Package 'energyGOF'

December 1, 2025

Title Goodness-of-Fit Tests for Univariate Data via Energy

Version 0.1

Description Conduct one- and two-sample goodness-of-fit tests for univariate data. In the one-sample case, normal, uniform, exponential, Bernoulli, binomial, geometric, beta, Poisson, log-normal, Laplace, asymmetric Laplace, inverse Gaussian, half-normal, chi-squared, gamma, F, Weibull, Cauchy, and Pareto distributions are supported. egof.test() can also test goodness-of-fit to any distribution with a continuous distribution function. A subset of the available distributions can be tested for the composite goodness-of-fit hypothesis, that is, one can test for distribution fit with unknown parameters. P-values are calculated via parametric bootstrap.

License GPL (>= 3)

Encoding UTF-8

Imports energy, gsl, boot, fitdistrplus, statmod

URL https://github.com/jthaman/energyGOF

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author John Haman [aut, cre]

Maintainer John Haman <mail@johnhaman.org>

Repository CRAN

Date/Publication 2025-12-01 14:40:15 UTC

Contents

`energyGOF-package` *energyGOF: Goodness-of-Fit Tests via the Energy of Data*

Description

Conduct one- and two-sample goodness-of-fit tests for univariate data. In the one-sample case, normal, uniform, exponential, Bernoulli, binomial, geometric, beta, Poisson, lognormal, Laplace, asymmetric Laplace, inverse Gaussian, half-normal, chi-squared, gamma, F, Weibull, Cauchy, and Pareto distributions are supported. egof.test() can also test goodness-of-fit to any distribution with a continuous distribution function. A subset of the available distributions can be tested for the composite goodness-of-fit hypothesis, that is, one can test for distribution fit with unknown parameters. P-values are calculated via parametric bootstrap.

Getting Started

The main entry point is energyGOF.test(). The only documentation you need to read is `energyGOF.test()` and energyGOF-package.

Here is a simple example to get you going

```
x <- rnorm(10)

## Composite energy goodness-of-fit test (test for Normality with unknown
## parameters)

energyGOF.test(x, "normal", nsim = 1e5)

## Simple energy goodness-of-fit test (test for Normality with known
## parameters). egof.test is an alias for energyGOF.test.
```

```
egof.test(x, "normal", nsim = 1e5, mean = 0, sd = 1)


## Two-sample test
y <- rt(10, 1)
egof.test(x, y, nsim = 1e5)

## Test agaist any distribution function by transforming data to uniform
egof.test(y, pt, nsim = 1e5)
```

You may alternatively use the `energyGOFdist()` function, which is a different interface using S3 objects, but it provides the same result. There is a lot of documentation in this package for the various S3 constructors that are needed by `energyGOFdist()`, **BUT** if you just want to do some testing and use the standard interface, you can probably ignore all of that and just read the page for `energyGOF.test()`.

Distributions Supported

The following distributions are supported.

Distribution	Function	Parameters	Composite_Test
Asymmetric Laplace	alaplace_dist	location, scale, skew	TRUE
Asymmetric Laplace	asymmetric_laplace_dist	location, scale, skew	TRUE
Bernoulli	bernoulli_dist	prob	FALSE
Beta	beta_dist	shape1, shape2	TRUE
Binomial	binomial_dist	size, prob	FALSE
Cauchy	cauchy_dist	location, scale, pow	TRUE
Chi-Squared	chisq_dist	df	FALSE
Exponential	exp_dist	rate	TRUE
Exponential	exponential_dist	rate	TRUE
F	f_dist	df1, df2	FALSE
Gamma	gamma_dist	shape, rate	FALSE
Geometric	geometric_dist	prob	FALSE
Half-Normal	halfnormal_dist	scale	TRUE
Inverse Gaussian	inverse_gaussian_dist	mean, shape	TRUE
Inverse Gaussian	invgauss_dist	mean, shape	TRUE
Laplace	laplace_dist	location, scale	TRUE
Lognormal	lognormal_dist	meanlog, sdlog	TRUE
Normal	normal_dist	mean, sd	TRUE
Pareto (Type I)	pareto_dist	scale, shape, pow	TRUE
Poisson	poisson_dist	lambda	TRUE
Uniform	uniform_dist	min, max	FALSE
Weibull	weibull_dist	shape, scale	TRUE

Simple and Composite Testing

There are two types of goodness-of-fit tests covered by the energyGOF package, *simple* and *composite*. It's important to know the difference because they yield different results. Simple GOF tests

test the data `x` against a specific distribution with *known parameters* that you must pass to energyGOF.test in the ellipsis argument (...). You should use a simple GOF test if you wish to test questions like "my data are Normal with mean 1 and sd 2". energyGOF() can also conduct *some* composite GOF tests. A composite test is performed if no parameters are passed in the ellipsis argument (...). You should conduct a composite test if your research question is "my data are Normal, but I don't know what the parameters are." Obviously, this composite question is much more common in practice.

All the composite tests in energyGOF assume that *none* of the parameters are known. So while there is a statistical test of Normality with known mean and unknown sd, this is not implemented in the energyGOF package. So, either pass all the distribution parameters or none of them. (In the special case of the Normal distribution, you can use the energy::energy package to test the GOF hypothesis with any combination of known and unknown parameters.)

For each test, energyGOF.test() calculates the test statistic and a *p*-value. In all cases the *p*-value is calculated via parametric bootstrap. For large `nsim`, the *p*-values should be reasonably honest in small-ish samples. You may need to perform a sensitivity study to find a reasonable `nsim` for your particular testing problem.

Power Analyses

Please see the repository https://github.com/jthaman/energyGOF-power for examples of how to conduct power analyses with energyGOF, and for preliminary performance data agaist alternative methods.

About Energy

Székely, G. J., & Rizzo, M. L. (2023) provide the motivation:

"Data energy is a real number (typically a non-negative number) that depends on the distances between data. This concept is based on the notion of Newton's gravitational potential energy, which is also a function of the distance between bodies. The idea of data energy or energy statistics is to consider statistical observations (data) as heavenly bodies governed by the potential energy of data, which is zero if and only if an underlying statistical hypothesis is true."

The notation X' indicates that X' is an independent and identically distributed copy of X.

If X and Y are independent and $E(|X|^s + |Y|^s)$ is finite, then for $0 < s < 2$,

$$2E|X - Y|^s - E|X - X'|^s - E|Y - Y'|^s \geq 0.$$

Equality is attained if and only if X and Y are identically distributed. The left side of the equation is the energy between X and Y. Energy can be generalized to multivariate data and even more exotic data types, but in this R package, we only treat univariate data.

The concept of data energy between two random variables can be adapted to the one-sample goodness-of-fit problem. The one-sample s-energy is

$$E^* = \frac{2}{n}\sum_i E|x_i - Y|^s - E|Y - Y'|^s - \frac{1}{n^2}\sum_i\sum_j |x_i - x_j|^s,$$

when $0 < s < 2$ and $E|X|^s, E|Y|^s < \infty$.

In most tests in the energyGOF package $s = 1$. In some cases (Pareto and Cauchy), $E|Y|$ is not finite, so we need to use an $s < 1$. This is done by passing pow into ... (but in all tests a default pow is provided). These tests are called *generalized* energy goodness-of-fit tests in this package as well as in Székely, G. J., & Rizzo, M. L. (2023).

To connect energy back to GOF testing, in the one-sample goodness-of-fit regime, we test if a sample $x_1, \ldots, x_n \sim X$ (where the distribution of X is hidden) follows the same distribution as Y, which is specified. If X and Y have the same distribution, then the distribution of $Q = nE^*$ is a quadratic form of centered Gaussian random variables with expected value $E|Y - Y'|^s$. If X and Y differ, then $Q \to \infty$ with n. So, Q provides a consistent goodness-of-fit test, even in some situations where $E|Y|$ is not finite. Asymptotic theory of V-statistics can be applied to prove that tests based on Q are statistically consistent goodness-of-fit tests.

Author(s)

John T. Haman

References

Székely, G. J., & Rizzo, M. L. (2023). The energy of data and distance correlation. Chapman and Hall/CRC.

Székely, G. J., & Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. Journal of statistical planning and inference, 143(8), 1249-1272.

Li, Y. (2015). Goodness-of-fit tests for Dirichlet distributions with applications. Bowling Green State University.

Rizzo, M. L. (2002). A new rotation invariant goodness-of-fit test (PhD thesis). Bowling Green State University

Haman, J. T. (2018). The energy goodness-of-fit test and EM type estimator for asymmetric Laplace distributions (Doctoral dissertation, Bowling Green State University).

Ofosuhene, P. (2020). The energy goodness-of-fit test for the inverse Gaussian distribution (Doctoral dissertation, Bowling Green State University).

Rizzo, M. L. (2009). New goodness-of-fit tests for Pareto distributions. ASTIN Bulletin: The Journal of the IAA, 39(2), 691-715.

Yang, G. (2012). The Energy Goodness-of-fit Test for Univariate Stable Distributions (Doctoral dissertation, Bowling Green State University).

See Also

Useful links:

- https://github.com/jthaman/energyGOF

`asymmetric_laplace_dist` *Create an asymmetric Laplace distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against an asymmetric Laplace distribution. If all three parameters are NULL, perform a composite test. This is exactly the distribution corresponding to the PDF

$$f(x|\theta,\sigma,\kappa) = \frac{\sqrt{2}\kappa}{\sigma(1+\kappa^2)} \begin{cases} \exp\left(-\frac{\sqrt{2}\kappa|x-\theta|}{\sigma}\right), & x \geq \theta, \\ \exp\left(-\frac{\sqrt{2}|x-\theta|}{\kappa\sigma}\right), & x < \theta. \end{cases},$$

where θ = `location`, σ = `scale`, and κ = `skew`.

Usage

```
asymmetric_laplace_dist(location = NULL, scale = NULL, skew = NULL)

alaplace_dist(location = NULL, scale = NULL, skew = NULL)
```

Arguments

`location`	NULL, or a location parameter
`scale`	NULL, or a positive scale parameter
`skew`	NULL, or a positive skewness parameter. Skew = 1 corresponds to a symmetric Laplace distribution (though note the difference between the PDF in this description and the one in `laplace_dist()`).

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)

- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform x. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- asymmetric_laplace_dist(0, 1, .5)
x <- d$sampler(10, d$par)

egofd(x, d, 0)
```

`bernoulli_dist` *Create a Bernoulli distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Bernoulli distribution. Only simple tests are implemented.

Usage

```
bernoulli_dist(prob = 0.5)
```

Arguments

`prob`	Same as `rbinom()`, but must be length 1.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.

- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- bernoulli_dist(.5)
egofd(rbinom(10, 1, .5), d, 0)
```

`beta_dist` *Create a beta distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a beta distribution. If shape1 and shape2 are NULL, a composite test is performed, otherwise a simple test is performed.

Usage

```
beta_dist(shape1 = NULL, shape2 = NULL)
```

Arguments

`shape1`	Same as `rbeta()`, but must be length 1.
`shape2`	Same as `rbeta()`, but must be length 1.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- beta_dist(5, 5)
egofd(rbeta(10, 5, 5), d, 0)
```

`binomial_dist` *Create a Binomial distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Binomial distribution. Only a simple GOF test is supported.

Usage

```
binomial_dist(size = 1, prob = 0.5)
```

Arguments

size	Same as stats::rbinom(), but must be length 1.
prob	Same as stats::rbinom(), but must be length 1.

Value

S3 data object containing the following fields.

- name: String
- composite_p: Composite predicate. TRUE if test is composite.
- par: Distribution parameters, list of the formals.
- sampler_par: Distribution parameters used for the calculation of energy statistic. These may be different than par.
- par_domain: Function used to ensure par and sampler_par are valid for this distribution
- support: Function to check that data x can be tested against y
- sampler: Function used for rng by boot::boot()
- EYY: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- EXYhat: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in egof.test or egofd).
- xform: Function that may be used to transform x. Only available in certain distribution objects.
- statistic: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- notes: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- binomial_dist(1, 0.5)
egofd(rbinom(10, 1, .5), d, 0)
```

`cauchy_dist` *Create a Cauchy distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the generalized energy goodness-of-fit test against a Cauchy distribution. If `location` and `scale` are both NULL, perform a composite test.

Usage

```
cauchy_dist(location = NULL, scale = NULL, pow = 0.5)
```

Arguments

`location`	NULL, or same as in `stats::rcauchy()`
`scale`	NULL, or same as in `stats::rcauchy()`
`pow`	Optionally set the exponent of the energy test. 0 < pow < 1 is required for the Cauchy distribution. Default is 0.5.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- cauchy_dist(4, 4)
x <- rcauchy(10, 4, 4)
egofd(x, d, 0)
```

---

chisq_dist *Create a Chi-squared distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Chi-squared distribution. Only simple tests are supported.

Usage

```
chisq_dist(df = 2)
```

Arguments

df	Same as in stats::rchisq().

Value

S3 data object containing the following fields.

- name: String
- composite_p: Composite predicate. TRUE if test is composite.
- par: Distribution parameters, list of the formals.
- sampler_par: Distribution parameters used for the calculation of energy statistic. These may be different than par.
- par_domain: Function used to ensure par and sampler_par are valid for this distribution
- support: Function to check that data x can be tested against y
- sampler: Function used for rng by boot::boot()
- EYY: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- EXYhat: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in egof.test or egofd).
- xform: Function that may be used to transform x. Only available in certain distribution objects.
- statistic: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.

- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- chisq_dist(4)
egofd(rchisq(10, 4), d, 0)
```

`energyGOF.test` *Goodness-of-fit tests for univariate data via energy*

Description

Perform a goodness-of-fit test of univariate data `x` against a target `y`. `y` may be one of the following:

- A string naming a distribution. For example, "normal". Both simple (known parameter) and composite (unknown parameter) tests are supported, but not all distributions allow for a composite test. See energyGOF-package for the table of supported distributions.
  - Result: A parametric goodness-of-fit test is performed.
  - Allowable values: uniform, exponential, bernoulli, binomial, geometric, normal, gaussian, beta, poisson, lognormal, lnorm, laplace, doubleexponential, asymmetriclaplace, alaplace, inversegaussian, invgaussian, halfnormal, chisq, chisquared, f, gamma, weibull, cauchy, pareto.
- A numeric vector of data.
  - Result: A two-sample, non-parametric goodness-of-fit test is performed to test if x and y are equal in distribution.
- A continuous cumulative distribution function. For example, `pt`. Only simple tests are supported.
  - Result: $y(x)$ is tested for uniformity.

P-values are determined via parametric bootstrap. For distributions where $E|Y|$ is not finite (Cauchy, Pareto), a *generalized* energy goodness-of-fit test is performed, and an additional tuning parameter `pow` is required.

Usage

```
energyGOF.test(x, y, nsim, ...)

egof.test(x, y, nsim, ...)
```

Arguments

x	A numeric vector.
y	A string, distribution function, or numeric vector. The distribution to test x against.
nsim	A non-negative integer. The number of parametric bootstrap replicates taken to calculate the *p*-value. If 0, no simulation.
...	If y is a string or distribution function, parameters of the distribution y. Required for a simple test. For distributions in the stats library, parameter argument names are identical. If y is a string, to test the *composite* goodness-of-fit hypothesis that x is distributed according to the *family of distributions* y, don't pass parameters in .... For *generalized* energy tests, you can also optionally pass the generalized energy exponent pow here. Composite testing is not supported if y is a function. (As you can see, there is a lot going on in ... and if you don't like that, you may want to check out energyGOFdist() for a structured interface.)

Value

If y is a string or function, return an object of class 'htest' representing the result of the energy goodness-of-fit hypothesis test. The htest object has the elements:

- method: Simple or Composite
- data.name
- distribution: The distribution object created to test
- parameter: List of parameters if the test is simple
- nsim: Number of bootstrap replicates
- composite_p: TRUE/FALSE composite predicate
- statistic: The value of the energy statistic ($Q = nE^*$)
- p.value
- sim_reps: bootstrap replicates of energy statistic
- estimate: Any parameter estimates, if the test is composite

If y is numeric, return the same htest object as energy::eqdist.etest().

Author(s)

John T. Haman

See Also

- energyGOF-package for specifics on the distributions available to test.
- energyGOFdist() for the alternate S3 interface for parametric testing.
- Distributions for a list of distributions available in most R installations.
- energy::eqdist.etest() for information on the two-sample test.

- energy::normal.test() for the energy goodness-of-fit test with unknown parameters. The tests for (multivariate) Normal in the energy package are implemented with compiled code, and are faster than the one available in the energyGOF package.
- energy::poisson.mtest() for a different Poisson goodness-of-fit test based on mean distances.

Examples

```
x <- rnorm(10)
y <- rt(10, 4)

## Composite energy goodness-of-fit test (test for Normality with unknown
## parameters)

energyGOF.test(x, "normal", nsim = 10)

## Simple energy goodness-of-fit test (test for Normality with known
## parameters). egof.test is an alias for energyGOF.test.

egof.test(x, "normal", nsim = 10, mean = 0, sd = 1)

## Alternatively, use the energyGOFdist generic directly so that you do not need
## to pass parameter names into `...`

energyGOFdist(x, normal_dist(0, 1), nsim = 10)

## Conduct a two-sample test

egof.test(x, y, 0)

## Conduct a test against any continuous distribution function

egof.test(x, pcauchy, 0)

## Simple energy goodness-of-fit test for Weibull distribution

y <- rweibull(10, 1, 1)
energyGOF.test(y, "weibull", shape = 1, scale = 3, nsim = 10)

## Alternatively, use the energyGOFdist generic directly, which is slightly less
## verbose. egofd is an alias for energyGOFdist.

egofd(y, weibull_dist(1, 3), nsim = 10)

## Conduct a generalized GOF test. `pow` is the exponent *s* in the generalized
## energy statistic. Pow is only necessary when testing Cauchy, and
## Pareto distributions. If you don't set a pow, there is a default for each
## of the distributions, but the default isn't necessarily better than any
## other number.

egofd(rcauchy(100),
   cauchy_dist(location = 0, scale = 1, pow = 0.5),
```

```
    nsim = 10)

## energyGOF does not support tests with a mix of known and unknown
## parameters, so this will result in an error.


  energyGOF.test(x, "normal", mean = 0, nsim = 10) # sd is missing
```

`energyGOFdist` *S3 Interface to Parametric Goodness-of-Fit Tests via Energy*

Description

This is an alternative interface that provides the same parametric tests as `energyGOF.test()`, but allows the user to directly pass a distribution object like `normal_dist()` (Distribution objects are specific to the implementation of this R package). The advantage is that you do not need to pass distribution parameters into a `...` argument as in energyGOF.test(). energyGOF.test() uses this function under the hood, but it's perfectly suitable for the user to use as well.

Usage

```
energyGOFdist(x, dist, nsim)

egofd(x, dist, nsim)
```

Arguments

`x`	A numeric vector.
`dist`	An object of class GOFDist. The distribution to test `x` against. GOFDist objects are created with the various "*_dist()" functions in this package. See, for example, `normal_dist()` for details on these class objects.
`nsim`	A non-negative integer. The number of parametric bootstrap replicates taken to calculate the *p*-value. If 0, no simulation.

Value

Return an object of class 'htest' representing the result of the energy goodness-of-fit hypothesis test. The htest object has the elements:

- `method`: Simple or Composite
- `data.name`
- `distribution`: The distribution object created to test
- `parameter`: List of parameters if the test is simple
- `nsim`: Number of bootstrap replicates

- composite_p: TRUE/FALSE composite predicate
- statistic: The value of the energy statistic ($Q = nE^*$)
- p.value
- sim_reps: bootstrap replicates of energy statistic
- estimate: Any parameter estimates, if the test is composite

Author(s)

John T. Haman

Examples

```
## Simple normal test
energyGOFdist(rnorm(10), normal_dist(0, 1), nsim = 10)

## Simple Poisson test
egofd(rpois(10,1), poisson_dist(1), nsim = 0) # No p-value

## Composite Normal test
egofd(rnorm(10), normal_dist(), nsim = 10)
```

---

exponential_dist *Create an Exponential distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against an exponential distribution. If rate is NULL, a composite test is performed.

Usage

```
exponential_dist(rate = NULL)
```

Arguments

rate	NULL, or a positive rate parameter as in rexp(), but must be length 1.

Value

S3 data object containing the following fields.

- name: String
- composite_p: Composite predicate. TRUE if test is composite.
- par: Distribution parameters, list of the formals.

- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- exponential_dist(1)
egofd(rexp(10, 1), d, 0)
```

`f_dist` *Create an F distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a F distribution. Only simple tests are supported.

Usage

```
f_dist(df1 = 3, df2 = 3)
```

Arguments

`df1`	Positive.
`df2`	Must be greater than 2.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- f_dist(3, 3)
egofd(rf(10, 3, 3), d, 0)
```

---

gamma_dist *Create a gamma distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Gamma distribution. Only simple tests are supported.

Usage

```
gamma_dist(shape = 1, rate = 1)
```

Arguments

shape	Same shape parameter in stats::rgamma() (must be length 1)
rate	Same rate parameter in stats::rgamma() (must be length 1)

Value

S3 data object containing the following fields.

- name: String
- composite_p: Composite predicate. TRUE if test is composite.
- par: Distribution parameters, list of the formals.
- sampler_par: Distribution parameters used for the calculation of energy statistic. These may be different than par.
- par_domain: Function used to ensure par and sampler_par are valid for this distribution
- support: Function to check that data x can be tested against y
- sampler: Function used for rng by boot::boot()
- EYY: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- EXYhat: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in egof.test or egofd).
- xform: Function that may be used to transform x. Only available in certain distribution objects.
- statistic: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- notes: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- gamma_dist(4, 4)
egofd(rgamma(10, 4, 4), d, 0)
```

`geometric_dist` *Create a geometric distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a geometric distribution. Only a simple test is supported.

Usage

```
geometric_dist(prob = 0.5)
```

Arguments

`prob`	Same as `rgeom()`, but must be length 1.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- geometric_dist(.5)
egofd(rgeom(10, .5), d, 0)
```

`halfnormal_dist` *Create a half-normal distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a half-normal distribution. If scale is NULL, a composite test is performed.

This is exactly the distribution of $|X|$, where X $N(0, \theta = scale)$

Usage

```
halfnormal_dist(scale = NULL)
```

Arguments

`scale`	NULL, or a positive scale parameter, like sd in `rnorm()`. Must be length 1.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- halfnormal_dist(4)
egofd(abs(rnorm(10, 4)), d, 0)
```

---

inverse_gaussian_dist *Create an inverse Gaussian distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against an inverse Gaussian distribution. If mean and shape are both NULL, perform a composite test. This is exactly the distribution corresponding to the PDF

$$f(x|\mu,\lambda) = \left(\frac{\lambda}{2\pi x^3}\right)^{1/2} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right), \qquad x > 0,$$

where mean is μ and shape is λ.

Usage

```
inverse_gaussian_dist(mean = NULL, shape = NULL)

invgauss_dist(mean = NULL, shape = NULL)
```

Arguments

mean	NULL or a positive mean parameter
shape	NULL or a positive shape parameter

Details

This distribution requires an intense amount of numerical integration for the simple (known parameters) case, and the implementation seems to be fine for samples up to 1000. For the composite case, the data are transformed to a Chi-squared distribution (conditional on the parameter estimates), and the performance is much better, as there is no numerical integration in this case.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- inverse_gaussian_dist(4, 4)
x <- d$sampler(10, d$par)

egofd(x, d, 0)
```

`laplace_dist` *Create a Laplace distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Laplace distribution. If location and scale are both NULL, a composite test is performed.

This is exactly the distribution corresponding to the PDF

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

where `location` = μ and `scale` = b.

Usage

```
laplace_dist(location = NULL, scale = NULL)
```

Arguments

`location`	NULL, or the median of the distribution
`scale`	NULL or a positive scale parameter

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- laplace_dist(1, 1)

x <- d$sampler(10, d$par)

egofd(x, d, 0)
```

`lognormal_dist` *Create a lognormal distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a lognormal distribution. If `meanlog` and `sdlog` are both `NULL`, a composite test is performed.

Usage

```
lognormal_dist(meanlog = NULL, sdlog = NULL)
```

Arguments

`meanlog`	NULL or as in `rlnorm()`, must be length 1.
`sdlog`	NULL or as in `rlnorm()`, must be length 1. d <- lognormal_dist(0, 1) x <- d$sampler(10, d$par) egofd(x, d, 0)

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)

- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform x. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

`normal_dist` *Create a Normal distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a normal distribution. If `mean` and `sd` are both NULL, perform a composite test.

Usage

```
normal_dist(mean = NULL, sd = NULL)
```

Arguments

`mean`	NULL, or if specified, same as `rnorm()`, but must be length 1.
`sd`	NULL, or if specified, Same as `rnorm()`, but must be length 1

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`

- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform x. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- normal_dist(0, 1)

# Composite test
dc <- normal_dist()
egofd(rnorm(10), dc, 0)


### Expected distances:

d$EYY(d$par)

## should be about the same as mean(abs(rnorm(1e5) - rnorm(1e5)))

x <- 3

d$EXYhat(3, d$par)

## should be about the same as mean(abs(x - rnorm(1e5)))
```

`pareto_dist` *Create a Pareto (type I) distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Pareto distribution. If `scale` and `shape` are both NULL, perform a composite test.

Usage

```
pareto_dist(scale = NULL, shape = NULL, pow = NULL)
```

Arguments

scale	NULL or a positive scale parameter
shape	NULL or a positive shape parameter. If shape > 1, shape is used to transform x
pow	Optional exponent of the energy test. Pow must be less than shape. If shape > 1 and pow != 1, pow will be scaled down.

Details

If shape > 1, the energy test is more difficult, so data are transformed to data^shape ~ Pareto(scale^shape, 1).

Value

S3 data object containing the following fields.

- name: String
- composite_p: Composite predicate. TRUE if test is composite.
- par: Distribution parameters, list of the formals.
- sampler_par: Distribution parameters used for the calculation of energy statistic. These may be different than par.
- par_domain: Function used to ensure par and sampler_par are valid for this distribution
- support: Function to check that data x can be tested against y
- sampler: Function used for rng by boot::boot()
- EYY: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- EXYhat: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to y and x is the data under test (which is passed in egof.test or egofd).
- xform: Function that may be used to transform x. Only available in certain distribution objects.
- statistic: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- notes: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- pareto_dist(1, .5)
x <- d$sampler(10, d$par)
egofd(x, d, 0)
```

`poisson_dist` *Create a Poisson distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Poisson distribution. If lambda is NULL, a composite test is performed.

Usage

```
poisson_dist(lambda = NULL)
```

Arguments

`lambda`	NULL, or if specified, same as the lambda in `rpois()`, but must be length 1.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- poisson_dist(1)

egofd(rpois(10, 1), d, 0)
```

`uniform_dist` *Create a Uniform distribution object for energy testing*

Description

Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a uniform distribution. Only simple tests are implemented.

Usage

```
uniform_dist(min = 0, max = 1)
```

Arguments

`min`	Same as in `runif()`, but must be length 1
`max`	Same as in `runif()`, but must be length 1

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.
- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.

- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- uniform_dist(0, 1)

egofd(runif(10), d, 0)
```

`weibull_dist` *Create a Weibull distribution object for energy testing*

Description

Create a Weibull distribution object for energy testing

Usage

```
weibull_dist(shape = NULL, scale = NULL)
```

Arguments

`shape`	NULL, or if specified, same as the shape parameter in `stats::rweibull()`
`scale`	NULL, or if specified, same as the scale parameter in `stats::rweibull()` #' @description Create an S3 object that sets all the required data needed by energyGOFdist to execute the energy goodness-of-fit test against a Weibull distribution. If `shape` and `scale` are both NULL, perform a composite test.

Value

S3 data object containing the following fields.

- `name`: String
- `composite_p`: Composite predicate. TRUE if test is composite.
- `par`: Distribution parameters, list of the formals.
- `sampler_par`: Distribution parameters used for the calculation of energy statistic. These may be different than `par`.

- `par_domain`: Function used to ensure `par` and `sampler_par` are valid for this distribution
- `support`: Function to check that data `x` can be tested against `y`
- `sampler`: Function used for rng by `boot::boot()`
- `EYY`: Function to compute $E|Y - Y'|$ (or $E|Y - Y'|^{pow}$, for the generalized test.)
- `EXYhat`: Function to compute $\frac{1}{n}\sum_i E|x_i - Y|$ (or $\frac{1}{n}\sum_i E|x_i - Y|^{pow}$), where Y is distributed according to `y` and `x` is the data under test (which is passed in `egof.test` or `egofd`).
- `xform`: Function that may be used to transform `x`. Only available in certain distribution objects.
- `statistic`: Function that returns a list of maximum likelihood estimates. Only available in certain distribution objects.
- `notes`: Distribution specific messages. Only used in certain distribution objects.

Note: Some distributions do not have notes, xform, and statistic fields. This is because either a composite test is not implemented, or because a data transformation is not needed.

Author(s)

John T. Haman

Examples

```
d <- weibull_dist(3, 3)
egofd(rweibull(10, 3, 3), d, 0)
```

Index