JohnsonSB

The Johnson SB Distribution

## **Description**

Random generation for the Johnson SB distribution with parameters g and d.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If g and d are not specified they assume the default values of 0 and 1, respectively.

The Johnson SB distribution has density:

$$\frac{d}{\sqrt{2\pi}} \frac{1}{x(1-x)} e^{-\frac{1}{2} \left(g + d \ln \frac{x}{1-x}\right)^2}$$

where d > 0.

law0021.SkewNormal 41

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See Distributions for other standard distributions.

### **Examples**

```
res <- gensample(20,10000,law.pars=c(8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0021.SkewNormal

The Skew Normal Distribution

## **Description**

Random generation for the Skew Normal distribution with parameters xi, omega^2 and alpha. This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If xi, omega^2 and alpha are not specified they assume the default values of 0, 1 and 0, respectively. The Skew Normal distribution with parameters xi =  $\xi$ , omega^2 =  $\omega^2$  and alpha =  $\alpha$  has density:

$$\left(\frac{2}{\omega}\right)\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\left(\frac{x-\xi}{\omega}\right)\right)$$

where  $\phi(x)$  is the standard normal probability density function and  $\Phi(x)$  is its cumulative distribution function.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

42 law0022.ScaleCont

### See Also

See law0002.Normal for the Normal distribution. See Distributions for other standard distributions

### **Examples**

```
res <- gensample(21,10000,law.pars=c(8,6,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0022.ScaleCont

The Scale Contaminated Distribution

## **Description**

Random generation for the Scale Contaminated distribution with parameters p and d.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If p or d are not specified they assume the default values of 0.5 and 0, respectively.

The Scale Contaminated distribution has density:

$$frac1\sqrt{2\pi}\left[\frac{p}{d}e^{-\frac{x^2}{2d^2}} + (1-p)e^{-\frac{x^2}{2}}\right]$$

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Distributions for other standard distributions.

```
res <- gensample(22,10000,law.pars=c(0.8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0023.GeneralizedPareto

The Generalized Pareto Distribution

### **Description**

Random generation for the Generalized Pareto distribution with parameters mu, sigma and xi.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu, sigma and xi are not specified they assume the default values of 0, 1 and 0, respectively.

The Generalized Pareto distribution with parameters  $mu = \mu$ , sigma =  $\sigma$  and  $xi = \xi$  has density:

$$\frac{1}{\sigma} \left( 1 + \frac{\xi(x-\mu)}{\sigma} \right)^{\left(-\frac{1}{\xi} - 1\right)}$$

where  $x \ge \mu$  if  $\xi \ge 0$  and  $x \le \mu - \sigma/\xi$  if  $\xi < 0$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See Distributions for other standard distributions.

```
res <- gensample(23,10000,law.pars=c(8,6,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0024.GeneralizedError

law0024.GeneralizedError

The Generalized Error Distribution

### **Description**

Random generation for the Generalized Error distribution with parameters mu, sigma and p.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu, sigma and p are not specified they assume the default values of 0, 1 and 1, respectively.

The Generalized Error distribution with parameters mu =  $\mu$ , sigma =  $\sigma$  and p = p has density:

$$\frac{1}{2p^{1/p}\Gamma(1+1/p)\sigma}\exp\left[-\frac{1}{p\sigma^p}|x-\mu|^p\right]$$

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See Distributions for other standard distributions.

```
res <- gensample(24,10000,law.pars=c(8,6,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0025.Stable 45

law0025.Stable

The Stable Distribution

# Description

Random generation for the Stable distribution with parameters stability, skewness, scale and location.

This generator is called by function gensample to create random variables based on its parameters.

### **Details**

If stability, skewness, scale and location are not specified they assume the default values of 2, 0, 1 and 0, respectively.

The Stable distribution with parameters stability =  $\alpha$ , skewness =  $\beta$ , scale = c and location =  $\mu$  doesn't have an analytically expressible probability density function, except for some parameter values. The parameters have conditions :  $0 < \alpha \le 2, -1 \le \beta \le 1$  and c > 0.

The mean of Stable distribution is defined  $\mu$  when  $\alpha > 1$ , otherwise undefined.

The variance of Stable distribution is defined  $2c^2$  when  $\alpha = 2$ , otherwise infinite.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Distributions for other standard distributions.

```
res <- gensample(25,10000,law.pars=c(2,1,1,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

46 law0026.Gumbel

law0026.Gumbel

The Gumbel Distribution

## **Description**

Random generation for the Gumbel distribution with parameters mu and sigma.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu or sigma are not specified, they assume the default values of 1.

The Gumbel distribution with parameters mu =  $\mu$  and sigma =  $\sigma$  has density:

$$\frac{1}{\sigma} \exp\left\{-\exp\left[-\left(\frac{x-\mu}{\sigma}\right)\right] - \left(\frac{x-\mu}{\sigma}\right)\right\}$$

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See law0028. GeneralizedExtValue for the Generalized Extreme Value distribution. See Distributions for other standard distributions.

```
res <- gensample(26,10000,law.pars=c(9,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0027.Frechet 47

law0027.Frechet

The Frechet Distribution

# **Description**

Random generation for the Frechet distribution with parameters mu, sigma and alpha.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu, sigma and alpha are not specified they assume the default values of 0, 1 and 1, respectively.

The Frechet distribution with parameters mu =  $\mu$ , sigma =  $\sigma$  and alpha =  $\alpha$  has density:

$$frac\alpha\sigma\left(\frac{x-\mu}{\sigma}\right)_{+}^{-\alpha-1}\exp\left\{-\left(\frac{x-\mu}{\sigma}\right)^{-\alpha}\right\}$$

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

See Distributions for other standard distributions.

```
res <- gensample(27,10000,law.pars=c(8,6,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0028.GeneralizedExtValue

The Generalized Extreme Value Distribution

### **Description**

Random generation for the Generalized Extreme Value distribution with parameters mu, sigma and xi.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu, sigma and xi are not specified they assume the default values of 0, 1 and 1, respectively.

The Generalized Extreme Value distribution with parameters mu =  $\mu$ , sigma =  $\sigma$  and xi =  $\xi$  has density:

$$[1+z]_{+}^{-\frac{1}{\xi}-1} \exp\left\{-[1+z]_{+}^{-\frac{1}{\xi}}\right\}/\sigma$$

for  $\xi > 0$  or  $\xi < 0$ , where  $z = \xi(x - \mu)/\sigma$ . If  $\xi = 0$ , PDF is as same as in the Gumbel distribution.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See law0026. Gumbel for the Gumbel distribution. See Distributions for other standard distributions.

```
res <- gensample(28,10000,law.pars=c(8,6,2))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0029.GeneralizedArcsine

The Generalized Arcsine Distribution

## **Description**

Random generation for the Generalized Arcsine distribution with parameters alpha.

This generator is called by function gensample to create random variables based on its parameter.

### **Details**

If alpha is not specified it assumes the default value of 0.5.

The Generalized Arcsine distribution with parameter alpha =  $\alpha$  has density:

$$\frac{\sin(\pi\alpha)}{\pi}x^{-\alpha}(1-x)^{\alpha-1}$$

for  $0 < \alpha < 1$ .

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See Distributions for other standard distributions.

```
res <- gensample(29,10000,law.pars=0.8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

50 law0030.FoldedNormal

law0030.FoldedNormal The Folded Normal Distribution

### **Description**

Random generation for the Folded Normal distribution with parameters mu and sigma.

This generator is called by function gensample to create random variables based on its parameters.

## **Details**

If mu and sigma are not specified they assume the default values of 0 and 1, respectively.

The Folded Normal distribution with parameters mu =  $\mu$  and sigma =  $\sigma$  has density:

```
dnorm(x, mu, sigma2) + dnorm(x, -mu, sigma2)
```

for  $x \geq 0$ .

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See law0002.Normal for the Normal distribution. See Distributions for other standard distributions.

```
res <- gensample(30,10000,law.pars=c(8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0031.MixtureNormal 51

law0031.MixtureNormal The Mixture Normal Distribution

## **Description**

Random generation for the Mixture Normal distribution with parameters p, m and d.

This generator is called by function gensample to create random variables based on its parameters.

### **Details**

If p, m and d are not specified they assume the default values of 0.5, 0 and 1, respectively.

The Mixture Normal distribution has density:

$$p\frac{1}{d\sqrt{2\pi}}e^{-\frac{(x-m)^2}{2d^2}} + (1-p)\frac{1}{d\sqrt{2\pi}}e^{-\frac{x^2}{2}}$$

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See law0002.Normal for the Normal distribution. See Distributions for other standard distributions.

```
res <- gensample(31,10000,law.pars=c(0.9,8,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0032.TruncatedNormal

The Truncated Normal Distribution

## **Description**

Random generation for the Truncated Normal distribution with parameters a and b.

This generator is called by function gensample to create random variables based on its parameters.

### **Details**

If a and b are not specified they assume the default values of 0 and 1, respectively.

The Truncated Normal distribution with parameters mu =  $\mu$  and sigma =  $\sigma$  has density:

$$\frac{\exp(-x^2/2)}{\sqrt{2\pi}(\Phi(b)-\Phi(a))}$$

for  $a \leq x \leq b$ , where  $\phi(x)$  is the standard normal probability density function and  $\Phi(x)$  is its cumulative distribution function.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See law0002.Normal for the Normal distribution. See Distributions for other standard distributions.

```
res <- gensample(32,10000,law.pars=c(2,3))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

*law0033.Nout* 53

law0033.Nout

The Normal with outliers Distribution

### **Description**

Random generation for the Normal with outliers distribution with parameter a which belongs to  $\{1,2,3,4,5\}$ .

This generator is called by function gensample to create random variables based on its parameter.

#### **Details**

If a is not specified it assumes the default value of 1.

Five cases of standard normal distributions with outliers, hereon termed Nout1 to Nout5, consisting of observations drawn from a standard normal distribution where some of the values are randomly replaced by extreme observations.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Romao, X., Delgado, R. and Costa, A. (2010), An empirical power comparison of univariate goodness-of-fit tests for normality, *Journal of Statistical Computation and Simulation*, **80**(5), 545–591.

#### See Also

See law0002.Normal for the Normal distribution. See Distributions for other standard distributions.

```
res <- gensample(33,10000,law.pars=4)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0034.GeneralizedExpPower

The Generalized Exponential Power Distribution

## **Description**

Random generation for the Generalized Exponential Power distribution with parameters t1, t2 and t3.

This generator is called by function gensample to create random variables based on its parameters.

### **Details**

If t1, t2 and t3 are not specified they assume the default value of 0.5, 0 and 1, respectively.

The Generalized Exponential Power distribution has density:

$$p(x; \gamma, \delta, \alpha, \beta, z_0) \propto e^{-\delta |x|^{\gamma} |x|^{-\alpha} (\log |x|)^{-\beta}}$$

for  $x \ge z_0$ , and the density equals to  $p(x; \gamma, \delta, \alpha, \beta, z_0)$  for  $x < z_0$ .

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A., Lafaye de Micheaux, P. and Leblanc, A. (2013), Test of Normality Against Generalized Exponential Power Alternatives, *Communications in Statistics - Theory and Methods*, **42**(1), 164–190.

#### See Also

See Distributions for other standard distributions.

```
res <- gensample(34,10000,law.pars=c(1,8,4))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0035.Exponential 55

law0035.Exponential

The Exponential Distribution

# Description

Random generation for the Exponential distribution with rate rate (i.e., mean 1/rate).

This generator is called by function gensample to create random variables based on its parameter.

### **Details**

If rate is not specified it assumes the default value of 1.

The Exponential distribution with rate =  $\lambda$  has density:

$$\lambda exp^{-\lambda x}$$

for  $x \geq 0$ .

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See Distributions for other standard distributions.

```
res <- gensample(35,10000,law.pars=8)
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0036.AsymmetricLaplace

The Asymmetric Laplace Distribution

### **Description**

Random generation for the Asymmetric Laplace distribution with parameters mu, b and k.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If mu, b or k are not specified they assume the default values of 0, 1 and 2, respectively.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See package VGAM. See Distributions for other standard distributions.

## **Examples**

```
res <- gensample(36,10000,law.pars=c(9,2,6))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0037.NormalInvGaussian

The Normal-inverse Gaussian Distribution

## **Description**

Random generation for the Normal-inverse Gaussian distribution with parameters shape, skewness, location and scale.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If shape, skewness, location and scale are not specified they assume the default values of 1, 0, 0 and 1, respectively.

The Normal-inverse Gaussian distribution with parameters shape =  $\alpha$ , skewness =  $\beta$ , location =  $\mu$  and scale =  $\delta$  has density:

$$\frac{\alpha \delta K_1(\alpha \sqrt{\delta^2 + (x - \mu)^2})}{\pi \sqrt{\delta^2 + (x - \mu)^2}} e^{\delta \gamma + \beta(x - \mu)}$$

where  $\gamma = \sqrt(\alpha^2 - \beta^2)$  and  $K_1$  denotes a modified Bessel function of the second kind.

The mean and variance of NIG are defined respectively  $\mu + \beta \delta/\gamma$  and  $\delta \alpha^2/\gamma^3$ .

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See package fBasics. See Distributions for other standard distributions.

#### **Examples**

```
res <- gensample(37,10000,law.pars=c(3,2,1,0.5))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0038.AsymmetricPowerDistribution

The Asymmetric Power Distribution

#### **Description**

Random generation for the Asymmetric Power Distribution with parameters theta, phi, alpha and lambda.

This generator is called by function gensample to create random variables based on its parameters.

### **Details**

If theta, phi, alpha and lambda are not specified they assume the default values of 0, 1, 0.5 and 2, respectively.

The Asymmetric Power Distribution with parameters theta, phi, alpha and lambda has density:

$$f(u) = \frac{1}{\phi} \frac{\delta_{\alpha,\lambda}^{1/\lambda}}{\Gamma(1+1/\lambda)} \exp\left[-\frac{\delta_{\alpha,\lambda}}{\alpha^{\lambda}} \left| \frac{u-\theta}{\phi} \right|^{\lambda}\right]$$

if

$$u \leq 0$$

and

$$f(u) = \frac{1}{\phi} \frac{\delta_{\alpha,\lambda}^{1/\lambda}}{\Gamma(1+1/\lambda)} \exp\left[-\frac{\delta_{\alpha,\lambda}}{(1-\alpha)^{\lambda}} \left| \frac{u-\theta}{\phi} \right|^{\lambda}\right]$$

if

where 
$$0 < \alpha < 1, \lambda > 0$$
 and  $\delta_{\alpha,\lambda} = \frac{2\alpha^{\lambda}(1-\alpha)^{\lambda}}{\alpha^{\lambda}+(1-\alpha)^{\lambda}}$ .

The mean and variance of APD are defined respectively by

$$E(U) = \theta + \phi \frac{\Gamma(2/\lambda)}{\Gamma(1/\lambda)} [1 - 2\alpha] \delta_{\alpha,\lambda}^{-1/\lambda}$$

and

$$V(U) = \phi^2 \frac{\Gamma(3/\lambda)\Gamma(1/\lambda)[1-3\alpha+3\alpha^2] - \Gamma(2/\lambda)^2[1-2\alpha]^2}{\Gamma^2(1/\lambda)} \delta_{\alpha,\lambda}^{-2/\lambda}.$$

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Komunjer, I. (2007), Asymmetric Power Distribution: Theory and Applications to Risk Measurement, *Journal of Applied Econometrics*, **22**, 891–921.

### See Also

See Distributions for other standard distributions.

```
res <- gensample(38,10000,law.pars=c(3,2,0.5,1))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0039.modifiedAsymmetricPowerDistribution

The modified Asymmetric Power Distribution

## Description

Random generation for the modified Asymmetric Power Distribution with parameters theta, phi, alpha and lambda.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If theta, phi, alpha and lambda are not specified they assume the default values of 0, 1, 0.5 and 2, respectively.

The modified Asymmetric Power Distribution with parameters theta, phi, theta1 and theta2 has density:

$$f(x \mid \boldsymbol{\theta}) = \frac{(\delta_{\boldsymbol{\theta}}/2)^{1/\theta_2}}{\Gamma(1+1/\theta_2)} \exp\left[-\left(\frac{2(\delta_{\boldsymbol{\theta}}/2)^{1/\theta_2}}{1+sign(x)(1-2\theta_1)}|x|\right)^{\theta_2}\right]$$

where  $\theta = (\theta_2, \theta_1)^T$  is the vector of parameters,  $\theta_2 > 0, 0 < \theta_1 < 1$  and

$$\delta_{\theta} = \frac{2(\theta_1)^{\theta_2} (1 - \theta_1)^{\theta_2}}{(\theta_1)^{\theta_2} + (1 - \theta_1)^{\theta_2}}$$

The mean and variance of APD are defined respectively by

$$E(U) = \theta + 2^{1/\theta_2} \phi \Gamma(2/\theta_2) (1 - 2\theta_1) \delta^{-1/\theta_2} / \Gamma(1/\theta_2)$$

and

$$V(U) = 2^{2/\theta_2} \phi^2 \left( \Gamma(3/\theta_2) \Gamma(1/\theta_2) (1 - 3\theta_1 + 3\theta_1^2) - \Gamma^2(2/\theta_2) (1 - 2\theta_1)^2 \right) \delta^{-2/\theta_2} / \Gamma^2(1/\theta_2).$$

#### Author(s)

P. Lafaye de Micheaux

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A. and Lafaye de Micheaux, P. and Leblanc, A. (2016), Test of normality based on alternate measures of skewness and kurtosis,

### See Also

See Distributions for other standard distributions.

#### **Examples**

```
res <- gensample(39, 10000, law.pars = c(3, 2, 0.5, 1))
res$law
res$law.pars
mean(res$sample)
sd(res$sample)</pre>
```

law0040.Log-Pareto-tail-normal

The Log-Pareto-tail-normal Distribution

### Description

Random generation for the Log-Pareto-tail-normal distribution with parameters alpha, mu and sigma.

This generator is called by function gensample to create random variables based on its parameters.

#### **Details**

If alpha, mu and sigma are not specified they assume the default values of 1.959964, 0.0 and 1.0 respectively.

The log-Pareto-tailed normal distribution has a symmetric and continuous density that belongs to the larger family of log-regularly varying distributions (see Desgagne, 2015). This is essentially a normal density with log-Pareto tails. Using this distribution instead of the usual normal ensures whole robustness to outliers in the estimation of location and scale parameters and in the estimation of parameters of a multiple linear regression.

The density of the log-Pareto-tailed normal distribution with parameters alpha, mu and sigma is given by

$$g(y \mid \alpha, \mu, \sigma) = \begin{cases} \frac{1}{\sigma} \phi\left(\frac{y - \mu}{\sigma}\right) & \text{if} \quad \mu - \alpha \sigma \leq y \leq \mu + \alpha \sigma, \\ \phi(\alpha) \frac{\alpha}{|y - \mu|} \left(\frac{\log \alpha}{\log(|y - \mu|/\sigma)}\right)^{\beta} & \text{if} \quad |y - \mu| \geq \alpha \sigma, \end{cases}$$

where  $\beta=1+2\,\phi(\alpha)\,\alpha\log(\alpha)(1-q)^{-1}$  and  $q=\Phi(\alpha)-\Phi(-\alpha)$ . The functions  $\phi(\alpha)=\frac{1}{\sqrt{2\pi}}\exp[-\frac{\alpha^2}{2}]$  and  $\Phi(\alpha)$  are respectively the p.d.f. and the c.d.f. of the standard normal distribution. The domains of the variable and the parameters are  $-\infty < y < \infty, \, \alpha > 1, \, -\infty < \mu < \infty$  and  $\alpha > 0$ 

Note that the normalizing constant  $K_{(\alpha,\beta)}$  (see Desgagne, 2015, Definition 3) has been set to 1. The desirable consequence is that the core of the density, between  $\mu - \alpha \sigma$  and  $\mu + \alpha \sigma$ , becomes exactly the density of the  $N(\mu, \sigma^2)$ . This mass of the density corresponds to q. It follows that the parameter  $\beta$  is no longer free and its value depends on  $\alpha$  as given above.

For example, if we set  $\alpha=1.959964$ , we obtain  $\beta=4.083613$  and q=0.95 of the mass is comprised between  $\mu-\alpha\sigma$  and  $\mu+\alpha\sigma$ . Note that if one is more comfortable in choosing the central mass \$q\$ instead of choosing directly the parameter  $\alpha$ , then it suffices to use the equation  $\alpha=\Phi^{-1}((1+q)/2)$ , with the contrainst  $q>0.6826895 \Leftrightarrow \alpha>1$ .

The mean and variance of Log-Pareto-tail-normal are not defined.

many.crit 61

### Author(s)

P. Lafaye de Micheaux

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, Alain. Robustness to outliers in location-scale parameter model using log-regularly varying distributions. *Ann. Statist.* **43** (2015), no. 4, 1568–1595. doi:10.1214/15-AOS1316. http://projecteuclid.org/euclid.aos/1434546215.

### See Also

See Distributions for other standard distributions.

#### **Examples**

```
res <- gensample(40, 10000, law.pars = c(1.959964, 0.0, 1.0)) res$law res$law.pars
```

many.crit

Computation of critical values for several test statistics

## **Description**

Computation of critical values for several test statistics, several n values, and several level values, for a given distribution

### Usage

```
many.crit(law.index,stat.indices,M = 10^3,vectn = c(20,50,100),levels = c(0.05,0.1),
  alter = create.alter(stat.indices),law.pars = NULL,parstats = NULL,model = NULL,
  Rlaw=NULL, Rstats = NULL,center=FALSE, scale=FALSE)
```

## Arguments

law.index	law index as given by function getindex. length(law.index)=1
stat.indices	vector of statistic indices as given by function getindex.
М	number of Monte Carlo repetitions to use.
vectn	vector of number of observations for the samples to be generated.
levels	vector of required level values.

62 many.crit

alter named-list with type of test for each statistical test: alter[["statj"]]=0, 1, 2, 3

or 4; for each j in stat.indices (0: two.sided=bilateral, 1: less=unilateral, 2: greater=unilateral, 3: bilateral test that rejects H0 only for large values of the test statistic, 4: bilateral test that rejects H0 only for small values of the test statistic)

law.pars NULL or a vector of length at most 4 containing 4 possible parameters to generate

random values from distribution law(law.pars[j], j <= 4)

parstats named-list of parameter values for each statistic to simulate. The names of the

list should be  $\mathtt{stat} j, j$  taken in  $\mathtt{stat}.\mathtt{indices}.$  If  $\mathtt{stat} j\mathtt{=}\mathtt{NA},$  the default param-

eter values for the test statistic stat j will be used.

model NOT IMPLEMENTED YET. If NULL, no model is used. If an integer i > 0,

the model coded in the C function modelei is used. Else this should be an R function that takes three arguments: eps (vector of  $\epsilon$  values), thetavec (vector of  $\theta$  values) and xvec (vector or matrix of x values). This function should take a vector of errors, generate observations from a model (with parameters thetavec and values xvec) based on these errors, then compute and return the residuals from the model. See function modele 1.R in directory inst/doc/ for an example

in multiple linear regression.

Rlaw If 'law.index' is set to 0 then 'Rlaw' should be a (random generating) function.

Rstats A list of same length as stat.indices. If a value of the vector stat.indices

is set to 0, the corresponding component of the list Rstats should be an R function that outputs a list with components statistic (value of the test statistic), pvalue (pvalue of the test; if not computable should be set to 0), decision (1 if we reject the null, 0 otherwise), alter (see above), stat.pars (see above), pvalcomp (1L if the pvalue can be computed, 0L otherwise), nbparstat (length of stat.pars). If a value of stat.indices is not 0, then the corresponding com-

ponent of Rstats should be set to NULL.

center Logical. Should we center the data generated scale Logical. Should we center the data generated

#### Value

An object of class critvalues, which is a list where each element of the list contains a matrix for the corresponding statistic. This column matrices are: n values, level values, parameters of the test statistic (NA if none), left critical values and right critical values).

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See print.critvalues for a LaTeX output of the results of this function.

many.pval 63

### **Examples**

many.pval

Computes several p-values for many test statistics.

## Description

This function generates a sample of n observations from the law specified in law.index. It then computes the value of each test statistic specified in stat.indices and use it to obtain the corresponding p-value under the null. The computation of these p-values can be done using a Monte-Carlo simulation.

### Usage

# Arguments

stat.indices	vector of test statistic indices as given by function getindex (some components can be 0 if you want to use your own function for some test statistics; see 'Rstats' argument).
law.index	index of the distribution from which to generate observations used to compute the values of the test statistics specified with stat.indices.
n	integer. Size of the samples from which to compute the value of the test statistics.
М	integer. Number of Monte-Carlo repetitions. Only used when method = 'MC'.
N	integer. Number of p-values to compute for each test statistic.
alter	integer value in $\{0,1,2,3,4\}$ . Type of test. See function create.alter.
law.pars	vector of the parameter values for the law specified in law.index.
parstats	named-list of vectors of parameters for the test statistics specified in stat.indices. The names of the list should be $\mathtt{stat} xxx$ where $xxx$ are the indices specified in $\mathtt{stat.indices}$ .
null.dist	used only if method = 'MC'. Integer value (as given by function getindex) specifying the distribution under the null hypothesis.
null.pars	vector of parameters for the null distribution

64 many.pval

method character. Either 'direct' to compute the p-value under the null using for example

the asymptotic distribution of the test statistic under the null. This is not possible for all test statistics; or 'MC' to use a Monte-Carlo simulation to approximate the distribution of the test statistic under the null (specified by null.dist).

Rlaw.index If 'law.index' is set to 0 then 'Rlaw.index' should be a (random generating)

function.

Rnull.dist If 'null.dist' is set to 0 then 'Rnull.dist' should be a (random generating) func-

tion.

Rstats A list of same length as stat.indices. If a value of the vector stat.indices

is set to 0, the corresponding component of the list Rstats should be an R function that outputs a list with components statistic (value of the test statistic), pvalue (pvalue of the test; if not computable should be set to 0), decision (1 if we reject the null, 0 otherwise), alter (see above), stat.pars (see above), pvalcomp (1L if the pvalue can be computed, 0L otherwise), nbparstat (length of stat.pars). If a value of stat.indices is not 0, then the corresponding com-

ponent of Rstats should be set to NULL.

center Logical. Should we center the data generated scale Logical. Should we center the data generated

#### Value

pvals the N x length(stat.indices) matrix of p-values.

stat.indices same as input.

n same as input.

M same as input.

alter same as input.

parstats same as input.

null.dist same as input.

method same as input.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See calcFx, graph.

moments 65

## **Examples**

moments

Expectation and variance.

## Description

Evaluate the expectation and variance of a law.

### **Details**

```
Use the function by typing: moments j(x,par1,par2,etc.) where j is the index of the law and par1, par2, etc. are the parameters of law j.
```

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Normality.tests

Goodness-of-fit tests for normality.

### **Description**

List of goodness-of-fit tests for normality.

66 Normality.tests

#### **Details**

The statistic tests for normality are named in the form stat xxxx.

For the Lilliefors statistic see stat0001.Lilliefors.

For the Anderson-Darling statistic see stat0002. AndersonDarling.

For the 1st Zhang-Wu statistic see stat0003. ZhangWu1.

For the 2nd Zhang-Wu statistic see stat0004. ZhangWu2.

For the Glen-Leemis-Barr statistic see stat0005. GlenLeemisBarr.

For the D'Agostino-Pearson statistic see stat0006.DAgostinoPearson.

For the Jarque-Bera statistic see stat0007. JarqueBera.

For the Doornik-Hansen statistic see stat0008.DoornikHansen.

For the Gel-Gastwirth statistic see stat0009. GelGastwirth.

For the 1st Hosking statistic see stat0010. Hosking1.

For the 2nd Hosking statistic see stat0011. Hosking2.

For the 3rd Hosking statistic see stat0012. Hosking3.

For the 4th Hosking statistic see stat0013. Hosking4.

For the 1st Bontemps-Meddahi statistic see stat0014.BontempsMeddahi1.

For the 2nd Bontemps-Meddahi statistic see stat0015.BontempsMeddahi2.

For the Brys-Hubert-Struyf statistic see stat0016.BrysHubertStruyf.

For the Bonett-Seier statistic see stat0017.BonettSeier.

For the Brys-Hubert-Struyf & Bonett-Seier statistic see stat0018.Brys-HubertStruyf-BonettSeier.

For the 1st Cabana-Cabana statistic see stat0019. CabanaCabana1.

For the 2nd Cabana-Cabana statistic see stat0020. CabanaCabana2.

For the Shapiro-Wilk statistic see stat0021. ShapiroWilk.

For the Shapiro-Francia statistic see stat0022. ShapiroFrancia.

For the Shapiro-Wilk statistic modified by Rahman-Govindarajulu see stat0023.ShapiroWilk-RG.

For the D'Agostino statistic see stat0024. DAgostino.

For the Filliben statistic see stat0025.Filliben.

For the Chen-Shapiro statistic see stat0026. ChenShapiro.

For the 1st Zhang statistic see stat0027. ZhangQ.

For the 2nd Zhang statistic see stat0034. ZhangQstar.

For the 3rd Zhang statistic see stat0028.ZhangQQstar.

For the Barrio-Cuesta-Matran-Rodriguez statistic see stat0029.BarrioCuestaMatranRodriguez.

For the Coin statistic see stat0030.Coin.

For the Epps-Pulley statistic see stat0031.EppsPulley.

For the Martinez-Iglewicz statistic see stat0032.MartinezIglewicz.

For the Gel-Miao-Gastwirth statistic see stat0033.GelMiaoGastwirth.

For the Desgagne-LafayeDeMicheaux-Leblanc statistic see stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn.

plot.discrepancy 67

For the new CS Desgagne-LafayeDeMicheaux statistic see stat0036. DesgagneLafayeDeMicheaux-XAPD. For the new CS Desgagne-LafayeDeMicheaux statistic see stat0037. DesgagneLafayeDeMicheaux-ZEPD. For the Spiegelhalter statistic see stat0041. Spiegelhalter.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See Laplace.tests for goodness-of-fit tests for the Laplace distribution. See Uniformity.tests for goodness-of-fit tests for uniformity.

plot.discrepancy

p-value discrepancy plot.

## Description

This function produces a *p*-value discrepancy plot.

#### **Usage**

```
## S3 method for class 'discrepancy'
plot(x,legend.pos=NULL,...)
```

# Arguments

x Fx object as returned by function calcFx.

legend.pos If NULL, position of the legend will be computed automatically. Otherwise, it

should be either a character vector in "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center". Or a numeric vector of

length 2 giving the x-y coordinates of the legend.

... further arguments passed to the plot or points functions.

#### **Details**

See Section 2.3 in Lafaye de Micheaux, P. and Tran, V. A. (2014).

### Value

No return value. Displays a graph.

68 plot.pvalue

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

```
See plot.pvalue, plot.sizepower, graph.
```

## **Examples**

plot.pvalue

p-value plot.

# Description

This function produces a p-value plot.

## Usage

```
## S3 method for class 'pvalue'
plot(x,legend.pos=NULL,...)
```

### **Arguments**

x Fx object as returned by function calcFx.

legend.pos If NULL, position of the legend will be computed automatically. Otherwise, it

should be either a character vector in "bottomright", "bottom", "bottomleft", "left", "topleft", "topright", "right" and "center". Or a numeric vector of

length 2 giving the x-y coordinates of the legend.

... further arguments passed to the plot or points functions.

### **Details**

See Section 2.3 in Lafaye de Micheaux, P. and Tran, V. A. (2014).

plot.sizepower 69

#### Value

No return value. Displays a graph.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

```
See plot.discrepancy, plot.sizepower, graph.
```

## **Examples**

plot.sizepower

size-power curves.

### Description

This function produces a size-power curves plot.

## Usage

```
## S3 method for class 'sizepower'
plot(x, xnull,legend.pos=NULL,...)
```

## **Arguments**

x Fx object as returned by function calcFx.

xnull Fx object as returned by function calcFx, but computed under the null.

legend.pos If NULL, position of the legend will be computed automatically. Otherwise, it

should be either a character vector in "bottomright", "bottom", "bottomleft", "left", "topleft", "topright", "right" and "center". Or a numeric vector of

length 2 giving the x-y coordinates of the legend.

... further arguments passed to the plot or points functions.

70 powcomp.easy

### **Details**

See Section 2.3 in Lafaye de Micheaux, P. and Tran, V. A. (2014).

#### Value

No return value. Displays a graph.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See plot.pvalue, plot.discrepancy, graph.

## **Examples**

powcomp.easy

Computation of power and level tables for hypothesis tests.

## Description

Functions for the computation of power and level tables for hypothesis tests, in LaTeX format.

## Usage

```
powcomp.easy(params,M=10^5,model=NULL,Rlaws=NULL,Rstats=NULL,center=FALSE, scale=FALSE)
```

powcomp.easy 71

#### **Arguments**

М

number of Monte Carlo repetitions to use.

params

matrix with (at least) 11 named-columns with names (n, law, stat, level, cL, cR, alter, par1, par2, par3, par4). Each row of params gives the necessary parameters for a simulation of powers.

n: sample size;

law: integer giving the index of the law considered;

stat: integer giving the index of the test statistic considered (can be 0 if you want to use your own function for some test statistic; see 'Rstats' argument);

level: double, this is the significance level desired;

cL: left critical value (can be NA); cR: right critical value (can be NA);

alter: type of test (integer value in  $\{0,1,2,3,4\}$ );

parj: values of the parameters of the distribution specified by law (can be NA).

See 'Details section'.

model

NOT YET IMPLEMENTED. If NULL, no model is used. If an integer i>0, the model coded in the C function modelei is used. Else this should be an R function that takes three arguments: eps (vector of  $\epsilon$  values), the tavec (vector of  $\theta$  values) and xvec (vector or matrix of x values). This function should take a vector of errors, generate observations from a model (with parameters the tavec and values xvec) based on these errors, then compute and return the residuals from the model. See function modele 1.R in directory inst/doc/ for an example in multiple linear regression

Rlaws

When some law indices in second column of 'params' are equal to 0, this means that you will be using some R random generators not hardcoded in C in the package. In that case, you should provide the names of the random generation functions in the corresponding components of a list; the other components should be set to NULL.

Rstats

A list. If in a given row of the 'params' matrix, the value of 'stat' is set to 0, the corresponding component of the list 'Rstats' should be an R function that outputs a list with components 'statistic' (value of the test statistic), 'pvalue' (pvalue of the test; if not computable should be set to 0), 'decision' (1 if we reject the null, 0 otherwise), 'alter' (see above), 'stat.pars' (see above), 'pvalcomp' (1L if the pvalue can be computed, 0L otherwise), 'nbparstat' (length of stat.pars). If the value of 'stat' is not 0, then the corresponding component of 'Rstats' should

be set to 'NULL'.

center Logical. Should we center the data generated scale Logical. Should we center the data generated

#### **Details**

If both cL and cR are NA, no critical values are used and the decision to reject (or not) the hypothesis is taken using the *p*-value.

If a test statistic depends upon some parameters, these can be added (in a correct order) in the last columns of params. If other test statistics are considered simultaneously (in the same params

72 powcomp.fast

matrix) and if not all the test statistics have the same number of parameters, NA values should be used to complete empty cells of the matrix.

#### Value

The powers for the different statistics and laws specified in the rows of params, NOT YET provided in the form of a LaTeX table. This version is easier to use (but slower) than the powcomp.fast version. It should be used in the process of investigating the power of test statistics under different alternatives. But when you are ready to produce results for publication in a paper, please use the powcomp.fast version and its print method..

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## **Examples**

```
# Warning: the order of the parameters of the law (4 maximum) is important! sim1 <- c(n=100,law=2,stat=10,level=0.05,cL=NA,cR=0.35,alter=3, par1= 2.0,par2=NA,par3=NA,par4=NA,parstat1=NA,parstat2=NA) sim2 <- c(n=100,law=2,stat=17,level=0.10,cL=-0.30,cR=NA,alter=1, par1=-1.0,par2=3.0,par3=NA,par4=NA,parstat1=NA,parstat2=NA) sim3 <- c(n=100,law=2,stat=31,level=0.10,cL=NA,cR=0.50,alter=3, par1=-1.0,par2=3.0,par3=NA,par4=NA,parstat1=0.7,parstat2=NA) sim4 <- c(n=100,law=7,stat=80,level=0.10,cL=NA,cR=9.319,alter=3, par1=NA,par2=NA,par3=NA,par4=NA,parstat1=1,parstat2=5) params <- rbind(sim1,sim2,sim3,sim4) powcomp.easy(params,M=10^2) sim5 <- c(n=100,law=0,stat=80,level=0.10,cL=NA,cR=9.319,alter=3, par1=NA,par2=NA,par3=NA,par4=NA,parstat1=1,parstat2=5) params <- rbind(params,sim5) powcomp.easy(params,M=10^2,Rlaws=list(NULL,NULL,NULL,NULL,rnorm))
```

powcomp.fast

Computation of power and level tables for hypothesis tests.

## **Description**

Functions for the computation of power and level tables for hypothesis tests, with possible use of a cluster.

powcomp.fast 73

### Usage

## **Arguments**

law.indices vector of law indices as given by function getindex.

stat.indices vector of statistic indices as given by function getindex (some components can

be 0 if you want to use your own function for some test statistics; see 'Rstats'

argument).

vectn vector of sample sizes (n) values.

M number of Monte Carlo repetitions.

levels vector of significance levels for the test.

critval if not NULL, a named-list of critical values for each test statistic. The names of

the list should be  $\mathtt{stat} j, j$  taken in  $\mathtt{stat.indices}$ . Note that if a single value of  $\mathtt{critval\$stat} j$  is povided, then it is the right critical value. If two values are provided, then these are the left and right critical values, in that order. If NULL,  $\mathtt{critval}$  is computed using the function  $\mathtt{many.crit};$  in that case, be sure

to provide the correct value for null.law.index.

alter named-list of integer values (0: two.sided=bilateral, 1: less=unilateral, 2: greater=unilateral,

3: bilateral test that rejects H0 only for large values of the test statistic, 4: bilateral test that rejects H0 only for small values of the test statistic). The names of

the list should be stat j, j taken in stat.indices.

parlaws named-list of parameter values for each law to simulate. The names of the list

should be  $\mathtt{law} j, j$  taken in  $\mathtt{law}.\mathtt{indices}.$  The length of vector  $\mathtt{law} j$  should not be greater than 4 (we supposed than no common distribution has more than 4

parameters!).

parstats named-list of parameter values for each statistic to simulate. The names of the

list should be statj, j taken in stat.indices (in the same order). If NULL, the

default parameter values for these statistics will be used.

nbclus number of slaves to use for the computation on a cluster. This needs parallel

or Rmpi package to be installed and functionnal on the system. Also the mpd

daemon sould be started.

model NOT YET IMPLEMENTED. If NULL, no model is used. If an integer i > 0,

the model coded in the C function modelei is used. Else this should be an R function that takes three arguments: eps (vector of  $\epsilon$  values), thetavec (vector of  $\theta$  values) and xvec (vector or matrix of x values). This function should take a vector of errors, generate observations from a model (with parameters thetavec and values xvec) based on these errors, then compute and return the residuals from the model. See function modele 1.R in directory inst/doc/ for an example

in multiple linear regression.

null.law.index index of the law under the null. Only used, by many.crit function, if critval

is NULL.

74 powcomp.fast

null.law.pars vector of parameters corresponding to null.law.index.

Rlaws When some law indices in 'law.indices' are equal to 0, this means that you will

> be using some R random generators. In that case, you should provide the names of the random generation functions in the corresponding components of 'Rlaws'

list, the other components should be set to NULL.

Rstats A list. If in a given row of the 'params' matrix, the value of 'stat' is set to 0,

> the corresponding component of the list 'Rstats' should be an R function that outputs a list with components 'statistic' (value of the test statistic), 'pvalue' (pvalue of the test; if not computable should be set to 0), 'decision' (1 if we reject the null, 0 otherwise), 'alter' (see above), 'stat.pars' (see above), 'pvalcomp' (1L if the pvalue can be computed, 0L otherwise), 'nbparstat' (length of stat.pars). If the value of 'stat' is not 0, then the corresponding component of 'Rstats' should

be set to 'NULL'.

Logical. Should we center the data generated center Logical. Should we center the data generated scale

pvalcomp Integer. 1L to compute p-values, 0L not to compute them.

#### **Details**

This version is faster (but maybe less easy to use in the process of investigating the power of test statistics under different alternatives) than the powcomp.easy version.

#### Value

cL

A list of class power whose components are described below:

number of Monte Carlo repetitions.

law.indices vector of law indices as given by function getindex.

vectn vector of sample sizes.

stat.indices vector of test statistic indices as given by function getindex.

decision a vector of counts (between 0 and M) of the decisions taken for each one of the

> levels.len \* laws.len \* vectn.len \* stats.len combinations of (level,law,sample size,test statistic), to be understood in the following sense. The decision for the l-th level (in levels), d-th law (in law.indices), n-th sample size (in vectn)

and s-th test statistic (in stat.indices) is given by:

 $\texttt{decision}[s+\texttt{stats.len*}(n-1)+\texttt{stats.len*} \lor \texttt{ctn.len*}(d-1)+\texttt{stats.len*} \lor \texttt{vectn.len*}(l-1)+\texttt{stats.len*} \lor \texttt{ctn.len*}(l-1)+\texttt{stats.len*} \lor \texttt{ctn.len*}(l-1)+\texttt{stats.len*} \lor \texttt{ctn.len*}(l-1)+\texttt{ctn.le$ 

where stats.len, vectn.len, laws.len and levels.len are respectively the lengths of the vectors stat.indices, vectn, law.indices and levels.

levels vector of levels for the test. left critical values used.

right critical values used.

usecrit a vector of 1s and 0s depending if a critical value has been used or not. power.gui 75

type of each one of the tests in stat.indices used (0: two.sided=bilateral, 1: less=unilateral, 2: greater=unilateral, 3: bilateral test that rejects H0 only for large values of the test statistic, 4: bilateral test that rejects H0 only for small values of the test statistic).

nbparlaws default number of parameters used for each law in law.indices.

parlaws default values of the parameters for each law.

nbparstats default number of parameters for each test statistic in stat.indices.

parstats default values of the parameters for each test statistic.

nbclus number of CPUs used for the simulations.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# **Examples**

power.gui

PoweR GUI

# **Description**

Graphical user interface (GUI) for the package.

## Usage

```
power.gui()
```

76 print.critvalues

### **Details**

This GUI is a 5-tabbed notebook whose goal is to make our package easier to use :

- Tab 1 gensample : generate random samples from a law added in the package;
- Tab 2 statcompute : perform the test for a given index value of test statistic;
- Tab 3 many.crit: computation of critical values for several test statistics;
- Tab 4 powcomp. fast: computation of power and level tables for hypothesis tests;
- Tab 5 Examples: reproduce results from published articles.

Important note concerning 'Iwidgets': for the GUI to work, a third party software has to be installed.

Under Microsoft Windows:

First, install ActiveTcl following indications given here: 'http://www.sciviews.org/\_rgui/tcltk/TabbedNotebook.html'

After the installation of ActiveTcl and the modification of the PATH variable, launch from an MsDOS terminal (accessible through typing 'cmd' in the Start Menu) the following command: C:\Tcl\bin\teacup.exe install Iwidgets

You can then check the existence of a directory called 'Iwidgets4.0.2' in 'C:\Tcl\lib\teapot\package\tcl\lib'.

Under Linux:

Install 'iwidgets'.

### Value

No return value. Displays a graphical user interface.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

print.critvalues

Latex table for critical values

# **Description**

Transform the critical values given by function many.crit into a LaTeX code for creating the table of critical values.

### Usage

```
## S3 method for class 'critvalues'
print(x, digits = 3, latex.output = FALSE, template = 1, ...)
```

print.critvalues 77

# Arguments

X	critical values given by function many.crit.
digits	integer indicating the number of decimal places to be used.
latex.output	logical. If TRUE, we output LaTeX code for the table of critical values. If FALSE, we output this table in the R Console.
template	integer, template to use for the (LaTeX) printing of values. Only template = 1 is defined for the moment.
	further arguments passed to or from other methods.

#### Value

No return value. The function prints a formatted representation of critical values, optionally in 'LaTeX' format. The object printed is of class "critvaluesX", where X is the template number.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Puig, P. and Stephens, M. A. (2000), Tests of fit for the Laplace distribution, with applications, *Technometrics*, **42**, 417–424.

### See Also

```
See print.power.
```

## **Examples**

78 print.power

print.power	Latex table for power simulations	

# **Description**

Transform the power values given by function powcomp.fast into a LaTeX code for creating the table of power simulations.

## Usage

```
## S3 method for class 'power'
print(x, digits = 3, latex.output = FALSE, template = 1,
summaries = TRUE, ...)
```

## **Arguments**

X	power values given by function powcomp.fast.
digits	control the number of decimal places. It can take values from 0 to 3.
latex.output	logical. If TRUE, we output LateX code for the table of power simulations. If FALSE, we output this table in the R Console.
template	integer, template to use for the $(LaTeX)$ printing of values. Only template = 1 is defined for the moment.
summaries	logical, to display the summaries Average power table, Average gap table and Worst gap table.
	further arguments passed to or from other methods.

### Value

No return value. The function prints a formatted representation of power analysis results, optionally in 'LaTeX' format. The printed object is of class "powerX", where X is the template number.

# Author(s)

```
P. Lafaye de Micheaux, V. A. Tran
```

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Puig, P. and Stephens, M. A. (2000), Tests of fit for the Laplace distribution, with applications, *Technometrics*, **42**, 417–424.

# See Also

```
See print.critvalues.
```

pvalueMC 79

## **Examples**

pvalueMC

Monte-Carlo computation of a p-value for one single test statistic.

# **Description**

This function can compute the p-value associated with a test statistic value from a sample of observations.

## Usage

### **Arguments**

data	sample of observations.
stat.index	index of a test statistic as given by function getindex.
null.law.index	index of the distribution to be tested (the null hypothesis distribution), as given by function getindex.
М	number of Monte-Carlo repetitions to use.
alter	value (in {0,1,2,3,4}) giving the the type of test (See Section 3.3 in Lafaye de Micheaux, P. and Tran, V. A. (2014)).
null.law.pars	vector of parameters for the law. The length of this parameter should not exceed 4. If not provided, the default values are taken using getindex function.
stat.pars	a vector of parameters. If NULL, the default parameter values for the statistic specified by this statistic wil be used.
list.stat	if not NULL, a vector of test statistic values should be provided. If NULL, these values will be computed.
method	method to use for the computation of the $p$ -value. Only 'Fisher' method is available for the moment.

80 stat.cstr

center	Logical. Should we center the data generated	t
scale	Logical. Should we center the data generated	d

## Value

The Monte-Carlo p-value of the test.

# Author(s)

```
P. Lafaye de Micheaux, V. A. Tran
```

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

# See Also

```
See statcompute.
```

# **Examples**

```
x <- rnorm(100)
statcompute(1,x,level = c(0.05),alter = 3)$pvalue
pvalueMC(x,stat.index = 1,null.law.index = 2,M = 10^5,alter = 3)</pre>
```

stat.cstr

Gives information about a test statistic.

# Description

To obtain the name of a test as well as its default number of parameters and default parameter values.

# Usage

```
stat.cstr(stat.index, stat.pars = NULL, n = 0)
```

# Arguments

stat.index	a single integer value corresponding to the index of a test statistic as given by function getindex.
stat.pars	vector of the values of the parameters of the test specified in ${\sf stat.index}$ . If NULL, the default values are used.
n	integer giving the sample size (useful since some default values of the parameters might depend on the sample size).

stat0001.Lilliefors 81

### Value

name of the test.

nbparams default number of parameters of the test.

law.pars values of the parameters

alter 0: two.sided=bilateral, 1: less=unilateral, 2: greater=unilateral, 3: bilateral test

that rejects H0 only for large values of the test statistic, 4: bilateral test that

rejects H0 only for small values of the test statistic.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See law.cstr, getindex, getnbparlaws, getnbparstats.

# **Examples**

stat.cstr(80)

stat0001.Lilliefors

The Lilliefors test for normality

# Description

The Lilliefors test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Lilliefors, H. (1967), On the Kolmogorov-Smirnov test for normality with mean and variance unknown, \_Journal of the American Statistical Association\_, \*62\*, 399-402.

82 stat0003.ZhangWu1

## See Also

See package nortest. See Normality. tests for other goodness-of-fit tests for normality.

stat0002.AndersonDarling

The Anderson-Darling test for normality

## **Description**

The Anderson-Darling test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

D'Agostino, R.B. and Stephens, M.A. (1986), *Goodness-of-Fit Techniques*, Marcel Dekker, New York. (Table 4.9)

### See Also

See package nortest. See Normality.tests for other goodness-of-fit tests for normality.

stat0003.ZhangWu1

The 1st Zhang-Wu test for normality

# **Description**

The 1st Zhang-Wu test Z\_C for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

stat0004.ZhangWu2 83

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, J. and Wu, Y. (2005), Likelihood-ratio tests for normality, *Computational Statistics and Data Analysis*, **49**(3), 709–721.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0004.ZhangWu2

The 2nd Zhang-Wu test for normality

## **Description**

The 2nd Zhang-Wu test Z\_A for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, J. and Wu, Y. (2005), Likelihood-ratio tests for normality, *Computational Statistics and Data Analysis*, **49**(3), 709–721.

# See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0005.GlenLeemisBarr

The Glen-Leemis-Barr test for normality

# Description

The Glen-Leemis-Barr test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Glen, A.G., Leemis, L.M. and Barr, D.R. (2001), Order Statistics in Goodness-Of-Fit Testing, *IEEE Transactions on Reliability*, **50**(2), 209–213.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0006.DAgostinoPearson

The D'Agostino-Pearson test for normality

# **Description**

The D'Agostino-Pearson for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

stat0007.JarqueBera 85

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

D'Agostino, R.B. and Pearson, E.S (1973), Tests for Departure from Normality. Empirical Results for the Distributions of b2 and  $\sqrt{b1}$ , *Biometrika*, **60**(3), 613–622.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0007.JarqueBera

The Jarque-Bera test for normality

## **Description**

The Jarque-Bera test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Jarque, C.M. and Bera, A.K. (1987), A Test for Normality of Observations and Regression Residuals, *International Statistical Review*, **50**(2), 163–172.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

86 stat0009.GelGastwirth

stat0008.DoornikHansen

The Doornik-Hansen test for normality

# Description

The Doornik-Hansen test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Doornik, J.A. and Hansen, H. (1994), *An Omnibus Test for Univariate and Multivariate Normality*, Working Paper, Nuffield College, Oxford University, U.K.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0009.GelGastwirth The Gel-Gastwirth test for normality

# **Description**

The Gel-Gastwirth test for normality is used

- to compute its statistic and p-value by calling function  ${\tt statcompute};$
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

stat0010.Hosking1

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Gel, Y. and Gastwirth, J.L. (2008), The Robust Jarque-Bera Test of Normality, *Economics Letters*, **99**(1), 30–32.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0010.Hosking1

The 1st Hosking test for normality

## **Description**

The 1st Hosking test T\_{Lmom} for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hosking, J.R.M. (1990), L-moments: analysis and estimation of distributions using linear combinations of order statistics, *Journal of the Royal Statistical Society, Series B*, **52**, 105–124.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

88 stat0012.Hosking3

stat0011.Hosking2

The 2nd Hosking test for normality

## **Description**

The 2nd Hosking test  $T_{Lmom}^{(1)}$  for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hosking, J.R.M. (1990), L-moments: analysis and estimation of distributions using linear combinations of order statistics, *Journal of the Royal Statistical Society, Series B*, **52**, 105–124.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0012.Hosking3

The 3rd Hosking test for normality

# **Description**

The 3rd Hosking test  $T_{Lmom}^{(2)}$  for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

stat0013.Hosking4

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hosking, J.R.M. (1990), L-moments: analysis and estimation of distributions using linear combinations of order statistics, *Journal of the Royal Statistical Society, Series B*, **52**, 105–124.

## See Also

Normality. tests for other goodness-of-fit tests for normality.

stat0013.Hosking4

The 4th Hosking test for normality

### **Description**

The 4th Hosking test  $T_{Lmom}^{(3)}$  for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hosking, J.R.M. (1990), L-moments: analysis and estimation of distributions using linear combinations of order statistics, *Journal of the Royal Statistical Society, Series B*, **52**, 105–124.

# See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0014.BontempsMeddahi1

The 1st Bontemps-Meddahi test for normality

# **Description**

The 1st Bontemps-Meddahi test BM\_{3-4} for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Bontemps, C. and Meddahi, N. (2005), Testing Normality: A GMM Approach, *Journal of Econometrics*, **124**, 149–186.

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0015.BontempsMeddahi2

The 2nd Bontemps-Meddahi test for normality

# **Description**

The 2nd Bontemps-Meddahi test BM\_{3-6} for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Bontemps, C. and Meddahi, N. (2005), Testing Normality: A GMM Approach, *Journal of Econometrics*, **124**, 149–186.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0016.BrysHubertStruyf

The Brys-Hubert-Struyf test for normality

## **Description**

The Brys-Hubert-Struyf test T\_{MC-LR} for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Brys, G., Hubert, M. and Struyf, A. (2008), Goodness-of-fit tests based on a robust measure of skewness, *Computational Statistics*, **23**(3), 429–442.

### See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0017.BonettSeier The Bonett-Seier test for normality

# **Description**

The Bonett-Seier test T\_w for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Bonett, D.G. and Seier, E. (2002), A test of normality with high uniform power, *Computational Statistics and Data Analysis*, **40**, 435–445.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0018. Brys Hubert Struyf-Bonett Seier

The Brys-Hubert-Struyf & Bonett-Seier test for normality

## **Description**

The combination test for normality of Brys-Hubert-Struyf & Bonett-Seier is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Brys, G., Hubert, M. and Struyf, A. (2008), Goodness-of-fit tests based on a robust measure of skewness, *Computational Statistics*, **23**(3), 429–442.

Bonett, D.G. and Seier, E. (2002), A test of normality with high uniform power, *Computational Statistics and Data Analysis*, **40**, 435–445.

### See Also

See stat0016.BrysHubertStruyf for the Brys-Hubert-Struyf test. See stat0017.BonettSeier for the Bonett-Seier test. See Normality.tests for other goodness-of-fit tests for normality.

stat0019.CabanaCabana1 93

stat0019.CabanaCabana1

The 1st Cabana-Cabana test for normality

# Description

The 1st Cabana-Cabana test T\_{S,1} for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Cabana, A. and Cabana, E. (1994), Goodness-of-Fit and Comparison Tests of the Kolmogorov-Smirnov Type for Bivariate Populations, *The Annals of Statistics*, **22**(3), 1447–1459.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0020.CabanaCabana2

The 2nd Cabana-Cabana test for normality

# **Description**

The 2nd Cabana-Cabana test T\_{K,1} for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Cabana, A. and Cabana, E. (1994), Goodness-of-Fit and Comparison Tests of the Kolmogorov-Smirnov Type for Bivariate Populations, *The Annals of Statistics*, **22**(3), 1447–1459.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0021.ShapiroWilk The Shapiro-Wilk test for normality

## **Description**

The Shapiro-Wilk test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Shapiro, S.S. and Wilk, M.B. (1965), An analysis of variance test for normality (complete samples), *Biometrika*, **52**, 591–611.

# See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0022.ShapiroFrancia

The Shapiro-Francia test for normality

# Description

The Shapiro-Francia test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Shapiro, S.S. and Francia, R. (1972), An approximation analysis of variance test for normality, *Journal of the American Statistical Association*, **67**, 215–216.

# See Also

See package nortest. See Normality.tests for other goodness-of-fit tests for normality.

```
stat0023.ShapiroWilk-RG
```

The Shapiro-Wilk test for normality modified by Rahman-Govindarajulu

# Description

The Shapiro-Wilk test for normality modified by Rahman-Govindarajulu is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

96 stat0024.DAgostino

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Rahman, M.M. and Govindarajulu, Z. (1997), A modification of the test of Shapiro and Wilk for normality, *Journal of Applied Statistics*, **24**(2), 219–236.

#### See Also

See stat0021. ShapiroWilk for the Shapiro-Wilk test. See Normality. tests for other goodness-of-fit tests for normality.

stat0024.DAgostino

The D'Agostino test for normality

# Description

The D'Agostino test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

D'Agostino, R.B. (1971), An omnibus test of normality for moderate and large size samples, *Biometrika*, **58**, 341–348.

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0025.Filliben 97

stat0025.Filliben

The Filliben test for normality

## **Description**

The Filliben test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Filliben, J.J. (1975), The Probability Plot Correlation Coefficient Test for Normality, *Technometrics*, **17**(1), 111–117.

## See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0026.ChenShapiro The Chen-Shapiro test for normality

# **Description**

The Chen-Shapiro test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

98 stat0027.ZhangQ

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Chen, L. and Shapiro, S.S (1995), An alternative test for normality based on normalized spacings, *Journal of Statistical Computation and Simulation*, **53**, 269–288.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0027.ZhangQ

The 1st Zhang test for normality

## **Description**

The 1st Zhang test Q for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, P (1999), Omnibus test of normality using the Q statistic, *Journal of Applied Statistics*, **26**(4), 519–528.

# See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0028. ZhangQQstar The 3rd Zhang test for normality

## **Description**

The 3rd Zhang test Q-Q\* for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, P (1999), Omnibus test of normality using the Q statistic, *Journal of Applied Statistics*, **26**(4), 519–528.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat 0029. Barrio Cuesta Matran Rodriguez

The Barrio-Cuesta-Matran-Rodriguez test for normality

# Description

The Barrio-Cuesta Albertos-Matran-Rodriguez test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function  ${\tt compquant}$  or  ${\tt many.crit};$
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

100 stat0030.Coin

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Barrio, E. del, Cuesta-Albertos, J., Matran, C. and Rodriguez-Rodriguez, J. (1999), Tests of goodness-of-fit based on the L\_2-Wasserstein distance, *The Annals of Statistics*, **27**, 1230–1239.

## See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0030.Coin

The Coin test for normality

## **Description**

The Coin test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Coin, D. (2008), A goodness-of-fit test for normality based on polynomial regression, *Computational Statistics and Data Analysis*, **52**, 2185–2198.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0031.EppsPulley 101

stat0031.EppsPulley

The Epps-Pulley test for normality

# Description

The Epps-Pulley test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Epps, T.W. and Pulley, L.B. (1983), A test of normality based on empirical characteristic function, *Biometrika*, **70**(3), 723–726.

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0032.MartinezIglewicz

The Martinez-Iglewicz test for normality

# Description

The Martinez-Iglewicz test for normality is used

- to compute its statistic and p-value by calling function  ${\tt statcompute};$
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Martinez, J. and Iglewicz, B. (1981), A test for departure from normality based on a biweight estimator of scale, *Biometrika*, **68**(1), 331–333.

#### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0033.GelMiaoGastwirth

The Gel-Miao-Gastwirth test for normality

# **Description**

The Gel-Miao-Gastwirth test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Gel, Y.R., Miao, W. and Gastwirth, J.L. (2007), Robust directed tests of normality against heavy-tailed alternatives, *Computational Statistics and Data Analysis*, **51**, 2734–2746.

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0034.ZhangQstar 103

stat0034.ZhangQstar

The 2nd Zhang test for normality

# Description

The 2nd Zhang test Q\* for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, P (1999), Omnibus test of normality using the Q statistic, *Journal of Applied Statistics*, **26**(4), 519–528.

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn  $\textit{The $R\_n$ test for normality}$ 

# **Description**

The Desgagne-LafayeDeMicheaux-Leblanc R\_n test for normality is used

- to compute its statistic and p-value by calling function  ${\tt statcompute};$
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function  ${\tt powcomp.fast}$  or  ${\tt powcomp.easy}$ .

## Author(s)

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A., Lafaye de Micheaux, P. and Leblanc, A. (2013), Test of Normality Against Generalized Exponential Power Alternatives, *Communications in Statistics - Theory and Methods*, **42**, 164–190.

## See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0036.DesgagneLafayeDeMicheaux-XAPD  $\textit{The $X\_APD$ test for normality}$ 

# **Description**

The  $X_{APD}$  test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A. and Lafaye de Micheaux, P. (2017), A Powerful and Interpretable Alternative to the Jarque-Bera Test of Normality Based on 2nd-Power Skewness and Kurtosis, using the Rao's score test on the APD family, *Journal of Applied Statistics*, .

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0037.DesgagneLafayeDeMicheaux-ZEPD  $\textit{The Z} \ EPD \ \textit{test for normality}$ 

# **Description**

The Desgagne-LafayeDeMicheaux Z\_{EPD} test for normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A. and Lafaye de Micheaux, P. (2017), A Powerful and Interpretable Alternative to the Jarque-Bera Test of Normality Based on 2nd-Power Skewness and Kurtosis, using the Rao's score test on the APD family, *Journal of Applied Statistics*, .

# See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0038.Glen

The Glen-Leemis-Barr test for the Laplace distribution

## **Description**

The Glen-Leemis-Barr test is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

106 stat0039.Rayner1

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Glen, A., Leemis, L., and Barr, D. (2001) Order Statistics in Goodness of Fit Testing, *IEEE Transactions on Reliability*, **50**, Number 2, pp. 209-213.

## See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0039.Rayner1

The Rayner-Best statistic for the Laplace distribution

## **Description**

The Rayner-Best statistic for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Rayner, J. C. W. and Best, D. J. (1989), Smooth Tests of Goodness of Fit, Oxford University Press, New York.

# See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0040.Rayner2 107

stat0040.Rayner2

The Rayner-Best statistic for the Laplace distribution

# **Description**

The Rayner-Best statistic for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Rayner, J. C. W. and Best, D. J. (1989), Smooth Tests of Goodness of Fit, Oxford University Press, New York.

# See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0041.Spiegelhalter

The Spiegelhalter test for normality

# **Description**

The Spiegelhalter test for normality is used

- to compute its statistic and p-value by calling function  ${\tt statcompute};$
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Spiegelhalter, D.J. (1977), A test for normality against symmetric alternatives, *Biometrika*, **64**(2), 415–418.

## See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0042.AndersonDarling

The Anderson-Darling test for the Laplace distribution

# **Description**

The Anderson-Darling test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Yen, Vincent C. and Moore, Albert H. (1988), Modified goodness-of-fit test for the laplace distribution, *Communications in Statistics - Simulation and Computation*, **17**(1), 275–281.

# See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0043.CramervonMises 109

stat0043.CramervonMises

The Cramer-von Mises test for the Laplace distribution

# **Description**

The Cramer-von Mises test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Yen, Vincent C. and Moore, Albert H. (1988), Modified goodness-of-fit test for the laplace distribution, *Communications in Statistics - Simulation and Computation*, **17**(1), 275–281.

# See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0044.Watson

The Watson test for the Laplace distribution

# **Description**

The Watson test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function  ${\tt compquant}$  or  ${\tt many.crit};$
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Puig, P. and Stephens, M. A. (2000), Tests of fit for the Laplace distribution, with applications, *Technometrics*, **42**, 417–424.

## See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0045.KolmogorovSmirnov

The Kolmogorov-Smirnov test for the Laplace distribution

# Description

The Kolmogorov-Smirnov test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Puig, P. and Stephens, M. A. (2000), Tests of fit for the Laplace distribution, with applications, *Technometrics*, **42**, 417–424.

### See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0046.Kuiper 111

stat0046.Kuiper

The Kuiper test for the Laplace distribution

## Description

The Kuiper test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Puig, P. and Stephens, M. A. (2000), Tests of fit for the Laplace distribution, with applications, *Technometrics*, **42**, 417–424.

### See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0047.Meintanis1MO The 1st Meintanis test with moment estimators for the Laplace distribution

# Description

The 1st Meintanis test  $T_{n,a}^{(1)}$  with moment estimators test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## **Details**

If a is not specified it assumes the default value of 2.

112 stat0048.Meintanis1ML

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Meintanis, S.G. (2004), A Class of Omnibus Tests for the Laplace Distribution Based on the Empirical Characteristic Function, *Communications in Statistics - Theory and Methods*, **33**(4), 925–948.

### See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0048.Meintanis1ML The 1st Meintanis test with maximum likelihood estimators for the Laplace distribution

# **Description**

The 1st Meintanis test  $T_{n,a}^{(1)}$  with maximum likelihood estimators test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### **Details**

If a is not specified it assumes the default value of 2.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Meintanis, S.G. (2004), A Class of Omnibus Tests for the Laplace Distribution Based on the Empirical Characteristic Function, *Communications in Statistics - Theory and Methods*, **33**(4), 925–948.

# See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0049.Meintanis2MO

stat0049.Meintanis2MO The 2nd Meintanis test with moment estimators for the Laplace distribution

## **Description**

The 2nd Meintanis test  $T_{n,a}^{(2)}$  with moment estimators test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### Details

If a is not specified it assumes the default value of 0.5.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Meintanis, S.G. (2004), A Class of Omnibus Tests for the Laplace Distribution Based on the Empirical Characteristic Function, *Communications in Statistics - Theory and Methods*, **33**(4), 925–948.

### See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0050.Meintanis2ML The 2nd Meintanis test with maximum likelihood estimators for the Laplace distribution

# **Description**

The 2nd Meintanis test  $T_{n,a}^{(2)}$  with maximum likelihood estimators test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

114 stat0051.ChoiKim1

### **Details**

If a is not specified it assumes the default value of 0.5.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Meintanis, S.G. (2004), A Class of Omnibus Tests for the Laplace Distribution Based on the Empirical Characteristic Function, *Communications in Statistics - Theory and Methods*, **33**(4), 925–948.

### See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0051.ChoiKim1

The 1st Choi-Kim test for the Laplace distribution

## **Description**

The 1st Choi-Kim test  $T_{m,n}^{V}$  for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### **Details**

If m is not specified it assumes the default value from the Table 4 (Choi and Kim (2006)) which produces the maximum critical values of the test statistic. Note that m < (n/2) where n is the sample size.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Choi, B. and Kim, K. (2006), Testing goodness-of-fit for Laplace distribution based on maximum entropy, *Statistics*, **40**(6), 517–531.

stat0052.ChoiKim2 115

## See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0052.ChoiKim2

The 2nd Choi-Kim test for the Laplace distribution

## **Description**

The 2nd Choi-Kim test  $T_{m,n}^{E}$  for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### **Details**

If m is not specified it assumes the default value from the Table 4 (Choi and Kim (2006)) which produces the maximum critical values of the test statistic. Note that m < (n/2) where n is the sample size.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Choi, B. and Kim, K. (2006), Testing goodness-of-fit for Laplace distribution based on maximum entropy, *Statistics*, **40**(6), 517–531.

# See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0053.ChoiKim3

The 3rd Choi-Kim test for the Laplace distribution

# **Description**

The 3rd Choi-Kim test  $T_{m,n}^{C}$  for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### **Details**

If m is not specified it assumes the default value from the Table 4 (Choi and Kim (2006)) which produces the maximum critical values of the test statistic. Note that m < (n/2) where n is the sample size.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Choi, B. and Kim, K. (2006), Testing goodness-of-fit for Laplace distribution based on maximum entropy, *Statistics*, **40**(6), 517–531.

# See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0054.DesgagneMicheauxLeblanc-Gn

The Desgagne-Micheaux-Leblanc test for the Laplace distribution

## **Description**

The Desgagne-Micheaux-Leblanc test G\_n for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

stat0055.RaynerBest1

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A., Lafaye de Micheaux, P. and Leblanc, A., unpublished document.

### See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution

stat0055.RaynerBest1 The 1st Rayner-Best test for the Laplace distribution

# Description

The 1st Rayner-Best test V\_3 for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Rayner, J. C. W. and Best, D. J. (1989), *Smooth Tests of Goodness of Fit*, Oxford University Press, New York.

# See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0056.RaynerBest2 The 2nd Rayner-Best test for the Laplace distribution

# **Description**

The 2nd Rayner-Best test V\_4 for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Rayner, J. C. W. and Best, D. J. (1989), Smooth Tests of Goodness of Fit, Oxford University Press, New York.

# See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0057.LangholzKronmal

The Langholz-Kronmal test for the Laplace distribution

# **Description**

The Langholz-Kronmal test for the Laplace distribution is used

- to compute its statistic and p-value by calling function  ${\tt statcompute};$
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

stat0058.Kundu

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Langholz, B. and Kronmal, R. A. (1991), Tests of distributional hypotheses with nuisance parameters using Fourier series, *Journal of the American Statistical Association*, **86**, 1077–1084.

#### See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0058.Kundu

The Kundu test for the Laplace distribution

# **Description**

The Kundu test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Kundu, Debasis (2005), Discriminating between Normal and Laplace distributions, *Advances in ranking and selection, multiple comparisons, and reliability*, 65-79, Stat. Ind. Technol., Birkhauser Boston, Boston, MA.

### See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

120 stat0060.Gel

stat0059.Gulati

The Gulati test for the Laplace distribution

# **Description**

The Gulati test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Gulati, Sneh (2011), Goodness of fit test for the Rayleigh and the Laplace distributions, *International Journal of Applied Mathematics and Statistics*, **24**(SI-11A), 74–85.

## See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0060.Gel

The Gel test for the Laplace distribution

# **Description**

The Gel test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Gel, Yulia R. (2010), Test of fit for a Laplace distribution against heavier tailed alternatives, *Computational Statistics and Data Analysis*, **54**(4), 958–965.

#### See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0061.DesgagneMicheauxLeblanc-Lap1

The Desgagne-Micheaux-Leblanc test for the Laplace distribution

# **Description**

The Desgagne-Micheaux-Leblanc test DLLap1 for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A., Lafaye de Micheaux, P. and Leblanc, A., unpublished document.

### See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

122 stat0063.Kolmogorov

stat0062.DesgagneMicheauxLeblanc-Lap2

The Desgagne-Micheaux-Leblanc test for the Laplace distribution

## **Description**

The Desgagne-Micheaux-Leblanc test DLLap2 for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A., Lafaye de Micheaux, P. and Leblanc, A., unpublished document.

## See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0063.Kolmogorov

The Kolmogorov test for uniformity

# **Description**

The Kolmogorov test D\_n for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

#### **Details**

Note that n is the sample size.

# Author(s)

stat0064.CramervonMises 123

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Kolmogorov, A. N. (1933), Sulla determinazione empirica di una legge di distibuziane, *Giornale dell'Istituta Italiano degli Attuari*, **4**, 83–91.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0064.CramervonMises

The Cramer-von Mises test for uniformity

# **Description**

The Cramer-von Mises test  $W_{n}^{2}$  for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### **Details**

Note that n is the sample size.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Anderson, T. W. and Darling, D. A. (1954), A test of goodness-of-fit, *Journal of the American Statistical Association*, **49**, 765–769.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

124 stat0066.Durbin

stat0065.AndersonDarling

The Anderson-Darling test for uniformity

# Description

The Anderson-Darling test A\_{n}^{2} for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## **Details**

Note that n is the sample size.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Anderson, T. W. and Darling, D. A. (1954), A test of goodness-of-fit, *Journal of the American Statistical Association*, **49**, 765–769.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0066.Durbin

The Durbin test for uniformity

# **Description**

The Durbin test C\_n for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# **Details**

Note that n is the sample size.

stat0067.Kuiper 125

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Durbin, J. (1969), Test for serial correlation in regression analysis based on the periodogram of least-squares residuals, *Biometrika*, **56**, 1–16.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0067.Kuiper

The Kuiper test for uniformity

# **Description**

The Kuiper test K\_n for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## **Details**

Note that n is the sample size.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Brunk, H. D. (1962), On the range of the difference between hypothetical distribution function and Pyke's modified empirical distribution function, *Annals of Mathematical Statistics*, **33**, 525–532.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0068. HegazyGreen1 The 1st Hegazy-Green test for uniformity

## **Description**

The 1st Hegazy-Green test T\_1 for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hegazy, Y. A. S. and Green, J. R. (1975), Some new goodness-of-fit tests using order statistics, *Applied Statistics*, **24**, 299–308.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0069. Hegazy-Green 2 The 2nd Hegazy-Green test for uniformity

# **Description**

The 2nd Hegazy-Green test T\_2 for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

stat0070.Greenwood

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hegazy, Y. A. S. and Green, J. R. (1975), Some new goodness-of-fit tests using order statistics, *Applied Statistics*, **24**, 299–308.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0070.Greenwood

The Greenwood test for uniformity

# **Description**

The Greenwood test G(n) for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### **Details**

Note that n is the sample size.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Greenwood, M. (1946), The statistical study of infectious diseases, *Journal of Royal Statistical Society Series A*, **109**, 85–110.

# See Also

Uniformity. tests for other goodness-of-fit tests for uniformity.

128 stat0072.ReadCressie

stat0071.QuesenberryMiller

The Quesenberry-Miller test for uniformity

## **Description**

The Quesenberry-Miller test Q for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Quesenberry, C. P. and Miller, F. L. Jr. (1977), Power studies of some tests for uniformity, *Journal of Statistical Computation and Simulation*, **5**, 169–191.

# See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0072.ReadCressie The Read-Cressie test for uniformity

# Description

The Read-Cressie test 2nI^{lambda} for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

#### **Details**

If  $\lambda$  is not specified it assumes the default value of 1. Note that n is the sample size.

# Author(s)

stat0073.Moran 129

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Read, Timothy R. C. and Cressie, Noel A. C. (1988), *Goodness-of-fit statistics for discrete multi-variate data*, Springer Series in Statistics. Springer-Verlag, New York.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0073.Moran

The Moran test for uniformity

# **Description**

The Moran test M(n) for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### **Details**

Note that n is the sample size.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Moran, P. A. P. (1951), The random division of an interval - Part II, *Journal of Royal Statistical Society Series B*, **13**, 147–150.

# See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0075.Cressie2

stat0074.Cressie1

The 1st Cressie test for uniformity

# Description

The 1st Cressie test  $L_{n}^{(m)}$  for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## **Details**

If m is not specified it assumes the default value of 2. Note that n is the sample size.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Cressie, N. (1978), Power results for tests based on high order gaps, Biometrika, 65, 214-218.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0075.Cressie2

The 2nd Cressie test for uniformity

# **Description**

The 2nd Cressie test  $S_{n}^{(m)}$  for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## **Details**

If m is not specified it assumes the default value of 2. Note that n is the sample size.

stat0076. Vasicek 131

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Cressie, N. (1979), An optimal statistic based on higher order gaps, Biometrika, 66, 619-627.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0076.Vasicek

The Vasicek test for uniformity

## Description

The Vasicek test H(m,n) for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## **Details**

If m is not specified it assumes the default value of 2. Note that m < (n/2) where n is the sample size.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Vasicek, O. (1976), A test for normality based on sample entropy, *Journal of the Royal Statistical Society Series B*, **38**, 54–59.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

132 stat0078.Morales

stat0077.Swartz

The Swartz test for uniformity

# **Description**

The Swartz test A^{\*}(n) for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### **Details**

Note that n is the sample size.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Swartz, T. (1992), Goodness-of-fit tests using Kullback-Leibler information, *Communications in Statistics. Theory and Methods*, **21**, 711–729.

# See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0078.Morales

The Morales test for uniformity

# **Description**

The Morales test  $D_{n,m}(phi_lambda)$  for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

stat0079.Pardo 133

### **Details**

If  $\lambda$  and m are not specified they assume the default values of 0 and 2, respectively.

There are 3 choices for value of  $\lambda$ :  $\lambda = 0$ ,  $\lambda = -1$ , and  $\lambda != 0$ , != -1.

Note that n is the sample size.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Morales, D., Pardo, L., Pardo, M. C. and Vajda, I. (2003), Limit laws for disparities of spacings, *Journal of Nonparametric Statistics*, **15**(3), 325–342.

M. A. Marhuenda, Y. Marhuenda, D. Morales, (2005), Uniformity tests under quantile categorization, *Kybernetes*, **34**(6), 888–901.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0079.Pardo

The Pardo test for uniformity

# Description

The Pardo test  $E_{m,n}$  for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# **Details**

If m is not specified it assumes the default value of 2. Note that m < (n/2) where n is the sample size.

### Author(s)

134 stat0080.Marhuenda

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Pardo, M. C. (2003), A test for uniformity based on informational energy, *Statistical Papers*, **44**, 521–534.

### See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0080.Marhuenda

The Marhuenda test for uniformity

# **Description**

The Marhuenda test  $T_{n,m}^{1}$  for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### **Details**

If  $\lambda$  and m are not specified they assume the default values of 1 and 2, respectively. Note that n is the sample size.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

M. A. Marhuenda, Y. Marhuenda, D. Morales, (2005), Uniformity tests under quantile categorization, *Kybernetes*, **34**(6), 888–901.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0081.Zhang1 135

stat0081.Zhang1

The 1st Zhang test for uniformity

## **Description**

The 1st Zhang test Z\_A for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, J. (2002), Powerful goodness-of-fit tests based on the likelihood ratio, *Journal of the Royal Statistical Society Series B*, **64**, 281–294.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0082.Zhang2

The 2nd Zhang test for uniformity

# **Description**

The 2nd Zhang test Z\_C for uniformity is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

### Author(s)

136 stat0083.ttests

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Zhang, J. (2002), Powerful goodness-of-fit tests based on the likelihood ratio, *Journal of the Royal Statistical Society Series B*, **64**, 281–294.

## See Also

See Uniformity. tests for other goodness-of-fit tests for uniformity.

stat0083.ttests

Robustness of Student's t test for non-normality (one sample)

# **Description**

Robustness of Student's t test for non-normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

An Investigation of the Large-Sample/Small-Sample Approach to the One-Sample Test for a Mean (Sigma Unknown)

# See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0085.DesgagneMicheauxLeblanc-Lap3

The Desgagne-Micheaux-Leblanc test for the Laplace distribution

## **Description**

The Desgagne-Micheaux-Leblanc test DLLap3 for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Desgagne, A., Lafaye de Micheaux, P. and Leblanc, A., unpublished document.

## See Also

See package lawstat. See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0086.Lafaye

The volcano test of normality

# **Description**

The volcano test of normality is used

- to compute its statistic and p-value by calling function  ${\tt statcompute};$
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

## Author(s)

138 stat0088.Lafaye3

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0087.Lafaye2

The volcano test of normality with alpha integrated out

# Description

The volcano test of normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0088.Lafaye3

The volcano test of normality

# **Description**

The volcano test of normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

stat0089.Lafaye4 139

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## See Also

See Normality.tests for other goodness-of-fit tests for normality.

stat0089.Lafaye4

The volcano test of normality

# Description

The volcano test of normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

140 stat0091.Gonzales1

stat0090.Lafaye5

The volcano test of normality

### Description

The volcano test of normality is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

### See Also

See Normality. tests for other goodness-of-fit tests for normality.

stat0091.Gonzales1

A ratio goodness-of-fit test for the Laplace distribution

## **Description**

A ratio goodness-of-fit test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Gonzalez-Estrada, E., Villasenor, J. A. 2016. A ratio goodness-of-fit test for the Laplace distribution. *Statistics and Probability Letters*, 119, 30-35.

stat0092.Gonzales2

## See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0092.Gonzales2

A ratio goodness-of-fit test for the Laplace distribution

## **Description**

A ratio goodness-of-fit test for the Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Gonzalez-Estrada, E., Villasenor, J. A. 2016. A ratio goodness-of-fit test for the Laplace distribution. *Statistics and Probability Letters*, 119, 30-35.

### See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0093.Hogg1

More Light on the Kurtosis and Related Statistics (for the Laplace distribution)

# **Description**

More Light on the Kurtosis and Related Statistics is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

142 stat0094.Hogg2

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hogg, R. V. 1972. More Light on the Kurtosis and Related Statistics. *Journal of the American Statistical Association*, **67(338)**, 422-424.

#### See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0094.Hogg2 More Light on the Kurtosis and Related Statistics (for the Laplace distribution)

# **Description**

More Light on the Kurtosis and Related Statistics is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hogg, R. V. 1972. More Light on the Kurtosis and Related Statistics. *Journal of the American Statistical Association*, **67(338)**, 422-424.

### See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0095.Hogg3 143

# **Description**

More Light on the Kurtosis and Related Statistics is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

### Author(s)

P. Lafaye de Micheaux, V. A. Tran

### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hogg, R. V. 1972. More Light on the Kurtosis and Related Statistics. *Journal of the American Statistical Association*, **67(338)**, 422-424.

# See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

stat0096.Hogg4	More Light on the Kurtosis and Related Statistics (for the Laplace distribution)

# Description

More Light on the Kurtosis and Related Statistics is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp.fast or powcomp.easy.

# Author(s)

144 stat0097.Rizzo

## References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Hogg, R. V. 1972. More Light on the Kurtosis and Related Statistics. *Journal of the American Statistical Association*, **67(338)**, 422-424.

#### See Also

See Laplace.tests for other goodness-of-fit tests for the Laplace distribution.

stat0097.Rizzo	Expected distances and goodness-of-fit for the asymmetric Laplace distribution

# **Description**

Expected distances and goodness-of-fit for the asymmetric Laplace distribution is used

- to compute its statistic and p-value by calling function statcompute;
- to compute its quantiles by calling function compquant or many.crit;
- to compute its power by calling function powcomp. fast or powcomp. easy.

# Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Rizzo, M. L., Haman, J. T. 2016. Expected distances and goodness-of-fit for the asymmetric Laplace distribution. *Statist. Probab. Lett.*, **117**, 158-164.

### See Also

See Laplace. tests for other goodness-of-fit tests for the Laplace distribution.

statcompute 145

atcompute
atcompute

#### **Description**

Performs the hypothesis test for those added in the package.

## Usage

#### **Arguments**

stat.index one statistic index as given by function getindex. data sample from which to compute the statistic. levels vector of desired significance levels for the test. critvalL NULL or vector of left critival values. critvalR NULL or vector of right critival values. alter 0: two.sided=bilateral, 1: less=unilateral, 2: greater=unilateral, 3: bilateral test that rejects H0 only for large values of the test statistic, 4: bilateral test that rejects H0 only for small values of the test statistic. a vector of parameters. If NULL, the default parameter values for this statistic stat.pars will be used. pvalcomp 1L to compute the p-value, 0L otherwise. check Logical. If FALSE it will execute much faster, but in this case be sure to give a value to the 'stat.pars' argument; if you don't know what value to give, use rep(0.0, getnbparstats(stat.index)) as a default value.

#### **Details**

The function statcompute() should not be used in simulations since it is NOT fast. Consider instead using powcomp.easy or powcomp.fast. See also in the Example section below for a fast approach using the .C function (but be warned that giving wrong values of arguments can crash your session!).

#### Value

# A list with components:

statistic the test statistic value pvalue the p-value decision the vector of decisions, same length as levels alter alter stat.pars stat.pars symbol how the test is noted

146 testsPureR

#### Author(s)

P. Lafaye de Micheaux, V. A. Tran

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

## **Examples**

```
data <- rnorm(50)
# Shapiro-Wilk test:
statcompute(21, data, levels = c(0.05, 0.1), critvalL = NULL, critvalR = NULL,
            alter = 0, stat.pars = NULL)
# Identical to:
shapiro.test(data)
# The function statcompute() should not be used in simulations since it
# is NOT fast. Consider instead the call below (but see the Details
# Section):
.C("stat21", data = data, n = 50L, levels = 0.05, nblevels = 1L, name =
rep(" ", 50), getname = 0L, statistic = 0, pvalcomp = 1L, pvalue = 0, cL = 0.0,
cR = 0.0, usecrit = 0L, alter = 4L, decision = 0L, stat.pars = 0.0,
nbparstat = 0L)
# Another option is to use the 'pvalcomp' and 'check' arguments as
# follows which can be much faster (when computing the p-value takes time)
statcompute(21, data, levels = c(0.05, 0.1), critvalL = NULL, critvalR = NULL,
            alter = 0, stat.pars = NULL, pvalcomp = 0L, check = FALSE)
```

testsPureR

Computation of test statistic values in pure R.

## **Description**

Alternate way to compute test statistic values (only) in pure R instead of C/C++, for clarity reasons.

### **Details**

```
Use the function by typing: {\tt stat} j({\tt x,par1,par2}, etc.) where j is the index of the test and par1, par2, etc. are the parameters of test j, if any.
```

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

Uniformity.tests 147

#### References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

Uniformity.tests

Goodness-of-fit tests for uniformity

# **Description**

List of goodness-of-fit tests for uniformity.

#### **Details**

The statistic tests for uniformity are named in the form statxxxx.

For the Kolmogorov statistic see stat0063.Kolmogorov.

For the Cramer-von Mises statistic see stat0064. CramervonMises.

For the Anderson-Darling statistic see stat0065. AndersonDarling.

For the Durbin statistic see stat0066. Durbin.

For the Kuiper statistic see stat0067. Kuiper.

For the 1st Hegazy-Green statistic see stat0068. HegazyGreen1.

For the 2nd Hegazy-Green statistic see stat0069. HegazyGreen2.

For the Greenwood statistic see stat0070. Greenwood.

For the Quesenberry-Miller statistic see stat0071.QuesenberryMiller.

For the Read-Cressie statistic see stat0072.ReadCressie.

For the Moran statistic see stat0073. Moran.

For the 1st Cressie statistic see stat0074. Cressie1.

For the 2nd Cressie statistic see stat0075. Cressie2.

For the Vasicek statistic see stat0076. Vasicek.

For the Swartz statistic see stat0077. Swartz.

For the Morales statistic see stat0078. Morales.

For the Pardo statistic see stat0079. Pardo.

For the Marhuenda statistic see stat0080. Marhuenda.

For the 1st Zhang statistic see stat0081. Zhang1.

For the 2nd Zhang statistic see stat0082. Zhang2.

## Author(s)

P. Lafaye de Micheaux, V. A. Tran

# References

Pierre Lafaye de Micheaux, Viet Anh Tran (2016). PoweR: A Reproducible Research Tool to Ease Monte Carlo Power Simulation Studies for Studies for Goodness-of-fit Tests in R. *Journal of Statistical Software*, **69(3)**, 1–42. doi:10.18637/jss.v069.i03

#### See Also

See Normality.tests for goodness-of-fit tests for normality. See Laplace.tests for goodness-of-fit tests for the Laplace distribution.

# **Index**

* Anderson-Darling	stat0053.ChoiKim3,116
stat0002.AndersonDarling,82	* Coin
stat0042.AndersonDarling, 108	stat0030.Coin, 100
stat0065.AndersonDarling, 124	* Cramer-von Mises
* Asymmetric Laplace	stat0043.CramervonMises, 109
law0036.AsymmetricLaplace, 56	stat0064.CramervonMises, 123
* Asymmetric Power Distribution	* Cressie
law0038.AsymmetricPowerDistribution,	stat0074.Cressie1, 130
57	stat0075.Cressie2, 130
* Average Uniform	* D'Agostino-Pearson
law0014.AverageUnif,35	stat0006.DAgostinoPearson,84
* Barrio-Cuesta-Matran-Rodriguez	* D'Agostino
stat0029.BarrioCuestaMatranRodriguez,	stat0024.DAgostino,96
99	* Desgagne-LafayeDeMicheaux-Leblanc
* Beta	stat0035.DesgagneLafayeDeMicheauxLeblanc-Rr
law0006.Beta, 27	103
* Bonett-Seier	* Desgagne-LafayeDeMicheaux-ZEPD
stat0017.BonettSeier,91	stat0037.DesgagneLafayeDeMicheaux-ZEPD,
stat0018.BrysHubertStruyf-BonettSeier,	105
92	* Desgagne-LafayeDeMicheaux
* Bontemps-Meddahi	stat0036.DesgagneLafayeDeMicheaux-XAPD,
${\sf stat0014.BontempsMeddahi1}, {\sf 90}$	104
stat0015.BontempsMeddahi2,90	* Desgagne-Micheaux-Leblanc
* Brys-Hubert-Struyf	stat0054.DesgagneMicheauxLeblanc-Gn,
stat0016.BrysHubertStruyf, 91	116
stat0018.BrysHubertStruyf-BonettSeier, 92	<pre>stat0061.DesgagneMicheauxLeblanc-Lap1,</pre>
* Cabana-Cabana	stat0062.DesgagneMicheauxLeblanc-Lap2,
stat0019.CabanaCabana1,93	122
stat0020.CabanaCabana2,93	stat0085.DesgagneMicheauxLeblanc-Lap3,
* Cauchy	137
law0003.Cauchy, 24	* Doornik-Hansen
* Chen-Shapiro	stat0008.DoornikHansen,86
stat0026.ChenShapiro,97	* Durbin
* Chi-Squared	stat0066.Durbin, 124
law0009.Chisquared,30	* Epps-Pulley
* Choi-Kim	stat0031.EppsPulley, 101
stat0051.ChoiKim1,114	* Exponential
stat0052.ChoiKim2, 115	law0035.Exponential, 55

* Filliben	* Kolmogorov-Smirnov
stat0025.Filliben,97	stat0045.KolmogorovSmirnov, 110
* Folded Normal	* Kolmogorov
law0030.FoldedNormal, 50	stat0063.Kolmogorov, 122
* Frechet	* Kuiper
law0027.Frechet, 47	stat0046.Kuiper, 111
* Gamma	stat0067.Kuiper, 125
law0005.Gamma, 26	* Kundu
* Gaussian	stat0058.Kundu, 119
law0002.Normal, 23	* Langholz-Kronmal
* Gel-Gastwirth	stat0057.LangholzKronmal, 118
stat0009.GelGastwirth,86	* Laplace
* Gel-Miao-Gastwirth	law0001.Laplace, 22
stat0033.GelMiaoGastwirth, 102	stat0038.Glen, <u>105</u>
* Gel	stat0039.Rayner1, <u>106</u>
stat0060.Gel, <u>120</u>	stat0040.Rayner2, 107
* Generalized Arcsine	stat0042.AndersonDarling, $108$
law0029.GeneralizedArcsine,49	stat0043.CramervonMises, 109
* Generalized Error	stat0044.Watson, 109
law0024.GeneralizedError,44	stat0045.KolmogorovSmirnov,110
* Generalized Exponential Power	stat0046.Kuiper, 111
law0034.GeneralizedExpPower, 54	stat0047.Meintanis1MO, 111
* Generalized Extreme Value	stat0048.Meintanis1ML, 112
law0028.GeneralizedExtValue,48	stat0049.Meintanis2MO, 113
* Generalized Pareto	stat0050.Meintanis2ML, 113
law0023.GeneralizedPareto,43	stat0051.ChoiKim1,114
* Glen-Leemis-Barr	stat0052.ChoiKim2,115
stat0005.GlenLeemisBarr,84	stat0053.ChoiKim3,116
* Greenwood	stat0054.DesgagneMicheauxLeblanc-Gn,
stat0070.Greenwood, 127	116
* Gulati	stat0055.RaynerBest1, 117
stat0059.Gulati, 120	stat0056.RaynerBest2, 118
* Gumbel	stat0057.LangholzKronmal, 118
law0026.Gumbel, 46	stat0058.Kundu, 119
* Hegazy-Green	stat0059.Gulati, 120
stat0068.HegazyGreen1, 126	stat0060.Gel, 120
stat0069.HegazyGreen2, 126	stat0061.DesgagneMicheauxLeblanc-Lap1,
* Hosking	121
stat0010.Hosking1,87	stat0062.DesgagneMicheauxLeblanc-Lap2,
stat0011.Hosking2,88	122
stat0012.Hosking3,88	stat0085.DesgagneMicheauxLeblanc-Lap3,
stat0013.Hosking4,89	137
* Jarque-Bera	stat0091.Gonzales1, 140
stat0007.JarqueBera, 85	stat0092.Gonzales2, 141
* Johnson SB	stat0093.Hogg1, 141
law0020. JohnsonSB, 40	stat0094.Hogg2, 142
* Johnson SU	stat0095.Hogg3, 143
law0017.JohnsonSU, 38	stat0096.Hogg4, <u>143</u>

stat0097.Rizzo, 144	* Read-Cressie
* Lilliefors	stat0072.ReadCressie, 128
stat0001.Lilliefors, 81	* Scale Contaminated
* Location Contaminated	law0022.ScaleCont, 42
law0019.LocationCont, 39	* Shapiro-Francia
* Log Normal	stat0022.ShapiroFrancia,95
law0010.LogNormal,31	* Shapiro-Wilk
* Log-Pareto-tail-normal	stat0021.ShapiroWilk,94
law0040.Log-Pareto-tail-normal, 60	stat0023.ShapiroWilk-RG, 95
* Logistic	* Shifted Exponential
law0004.Logistic, 25	law0012.ShiftedExp, 33
* Marhuenda	* Skew Normal
stat0080.Marhuenda, 134	law0021.SkewNormal,41
* Martinez-Iglewicz	* Spiegelhalter
stat0032.MartinezIglewicz, 101	stat0041.Spiegelhalter, 107
* Meintanis	* Stable
stat0047.Meintanis1MO, 111	law0025.Stable, 45
stat0048.Meintanis1ML, 112	* Student
stat0049.Meintanis2MO, 113	law0008.Student, 29
stat0050.Meintanis2ML, 113	* Swartz
* Mixture Normal	stat0077.Swartz, 132
law0031.MixtureNormal, 51	* Truncated Normal
* Morales	law0032.TruncatedNormal, 52
stat0078.Morales, 132	* Tukey
* Moran	law0018.Tukey, 39
stat0073.Moran, 129	* UUniform
* Normal with outliers	law0015.UUnif, 36
law0033.Nout, 53	* Uniform
* Normal-inverse Gaussian	law0007.Uniform, 28
law0037.NormalInvGaussian, 56	* VUniform
* Normal	law0016.VUnif, 37
law0002.Normal, 23	* Vasicek
law0021.SkewNormal,41	stat0076.Vasicek, 131
law0030.FoldedNormal, 50	* Watson
law0031.MixtureNormal, 51	stat0044.Watson, 109
law0032.TruncatedNormal, 52	* Weibull
law0033.Nout, 53	law0011.Weibull, 32
* Pardo	* Zhang-Wu
stat0079.Pardo, 133	stat0003.ZhangWu1,82
* Power Uniform	stat0004.ZhangWu2,83
law0013.PowerUnif, 34	* Zhang
* Quesenberry-Miller	stat0027.ZhangQ, 98
stat0071.QuesenberryMiller, 128	stat0028.ZhangQQstar,99
* Rahman-Govindarajulu	stat0034.ZhangQstar, 103
stat0023.ShapiroWilk-RG,95	stat0081.Zhang1, 135
* Rayner-Best	stat0082.Zhang2, 135
stat0055.RaynerBest1, 117	* datagen
stat0056.RaynerBest2, 118	compquant, 8

Distributions, 11	law0017.JohnsonSU, 38
gensample, 12	law0018.Tukey, 39
powcomp.easy, 70	law0019.LocationCont, 39
powcomp.fast,72	law0020.JohnsonSB, 40
* distributions	law0021.SkewNormal,41
densities, 10	law0022.ScaleCont,42
Distributions, 11	law0023.GeneralizedPareto,43
moments, 65	law0024.GeneralizedError,44
* distribution	law0025.Stable, 45
checklaw, 6	law0026.Gumbel, 46
compquant, 8	law0027.Frechet, 47
gensample, 12	law0028.GeneralizedExtValue,48
* documentation	law0029.GeneralizedArcsine,49
help.law, 18	law0030.FoldedNormal, 50
help.stat, 19	law0031.MixtureNormal, 51
Laplace.tests, 19	law0032.TruncatedNormal, 52
Normality.tests, 65	law0033.Nout, 53
Uniformity.tests, 147	law0034.GeneralizedExpPower,54
* dplot	law0035.Exponential, 55
graph, 17	law0036.AsymmetricLaplace, 56
* hplot	law0037.NormalInvGaussian,56
graph, 17	law0038.AsymmetricPowerDistribution,
plot.discrepancy,67	57
plot.pvalue, 68	law 0039. modified A symmetric Power Distribution,
plot.sizepower,69	59
* htest	law0040.Log-Pareto-tail-normal, $60$
many.pval, 63	* modified Asymmetric Power Distribution
powcomp.easy, $70$	law 0039. modified A symmetric Power Distribution,
powcomp.fast, 72	59
pvalueMC, 79	* nonparametric
statcompute, 145	calcFx, 5
* law	* normality
law0001.Laplace, 22	stat0001.Lilliefors, 81
law0002.Normal, 23	stat0002.AndersonDarling,82
law0003.Cauchy, 24	stat0003.ZhangWu1,82
law0004.Logistic, 25	stat0004.ZhangWu2,83
law0005.Gamma, 26	stat0005.GlenLeemisBarr,84
law0006.Beta, 27	stat0006.DAgostinoPearson,84
law0007.Uniform, 28	stat0007.JarqueBera,85
law0008.Student, 29	stat0008.DoornikHansen,86
law0009.Chisquared,30	stat0009.GelGastwirth,86
law0010.LogNormal, 31	stat0010.Hosking1,87
law0011.Weibull, 32	stat0011.Hosking2,88
law0012.ShiftedExp, 33	stat0012.Hosking3,88
law0013.PowerUnif, 34	stat0013.Hosking4,89
law0014.AverageUnif,35	${\sf stat0014.BontempsMeddahi1,90}$
law0015.UUnif, 36	stat0015.BontempsMeddahi2,90
law0016.VUnif, 37	stat0016.BrysHubertStruyf, 91

stat0017.BonettSeier,91	stat0010.Hosking1,87
stat0018.BrysHubertStruyf-BonettSeier,	stat0011.Hosking2,88
92	stat0012.Hosking3,88
stat0019.CabanaCabana1,93	stat0013.Hosking4,89
stat0020.CabanaCabana2,93	stat0014.BontempsMeddahi1,90
stat0021.ShapiroWilk,94	stat0015.BontempsMeddahi2,90
stat0022.ShapiroFrancia,95	stat0016.BrysHubertStruyf,91
stat0023.ShapiroWilk-RG,95	stat0017.BonettSeier, 91
stat0024.DAgostino,96	stat0018.BrysHubertStruyf-BonettSeier,
stat0025.Filliben,97	92
stat0026.ChenShapiro,97	stat0019.CabanaCabana1, 93
stat0027.ZhangQ,98	stat0020.CabanaCabana2,93
stat0028.ZhangQQstar,99	stat0021.ShapiroWilk,94
stat0029.BarrioCuestaMatranRodriguez,	stat0022.ShapiroFrancia,95
99	stat0023.ShapiroWilk-RG,95
stat0030.Coin, 100	stat0024.DAgostino,96
stat0031.EppsPulley, 101	stat0025.Filliben, 97
stat0032.MartinezIglewicz, 101	stat0026.ChenShapiro,97
stat0033.GelMiaoGastwirth, 102	stat0027.ZhangQ, 98
stat0034.ZhangQstar, 103	stat0028.ZhangQQstar,99
stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn,	stat0029.BarrioCuestaMatranRodriguez,
103	99
stat0036.DesgagneLafayeDeMicheaux-XAPD,	stat0030.Coin, 100
104	stat0031.EppsPulley, 101
stat0037.DesgagneLafayeDeMicheaux-ZEPD,	stat0032.MartinezIglewicz, 101
105	stat0033.GelMiaoGastwirth, 102
stat0038.Glen, 105	stat0034.ZhangQstar, 103
stat0039.Rayner1, 106	stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn,
stat0041.Spiegelhalter, 107	103
stat0083.ttests, 136	stat0036.DesgagneLafayeDeMicheaux-XAPD,
stat0086.Lafaye, 137	104
stat0087.Lafaye2, 138	stat0037.DesgagneLafayeDeMicheaux-ZEPD,
stat0088.Lafaye3, 138	105
stat0089.Lafaye4, 139	stat0038.Glen, 105
stat0090.Lafaye5, 140	stat0039.Rayner1, 106
* print	stat0040.Rayner2, 107
print.critvalues,76	stat0041.Spiegelhalter, 107
print.power, 78	stat0042.AndersonDarling, 108
* statistic	stat0043.CramervonMises, 109
stat0001.Lilliefors, 81	stat0044.Watson, 109
stat0002.AndersonDarling,82	stat0045.KolmogorovSmirnov,110
stat0003.ZhangWu1,82	stat0046.Kuiper, 111
stat0004.ZhangWu2,83	stat0047.Meintanis1MO, 111
stat0005.GlenLeemisBarr,84	stat0048.Meintanis1ML, 112
stat0006.DAgostinoPearson,84	stat0049.Meintanis2MO, 113
stat0007.JarqueBera, 85	stat0050.Meintanis2ML, 113
stat0008.DoornikHansen,86	stat0051.ChoiKim1, 114
stat0009.GelGastwirth,86	stat0052.ChoiKim2, 115

* stats
testsPureR, 146
* test
stat0001.Lilliefors, 81
stat0002.AndersonDarling,82
stat0003.ZhangWu1,82
stat0004.ZhangWu2,83
stat0005.GlenLeemisBarr,84
stat0006.DAgostinoPearson,84
stat0007.JarqueBera,85
stat0008.DoornikHansen,86
stat0009.GelGastwirth,86
stat0010.Hosking1,87
stat0011.Hosking2,88
stat0012.Hosking3,88
stat0013.Hosking4,89
stat0014.BontempsMeddahi1,90
stat0015.BontempsMeddahi2,90
stat0016.BrysHubertStruyf, 91
stat0017.BonettSeier, 91
stat0018.BrysHubertStruyf-BonettSeier,
92
stat0019.CabanaCabana1,93
stat0020.CabanaCabana2,93
stat0021.ShapiroWilk,94
stat0022.ShapiroFrancia,95
stat0023.ShapiroWilk-RG,95
stat0024.DAgostino,96
stat0025.Filliben, 97
stat0026.ChenShapiro,97
stat0027.ZhangQ, 98
stat0028.ZhangQQstar,99
stat0029.BarrioCuestaMatranRodriguez,
99
stat0030.Coin, 100
stat0031.EppsPulley, 101
stat0032.MartinezIglewicz, 101
stat0033.GelMiaoGastwirth, 102
stat0034.ZhangQstar, 103
stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn,
103
stat0036.DesgagneLafayeDeMicheaux-XAPD,
104
stat0037.DesgagneLafayeDeMicheaux-ZEPD,
105
stat0038.Glen, 105
stat0039.Rayner1, 106
stat0040.Rayner2, 107

stat0041.Spiegelhalter, 107	stat0086.Lafaye, 137
stat0042.AndersonDarling, 108	stat0087.Lafaye2, 138
stat0043.CramervonMises, 109	stat0088.Lafaye3, 138
stat0044.Watson, 109	stat0089.Lafaye4, 139
stat0045.KolmogorovSmirnov, 110	stat0090.Lafaye5, 140
stat0046.Kuiper, 111	stat0091.Gonzales1, 140
stat0047.Meintanis1MO, 111	stat0092.Gonzales2, 141
stat0048.Meintanis1ML, 112	stat0093.Hogg1, 141
stat0049.Meintanis2MO, 113	stat0094.Hogg2, 142
stat0050.Meintanis2ML, 113	stat0095.Hogg3, 143
stat0051.ChoiKim1, 114	stat0096.Hogg4, 143
stat0052.ChoiKim2, 115	stat0097.Rizzo,144
stat0053.ChoiKim3, 116	* uniformity
stat0054.DesgagneMicheauxLeblanc-Gn,	stat0063.Kolmogorov, 122
116	stat0064.CramervonMises, 123
stat0055.RaynerBest1, 117	stat0065.AndersonDarling, 124
stat0056.RaynerBest2, 118	stat0066.Durbin,124
stat0057.LangholzKronmal, 118	stat0067.Kuiper, 125
stat0058.Kundu, 119	stat0068.HegazyGreen1, 126
stat0059.Gulati, 120	stat0069.HegazyGreen2, 126
stat0060.Gel, 120	stat0070.Greenwood, 127
stat0061.DesgagneMicheauxLeblanc-Lap1,	stat0071.QuesenberryMiller, 128
121	stat0072.ReadCressie, 128
stat0062.DesgagneMicheauxLeblanc-Lap2,	stat0073.Moran, 129
122	stat0074.Cressie1,130
stat0063.Kolmogorov, 122	stat0075.Cressie2,130
stat0064.CramervonMises, 123	stat0076.Vasicek, 131
stat0065.AndersonDarling, 124	stat0077.Swartz, 132
stat0066.Durbin, 124	stat0078.Morales, 132
stat0007.Kuiper, 125	stat0079.Pardo, 133
stat0068.HegazyGreen1, 126	stat0080.Marhuenda, 134
	stat0081.Zhang1, 135
stat0069. HegazyGreen2, 126	stat0082.Zhang2, 135
stat0070. Greenwood, 127	stat0083.ttests, 136
stat0071.QuesenberryMiller, 128	* univar
stat0072.ReadCressie, 128	many.crit,61
stat0073.Moran, 129	pvalueMC, 79
stat0074.Cressie1, 130	* utilities
stat0075.Cressie2, 130	checklaw, 6
stat0076.Vasicek, 131	create.alter,9
stat0077. Swartz, 132	getindex, 14
stat0078.Morales, 132	getnbparlaws, 15
stat0079.Pardo, 133	getnbparstats, 16
stat0080.Marhuenda, 134	law.cstr, 21
stat0081.Zhang1, 135	stat.cstr, $80$
stat0082.Zhang2, 135	
stat0083.ttests, 136	calcFx, 5, 64, 67–69
stat0085.DesgagneMicheauxLeblanc-Lap3,	checklaw, 6, 13, 14
137	compquant, 8, 14, 81-91, 93-144

create.alter, 9, 63	getindex, 7–10, 13, 14, 15, 16, 18, 19, 21, 22, 61, 63, 73, 74, 79–81, 145
densities, 10	getnbparlaws, 14, 15, 16, 22, 81
Distributions, 11, 18, 22–59, 61	getnbparstats, 14, 15, 16, 22, 81
dlaw1 (densities), 10	graph, 17, 64, 68–70
dlaw10 (densities), 10	
dlaw11 (densities), 10	help.law, 18
dlaw12 (densities), 10	help.stat, 19
dlaw13 (densities), 10	
dlaw14 (densities), 10	indicator (densities), 10
dlaw15 (densities), 10	10 10 67 106 122 127
dlaw16 (densities), 10	Laplace.tests, 19, 19, 67, 106–122, 137,
dlaw17 (densities), 10	141–144, 148
dlaw18 (densities), 10	law.cstr, 14–16, 21, 81
dlaw19 (densities), 10	law0001 (law0001.Laplace), 22
dlaw2 (densities), 10	law0001.Laplace, 11, 22
dlaw20 (densities), 10	law0002 (law0002.Normal), 23
dlaw21 (densities), 10	law0002.Normal, 11, 23, 42, 50–53
dlaw22 (densities), 10	law0003 (law0003.Cauchy), 24
dlaw23 (densities), 10	law0003. Cauchy, 11, 24
dlaw24 (densities), 10	law0004 (law0004.Logistic), 25
dlaw25 (densities), 10	law0004.Logistic, <i>11</i> , 25
dlaw26 (densities), 10	law0005 (law0005.Gamma), 26
dlaw27 (densities), 10	law0005.Gamma, <i>11</i> , 26
	law0006 (law0006.Beta), 27
dlaw28 (densities), 10	law0006.Beta, <i>11</i> , 27
dlaw29 (densities), 10	law0007 (law0007.Uniform), 28
dlaw3 (densities), 10	law0007.Uniform, <i>11</i> , 28, <i>34–37</i>
dlaw30 (densities), 10	law0008 (law0008.Student), 29
dlaw31 (densities), 10	law0008.Student, <i>11</i> , 29
dlaw32 (densities), 10	law0009 (law0009.Chisquared), 30
dlaw33 (densities), 10	law0009.Chisquared, <i>11</i> , 30
dlaw34 (densities), 10	law0010(law0010.LogNormal), 31
dlaw35 (densities), 10	law0010.LogNormal, <i>11</i> , 31
dlaw36 (densities), 10	law0011 (law0011.Weibull), 32
dlaw37 (densities), 10	law0011.Weibull, <i>11</i> , 32
dlaw38 (densities), 10	law0012 (law0012.ShiftedExp), 33
dlaw39 (densities), 10	law0012.ShiftedExp, <i>11</i> , 33
dlaw4 (densities), 10	law0013 (law0013.PowerUnif), 34
dlaw40 (densities), 10	law0013.PowerUnif, <i>11</i> , 34
dlaw41 (densities), 10	law0014(law0014.AverageUnif), 35
dlaw42 (densities), 10	law0014.AverageUnif, <i>11</i> , 35
dlaw5 (densities), 10	law0015(law0015.UUnif), 36
dlaw6 (densities), 10	law0015.UUnif, <i>11</i> , 36
dlaw7 (densities), 10	law0016(law0016.VUnif), 37
dlaw8 (densities), 10	law0016.VUnif, <i>11</i> , 37
dlaw9 (densities), 10	law0017 (law0017. JohnsonSU), 38
	law0017.JohnsonSU, <i>11</i> , 38
gensample, 7, 12, 14, 22–57, 59, 60, 76	law0018 (law0018. Tukey), 39

law0018.Tukey, 11, 39	59
law0019 (law0019.LocationCont), 39	law0039.modifiedAsymmetricPowerDistribution,
law0019.LocationCont, <i>11</i> , 39	12, 59
law0020 (law0020. JohnsonSB), 40	law0040
law0020. JohnsonSB, <i>11</i> , 40	<pre>(law0040.Log-Pareto-tail-normal),</pre>
law0021 (law0021.SkewNormal), 41	60
law0021.SkewNormal, <i>11</i> , 41	law0040.Log-Pareto-tail-normal, 60
law0022 (law0022.ScaleCont), 42	
law0022.ScaleCont, 11, 42	many.crit, 61, 73, 76, 77, 81-91, 93-144
law0023 (law0023.GeneralizedPareto), 43	many.critR(many.crit), 61
law0023.GeneralizedPareto, 11, 43	many.pval, <i>5</i> , <i>6</i> , <i>17</i> , 63
law0024 (law0024.GeneralizedError), 44	moments, 65
law0024.GeneralizedError, 11, 44	moments1 (moments), 65
law0025 (law0025.Stable), 45	moments10 (moments), 65
law0025. Stable, 11, 45	moments11 (moments), 65
law0026 (law0026.Gumbel), 46	moments12 (moments), 65
law0026.Gumbel, 11, 46, 48	moments13 (moments), 65
law0027 (law0027.Frechet), 47	moments14 (moments), 65
law0027.Frechet, 11, 47	moments15 (moments), 65
law0028 (law0028.GeneralizedExtValue),	moments16 (moments), 65
48	moments17 (moments), 65
law0028.GeneralizedExtValue, 12, 46, 48	moments18 (moments), 65
law0029 (law0029. GeneralizedArcsine), 49	moments19 (moments), 65
	moments2 (moments), 65
law0029. GeneralizedArcsine, 12, 49	moments20 (moments), 65
law0030 (law0030.FoldedNormal), 50	moments21 (moments), 65
law0030.FoldedNormal, 12, 50	moments22 (moments), 65
law0031 (law0031.MixtureNormal), 51	moments23 (moments), 65
law0031.MixtureNormal, <i>12</i> , 51	moments24 (moments), 65
law0032 (law0032.TruncatedNormal), 52	moments25 (moments), 65
law0032.TruncatedNormal, 12, 52	moments26 (moments), 65
law0033 (law0033.Nout), 53	moments27 (moments), 65
law0033.Nout, <i>12</i> , 53	moments28 (moments), 65
law0034(law0034.GeneralizedExpPower),	moments29 (moments), 65
54	moments3 (moments), 65
law0034.GeneralizedExpPower, 12,54	moments30 (moments), 65
law0035 (law0035.Exponential), 55	moments31 (moments), 65
law0035.Exponential, <i>12</i> , <i>55</i>	moments32 (moments), 65
law0036 (law0036.AsymmetricLaplace), 56	moments33 (moments), 65
law0036.AsymmetricLaplace, 12,56	moments34 (moments), 65
law0037 (law0037.NormalInvGaussian), 56	moments35 (moments), 65
law0037.NormalInvGaussian, 12, 56	moments36 (moments), 65
law0038	moments37 (moments), 65
(law0038.AsymmetricPowerDistribution)	
57	moments39 (moments), 65
law0038.AsymmetricPowerDistribution,	moments4 (moments), 65
12, 57	moments40 (moments), 65
law0039	moments41 (moments), 65
(law0039 modifiedAsymmetricPowerDistr	· /

moments5 (moments), 65	stat0010(stat0010.Hosking1),87
moments6 (moments), 65	stat0010.Hosking1, <i>66</i> ,87
moments7 (moments), 65	stat0011(stat0011.Hosking2),88
moments8 (moments), 65	stat0011.Hosking2, <i>66</i> ,88
moments9 (moments), 65	stat0012(stat0012.Hosking3),88
	stat0012.Hosking3,66,88
Normality.tests, 19, 21, 65, 82–105, 108,	stat0013 (stat0013.Hosking4), 89
136, 138–140, 148	stat0013.Hosking4, 66, 89
	stat0014 (stat0014.BontempsMeddahi1), 90
plot.discrepancy, 6, 17, 67, 69, 70	stat0014.BontempsMeddahi1, 66, 90
plot.pvalue, 6, 17, 68, 68, 70	stat0015 (stat0015.BontempsMeddahi2), 90
plot.sizepower, 6, 17, 68, 69, 69	stat0015.BontempsMeddahi2, 66, 90
powcomp.easy, 14, 70, 74, 81-144	stat0016 (stat0016.BrysHubertStruyf), 91
powcomp.fast, 14, 72, 72, 76, 78, 81-144	stat0016.BrysHubertStruyf, 66, 91, 92
<pre>powcomp.fastR (powcomp.fast), 72</pre>	stat0017 (stat0017.BonettSeier), 91
power.gui, 75	
print.critvalues, 62, 76, 78	stat0017.BonettSeier, 66, 91, 92
print.critvalues1 (print.critvalues), 76	stat0018
print.index (getindex), 14	(stat0018.BrysHubertStruyf-BonettSeier)
print.power, 77, 78	92
print.power1 (print.power), 78	stat0018.BrysHubertStruyf-BonettSeier,
pvalueMC, 79	92
F	stat0019 (stat0019.CabanaCabana1), 93
rbeta, 27	stat0019.CabanaCabana1, 66, 93
rcauchy, 24	stat0020 (stat0020.CabanaCabana2), 93
rgamma, 26	stat0020.CabanaCabana2, 66, 93
rlaw (gensample), 12	stat0021(stat0021.ShapiroWilk),94
rlogis, 25	stat0021.ShapiroWilk,66,94,96
runif, 28	stat0022(stat0022.ShapiroFrancia),95
1 41121, 20	stat0022.ShapiroFrancia, $66$ , $95$
stat.cstr, <i>14–16</i> , <i>22</i> , 80	stat0023(stat0023.ShapiroWilk-RG),95
stat0001 (stat0001.Lilliefors), 81	stat0023.ShapiroWilk-RG,95
stat0001.Lilliefors, 66, 81	stat0024 (stat0024.DAgostino), 96
stat0002 (stat0002. AndersonDarling), 82	stat0024.DAgostino, 66, 96
stat0002. AndersonDarling, 66, 82	stat0025 (stat0025.Filliben), 97
stat0003 (stat0003. ZhangWu1), 82	stat0025.Filliben, 66, 97
stat0003. ZhangWu1, 66, 82	stat0026(stat0026.ChenShapiro),97
stat0004 (stat0004. ZhangWu2), 83	stat0026.ChenShapiro, 66, 97
stat0004. ZhangWu2, 66, 83	stat0027 (stat0027.ZhangQ), 98
stat0005 (stat0005.GlenLeemisBarr), 84	stat0027.ZhangQ, 66, 98
stat0005.GlenLeemisBarr, 66, 84	stat0028 (stat0028. ZhangQQstar), 99
stat0006.Gtat0006.DAgostinoPearson), 84	stat0028. ZhangQQstar, 66, 99
stat0006.DAgostinoPearson, 66, 84	stat0029
stat0007 (stat0007. JarqueBera), 85	(stat0029.BarrioCuestaMatranRodriguez),
stat0007. JarqueBera, 66, 85	99
stat0008 (stat0008.DoornikHansen), 86	stat0029.BarrioCuestaMatranRodriguez,
stat0008.DoornikHansen, 66, 86	66, 99
stat0009 (stat0009.GelGastwirth), 86	stat0030 (stat0030.Coin), 100
stat0009.GelGastwirth, 66,86	stat0030.Coin, 66, 100

```
stat0031 (stat0031.EppsPulley), 101
                                                stat0049 (stat0049.Meintanis2MO), 113
stat0031.EppsPulley, 66, 101
                                                stat0049.Meintanis2MO, 20, 113
stat0032(stat0032.MartinezIglewicz),
                                                stat0050 (stat0050.Meintanis2ML), 113
                                                stat0050.Meintanis2ML, 20, 113
stat0032.MartinezIglewicz, 66, 101
                                                stat0051 (stat0051.ChoiKim1), 114
stat0033 (stat0033.GelMiaoGastwirth),
                                                stat0051.ChoiKim1, 20, 114
        102
                                                stat0052 (stat0052.ChoiKim2), 115
stat0033.GelMiaoGastwirth, 66, 102
                                                stat0052.ChoiKim2, 20, 115
stat0034 (stat0034. ZhangQstar), 103
                                                stat0053 (stat0053.ChoiKim3), 116
stat0034. ZhangQstar, 66, 103
                                                stat0053.ChoiKim3, 20, 116
stat0035
                                                stat0054
        (stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn),(stat0054.DesgagneMicheauxLeblanc-Gn),
                                                        116
stat0035.DesgagneLafayeDeMicheauxLeblanc-Rn, stat0054.DesgagneMicheauxLeblanc-Gn,
        103
                                                        116
stat0036
                                                stat0055 (stat0055.RavnerBest1), 117
        (stat0036.DesgagneLafayeDeMicheaux-XABDat0055.RaynerBest1, 20, 117
                                                stat0056 (stat0056.RaynerBest2), 118
stat0036.DesgagneLafayeDeMicheaux-XAPD,
                                                stat0056.RaynerBest2, 20, 118
        104
                                                stat0057 (stat0057.LangholzKronmal), 118
stat0037
                                                stat0057.LangholzKronmal, 20, 118
        (stat0037.DesgagneLafayeDeMicheaux-ZEBDat0058(stat0058.Kundu), 119
                                                stat0058.Kundu, 20, 119
stat0037.DesgagneLafayeDeMicheaux-ZEPD,
                                                stat0059 (stat0059.Gulati), 120
                                                stat0059.Gulati, 20, 120
stat0038 (stat0038.Glen), 105
                                                stat0060 (stat0060.Gel), 120
stat0038.Glen, 20, 105
                                                stat0060.Gel, 20, 120
stat0039 (stat0039. Rayner1), 106
                                                stat0061
stat0039.Rayner1, 20, 106
                                                        (stat0061.DesgagneMicheauxLeblanc-Lap1),
stat0040 (stat0040.Rayner2), 107
                                                        121
stat0040.Rayner2, 20, 107
                                                stat0061.DesgagneMicheauxLeblanc-Lap1,
stat0041 (stat0041.Spiegelhalter), 107
                                                        121
stat0041. Spiegelhalter, 67, 107
                                                stat0062
stat0042 (stat0042.AndersonDarling), 108
                                                        (stat0062.DesgagneMicheauxLeblanc-Lap2),
stat0042. AndersonDarling, 20, 108
stat0043 (stat0043.CramervonMises), 109
                                                stat0062.DesgagneMicheauxLeblanc-Lap2,
stat0043.CramervonMises, 20, 109
                                                         122
stat0044 (stat0044.Watson), 109
                                                stat0063 (stat0063.Kolmogorov), 122
stat0044. Watson, 20, 109
                                                stat0063.Kolmogorov, 122, 147
                                                stat0064 (stat0064.CramervonMises), 123
stat0045 (stat0045.KolmogorovSmirnov),
                                                stat0064.CramervonMises, 123, 147
stat0045.KolmogorovSmirnov, 20, 110
                                                stat0065 (stat0065. AndersonDarling), 124
stat0046 (stat0046.Kuiper), 111
                                                stat0065.AndersonDarling, 124, 147
                                                stat0066 (stat0066.Durbin), 124
stat0046.Kuiper, 20, 111
stat0047 (stat0047.Meintanis1MO), 111
                                                stat0066.Durbin, 124, 147
stat0047.Meintanis1MO, 20, 111
                                                stat0067 (stat0067.Kuiper), 125
stat0048 (stat0048.Meintanis1ML), 112
                                                stat0067.Kuiper, 125, 147
stat0048.Meintanis1ML, 20, 112
                                                stat0068 (stat0068. HegazyGreen1), 126
```

atat0000 HamaniCarant 106 147	-t-t0001 C11 20 140
stat0068. HegazyGreen1, 126, 147	stat0091.Gonzales1, 20, 140
stat0069 (stat0069.HegazyGreen2), 126	stat0092 (stat0092.Gonzales2), 141
stat0069. HegazyGreen2, 126, 147	stat0092.Gonzales2, 20, 141
stat0070 (stat0070.Greenwood), 127	stat0093 (stat0093.Hogg1), 141
stat0070. Greenwood, 127, 147	stat0093.Hogg1, 20, 141
stat0071 (stat0071.QuesenberryMiller),	stat0094 (stat0094.Hogg2), 142
128	stat0094.Hogg2, 20, 142
stat0071.QuesenberryMiller, 128, 147	stat0095 (stat0095.Hogg3), 143
stat0072 (stat0072.ReadCressie), 128	stat0095.Hogg3, 20, 143
stat0072.ReadCressie, 128, 147	stat0096 (stat0096.Hogg4), 143
stat0073 (stat0073.Moran), 129	stat0096.Hogg4, 20, 143
stat0073.Moran, 129, <i>147</i>	stat0097 (stat0097.Rizzo), 144
stat0074 (stat0074.Cressie1), 130	stat0097.Rizzo, 20, 144
stat0074.Cressie1, 130, <i>147</i>	stat1 (testsPureR), 146
stat0075 (stat0075.Cressie2), 130	stat10 (testsPureR), 146
stat0075.Cressie2, 130, <i>147</i>	stat100 (testsPureR), 146
stat0076 (stat0076.Vasicek), 131	stat101 (testsPureR), 146
stat0076.Vasicek, 131, <i>147</i>	stat102 (testsPureR), 146
stat0077 (stat0077.Swartz), 132	stat103 (testsPureR), 146
stat0077.Swartz, 132, <i>147</i>	stat104 (testsPureR), 146
stat0078 (stat0078.Morales), 132	stat105 (testsPureR), 146
stat0078.Morales, 132, 147	stat106 (testsPureR), 146
stat0079 (stat0079.Pardo), 133	stat107 (testsPureR), 146
stat0079.Pardo, 133, 147	stat108 (testsPureR), 146
stat0080 (stat0080.Marhuenda), 134	stat11 (testsPureR), 146
stat0080.Marhuenda, 134, <i>147</i>	stat12 (testsPureR), 146
stat0081 (stat0081.Zhang1), 135	stat13 (testsPureR), 146
stat0081.Zhang1, 135, 147	stat14 (testsPureR), 146
stat0082 (stat0082.Zhang2), 135	stat15 (testsPureR), 146
stat0082.Zhang2, 135, 147	stat16 (testsPureR), 146
stat0083 (stat0083.ttests), 136	stat17 (testsPureR), 146
stat0083.ttests, 136	stat18 (testsPureR), 146
stat0085	stat19 (testsPureR), 146
(stat0085.DesgagneMicheauxLeblanc-La	
137	stat20 (testsPureR), 146
stat0085.DesgagneMicheauxLeblanc-Lap3,	stat21 (testsPureR), 146
137	stat22 (testsPureR), 146
stat0086 (stat0086.Lafaye), 137	stat23 (testsPureR), 146
stat0086.Lafaye, 137	stat24 (testsPureR), 146
stat0087 (stat0087.Lafaye2), 138	stat25 (testsPureR), 146
stat0087.Lafaye2, 138	stat26 (testsPureR), 146
stat0088 (stat0088.Lafaye3), 138	stat27 (testsPureR), 146
stat0088.Lafaye3, 138	stat28 (testsPureR), 146
stat0089 (stat0089.Lafaye4), 139	stat29 (testsPureR), 146
stat0089.Lafaye4, 139	stat3 (testsPureR), 146
stat0090 (stat0090.Lafaye5), 140	stat30 (testsPureR), 146
stat0090 (stat0090.Larayes), 140	stat30 (testsPureR), 146
	,
stat0091 (stat0091.Gonzales1), 140	stat32 (testsPureR), 146

stat33 (testsPureR), 146	stat77 (testsPureR), 146
stat34 (testsPureR), 146	stat78 (testsPureR), 146
stat35 (testsPureR), 146	stat79 (testsPureR), 146
stat36 (testsPureR), 146	stat8 (testsPureR), 146
stat37 (testsPureR), 146	stat80 (testsPureR), 146
stat38 (testsPureR), 146	stat81 (testsPureR), 146
stat39 (testsPureR), 146	stat82 (testsPureR), 146
stat4 (testsPureR), 146	stat83 (testsPureR), 146
stat40 (testsPureR), 146	stat84 (testsPureR), 146
stat41 (testsPureR), 146	stat85 (testsPureR), 146
stat42 (testsPureR), 146	stat86 (testsPureR), 146
stat43 (testsPureR), 146	stat87 (testsPureR), 146
stat44 (testsPureR), 146	stat88 (testsPureR), 146
stat45 (testsPureR), 146	stat89 (testsPureR), 146
stat46 (testsPureR), 146	stat9 (testsPureR), 146
stat47 (testsPureR), 146	stat90 (testsPureR), 146
	stat91 (testsPureR), 146
stat48 (testsPureR), 146	stat92 (testsPureR), 146
stat49 (testsPureR), 146	stat93 (testsPureR), 146
stat5 (testsPureR), 146	stat94 (testsPureR), 146
stat50 (testsPureR), 146	stat95 (testsPureR), 146
stat51 (testsPureR), 146	stat96 (testsPureR), 146
stat52 (testsPureR), 146	stat97 (testsPureR), 146
stat53 (testsPureR), 146	stat98 (testsPureR), 146
stat54 (testsPureR), 146	stat99 (tests) uren, 146
stat55 (testsPureR), 146	statcompute, 14, 76, 80–144, 145
stat56 (testsPureR), 146	3 ta teompates, 17, 70, 00 177, 113
stat57 (testsPureR), 146	testsPureR, 146
stat58 (testsPureR), 146	,
stat59 (testsPureR), 146	Uniformity.tests, 19, 21, 67, 123-136, 147
stat6 (testsPureR), 146	
stat60 (testsPureR), 146	
stat61 (testsPureR), 146	
stat62 (testsPureR), 146	
stat63 (testsPureR), 146	
stat64 (testsPureR), 146	
stat65 (testsPureR), 146	
stat66 (testsPureR), 146	
stat67 (testsPureR), 146	
stat68 (testsPureR), 146	
stat69 (testsPureR), 146	
stat7 (testsPureR), 146	
stat70 (testsPureR), 146	
stat71 (testsPureR), 146	
stat72 (testsPureR), 146	
stat73 (testsPureR), 146	
stat74 (testsPureR), 146	
stat75 (testsPureR), 146	
stat76 (testsPureR), 146	