

# Package ‘reliaR’

February 20, 2015

**Type** Package

**Title** Package for some probability distributions.

**Version** 0.01

**Date** 2010-10-28

**Author** Vijay Kumar <vkgkp@rediffmail.com> and Uwe Ligges  
<ligges@statistik.tu-dortmund.de>

**Maintainer** Vijay Kumar <vkgkp@rediffmail.com>

**Description** A collection of utilities for some reliability  
models/probability distributions.

**License** GPL-2

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2011-01-03 18:56:29

**NeedsCompilation** no

## R topics documented:

abic.burrX . . . . .	4
abic.chen . . . . .	5
abic.exp.ext . . . . .	6
abic.exp.power . . . . .	7
abic.expo.logistic . . . . .	8
abic.expo.weibull . . . . .	9
abic.flex.weibull . . . . .	10
abic.gen.exp . . . . .	11
abic.gompertz . . . . .	12
abic.gp.weibull . . . . .	13
abic.gumbel . . . . .	14
abic.inv.genexp . . . . .	15
abic.lfr . . . . .	16
abic.log.gamma . . . . .	17
abic.logis.exp . . . . .	18

abic.logis.rayleigh . . . . .	19
abic.loglog . . . . .	20
abic.moee . . . . .	21
abic.moew . . . . .	22
abic.weibull.ext . . . . .	23
bearings . . . . .	24
BurrX . . . . .	25
BurrXsurvival . . . . .	26
Chen . . . . .	28
Chensurvival . . . . .	30
conductors . . . . .	31
dataset2 . . . . .	32
EPsurvival . . . . .	33
ExpExt . . . . .	34
ExpExtsurvival . . . . .	35
ExpoLogistic . . . . .	37
ExpoLogisticsurvival . . . . .	38
ExpoWeibull . . . . .	39
ExpoWeibullsurvival . . . . .	41
ExpPower . . . . .	42
FlexWeibull . . . . .	43
FlexWeibullsurvival . . . . .	45
GenExp . . . . .	46
GenExpSurvival . . . . .	47
Gompertz . . . . .	49
GompertzSurvival . . . . .	50
GPWeibull . . . . .	51
GPWeibullsurvival . . . . .	53
Gumbel . . . . .	54
GumbelSurvival . . . . .	55
InvGenExp . . . . .	57
InvGenExpSurvival . . . . .	58
ks.burrX . . . . .	59
ks.chen . . . . .	61
ks.exp.ext . . . . .	62
ks.exp.power . . . . .	63
ks.expo.logistic . . . . .	64
ks.expo.weibull . . . . .	65
ks.flex.weibull . . . . .	67
ks.gen.exp . . . . .	68
ks.gompertz . . . . .	69
ks.gp.weibull . . . . .	70
ks.gumbel . . . . .	71
ks.inv.genexp . . . . .	72
ks.lfr . . . . .	74
ks.log.gamma . . . . .	75
ks.logis.exp . . . . .	76
ks.logis.rayleigh . . . . .	77

ks.loglog . . . . .	78
ks.moee . . . . .	79
ks.moew . . . . .	81
ks.weibull.ext . . . . .	82
LFR . . . . .	83
LFRsurvival . . . . .	84
Loggamma . . . . .	86
Loggammastrvival . . . . .	87
LogisExp . . . . .	88
LogisExpstrvival . . . . .	90
LogisRayleigh . . . . .	91
LogisRayleighstrvival . . . . .	92
Loglog . . . . .	93
Loglogstrvival . . . . .	95
MOEE . . . . .	96
MOEEstrvival . . . . .	97
MOEW . . . . .	99
MOEWstrvival . . . . .	100
pp.burrX . . . . .	101
pp.chen . . . . .	102
pp.exp.ext . . . . .	103
pp.exp.power . . . . .	104
pp.expo.logistic . . . . .	105
pp.expo.weibull . . . . .	106
pp.flex.weibull . . . . .	107
pp.gen.exp . . . . .	108
pp.gompertz . . . . .	109
pp.gp.weibull . . . . .	110
pp.gumbel . . . . .	111
pp.inv.genexp . . . . .	112
pp.lfr . . . . .	113
pp.log.gamma . . . . .	114
pp.logis.exp . . . . .	115
pp.logis.rayleigh . . . . .	116
pp.loglog . . . . .	117
pp.moee . . . . .	118
pp.moew . . . . .	119
pp.weibull.ext . . . . .	120
qq.burrX . . . . .	121
qq.chen . . . . .	122
qq.exp.ext . . . . .	123
qq.exp.power . . . . .	124
qq.expo.logistic . . . . .	125
qq.expo.weibull . . . . .	126
qq.flex.weibull . . . . .	127
qq.gen.exp . . . . .	128
qq.gompertz . . . . .	129
qq.gp.weibull . . . . .	130

qq.gumbel . . . . .	131
qq.inv.genexp . . . . .	132
qq.lfr . . . . .	133
qq.log.gamma . . . . .	134
qq.logis.exp . . . . .	135
qq.logis.rayleigh . . . . .	136
qq.loglog . . . . .	137
qq.moee . . . . .	138
qq.moew . . . . .	139
qq.weibull.ext . . . . .	140
reactorpump . . . . .	141
repairtimes . . . . .	142
stress . . . . .	143
sys2 . . . . .	143
WeibullExt . . . . .	144
WeibullExtSurvival . . . . .	146

<b>Index</b>	<b>148</b>
--------------	------------

---

abic.burrX	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for BurrX distribution</i>
------------	---

---

## Description

The function `abic.burrX()` gives the loglikelihood, AIC and BIC values assuming an BurrX distribution with parameters `alpha` and `lambda`.

## Usage

```
abic.burrX(x, alpha.est, lambda.est)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>

## Value

The function `abic.burrX()` gives the loglikelihood, AIC and BIC values.

## References

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

## See Also

[pp.burrX](#) for PP plot and [qq.burrX](#) for QQ plot

## Examples

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

## Values of AIC, BIC and LogLik for the data(bearings)
abic.burrX(bearings, 1.1989515, 0.0130847)
```

---

abic.chen	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for a sample from Chen distribution</i>
-----------	--

---

## Description

The function `abic.chen()` gives the loglikelihood, AIC and BIC values assuming Chen distribution with parameters `beta` and `lambda`. The function is based on the invariance property of the MLE.

## Usage

```
abic.chen(x, beta.est, lambda.est)
```

## Arguments

<code>x</code>	vector of observations
<code>beta.est</code>	estimate of the parameter <code>beta</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>

**Value**

The function `abic.chen()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.chen](#) for PP plot and [qq.chen](#) for QQ plot

**Examples**

```
## Load data sets

data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

## Values of AIC, BIC and LogLik for the data(sys2)
abic.chen(sys2, 0.262282404, 0.007282371)
```

---

abic.exp.ext	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Exponential Extension(EE) distribution</i>
--------------	---

---

**Description**

The function `abic.exp.ext()` gives the loglikelihood, AIC and BIC values assuming an Exponential Extension(EE) distribution with parameters alpha and lambda.

**Usage**

```
abic.exp.ext(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.exp.ext()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, Biometrika, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, Annals of Statistics, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, Journal of the Royal Statistical Society Series B 64, 1-34.

**See Also**

[pp.exp.ext](#) for PP plot and [qq.exp.ext](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

## Values of AIC, BIC and LogLik for the data(sys2)
abic.exp.ext(sys2, 1.0126e+01, 1.5848e-04)
```

---

abic.exp.power	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for a sample from Exponential Power(EP) distribution</i>
----------------	---

---

**Description**

The function `abic.exp.power()` gives the loglikelihood, AIC and BIC values assuming Chen distribution with parameters alpha and lambda. The function is based on the invariance property of the MLE.

**Usage**

```
abic.exp.power(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.exp.power()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi, S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.exp.power](#) for PP plot and [qq.exp.power](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

## Values of AIC, BIC and LogLik for the data(sys2)

abic.exp.power(sys2, 0.905868898, 0.001531423)
```

---

<code>abic.expo.logistic</code>	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Exponentiated Logistic(EL) distribution</i>
---------------------------------	--

---

**Description**

The function `abic.expo.logistic()` gives the loglikelihood, AIC and BIC values assuming an Exponentiated Logistic(EL) distribution with parameters alpha and beta.

**Usage**

```
abic.expo.logistic(x, alpha.est, beta.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta

**Value**

The function `abic.expo.logistic()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.expo.logistic](#) for PP plot and [qq.expo.logistic](#) for QQ plot

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

## Values of AIC, BIC and LogLik for the data(dataset2)
abic.expo.logistic(dataset2, 5.31302, 139.04515)
```

---

<code>abic.expo.weibull</code>	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Exponentiated Weibull(EW) distribution</i>
--------------------------------	---

---

**Description**

The function `abic.expo.weibull()` gives the loglikelihood, AIC and BIC values assuming an Exponentiated Weibull(EW) distribution with parameters alpha and theta.

**Usage**

```
abic.expo.weibull(x, alpha.est, theta.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
theta.est	estimate of the parameter theta

**Value**

The function `abic.expo.weibull()` gives the loglikelihood, AIC and BIC values.

**References**

Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.

Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.

Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.

Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.expo.weibull](#) for PP plot and [qq.expo.weibull](#) for QQ plot

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est =1.026465, theta.est = 7.824943

## Values of AIC, BIC and LogLik for the data(stress)
abic.expo.weibull(stress, 1.026465, 7.824943)
```

---

<code>abic.flex.weibull</code>	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for flexible Weibull(FW) distribution</i>
--------------------------------	--

---

**Description**

The function `abic.flex.weibull()` gives the loglikelihood, AIC and BIC values assuming a flexible Weibull(FW) distribution with parameters alpha and beta.

**Usage**

```
abic.flex.weibull(x, alpha.est, beta.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta

**Value**

The function `abic.flex.weibull()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, Biometrika, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, Annals of Statistics, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, Journal of the Royal Statistical Society Series B 64, 1-34.

**See Also**

[pp.flex.weibull](#) for PP plot and [qq.flex.weibull](#) for QQ plot

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

## Values of AIC, BIC and LogLik for the data(repairtimes)
abic.flex.weibull(repairtimes, 0.07077507, 1.13181535)
```

---

abic.gen.exp	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for a sample from Generalized Exponential distribution</i>
--------------	---

---

**Description**

The function `abic.gen.exp()` gives the loglikelihood, AIC and BIC values assuming an Generalized Exponential distribution with parameters alpha and lambda. The function is based on the invariance property of the MLE.

**Usage**

```
abic.gen.exp(x, alpha.est, lambda.est)
```

**Arguments**

```
x                vector of observations
alpha.est        estimate of the parameter alpha
lambda.est       estimate of the parameter lambda
```

**Value**

The function `abic.gen.exp()` gives the loglikelihood, AIC and BIC values.

**References**

Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.

**See Also**

[pp.gen.exp](#) for PP plot and [qq.gen.exp](#) for QQ plot

**Examples**

```
## Load data set
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609
abic.gen.exp(bearings, 5.28321139, 0.03229609)
```

---

abic.gompertz	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Gompertz distribution</i>
---------------	--

---

**Description**

The function `abic.gompertz()` gives the loglikelihood, AIC and BIC values assuming an Gompertz distribution with parameters alpha and theta.

**Usage**

```
abic.gompertz(x, alpha.est, theta.est)
```

**Arguments**

```
x                vector of observations
alpha.est        estimate of the parameter alpha
theta.est        estimate of the parameter theta
```

**Value**

The function `abic.gompertz()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, Biometrika, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, Annals of Statistics, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, Journal of the Royal Statistical Society Series B 64, 1-34.

**See Also**

[pp.gompertz](#) for PP plot and [qq.gompertz](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

## Values of AIC, BIC and LogLik for the data(sys2)
abic.gompertz(sys2, 0.00121307, 0.00173329)
```

---

abic.gp.weibull	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for generalized power Weibull(GPW) distribution</i>
-----------------	--

---

**Description**

The function `abic.gp.weibull()` gives the loglikelihood, AIC and BIC values assuming an generalized power Weibull(GPW) distribution with parameters alpha and theta.

**Usage**

```
abic.gp.weibull(x, alpha.est, theta.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
theta.est	estimate of the parameter theta

**Value**

The function `abic.gp.weibull()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.gp.weibull](#) for PP plot and [qq.gp.weibull](#) for QQ plot

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

## Values of AIC, BIC and LogLik for the data(repairtimes)
abic.gp.weibull(repairtimes, 1.566093, 0.355321)
```

---

abic.gumbel	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Gumbel distribution</i>
-------------	--

---

**Description**

The function `abic.gumbel()` gives the loglikelihood, AIC and BIC values assuming an Gumbel distribution with parameters  $\mu$  and  $\sigma$ .

**Usage**

```
abic.gumbel(x, mu.est, sigma.est)
```

**Arguments**

x	vector of observations
mu.est	estimate of the parameter $\mu$
sigma.est	estimate of the parameter $\sigma$

**Value**

The function `abic.gumbel()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.gumbel](#) for PP plot and [qq.gumbel](#) for QQ plot

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

## Values of AIC, BIC and LogLik for the data(dataset2)
abic.gumbel(dataset2, 212.157, 151.768)
```

---

abic.inv.genexp	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Inverse Generalized Exponential(IGE) distribution</i>
-----------------	--

---

**Description**

The function `abic.inv.genexp()` gives the loglikelihood, AIC and BIC values assuming an Inverse Generalized Exponential(IGE) distribution with parameters alpha and lambda.

**Usage**

```
abic.inv.genexp(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.inv.genexp()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.inv.genexp](#) for PP plot and [qq.inv.genexp](#) for QQ plot

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

## Values of AIC, BIC and LogLik for the data(repairtimes)
abic.inv.genexp(repairtimes, 1.097807, 1.206889)
```

---

abic.lfr	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for linear failure rate(LFR) distribution</i>
----------	--

---

**Description**

The function `abic.lfr()` gives the loglikelihood, AIC and BIC values assuming an linear failure rate(LFR) distribution with parameters alpha and beta.

**Usage**

```
abic.lfr(x, alpha.est, beta.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta

**Value**

The function `abic.lfr()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, Biometrika, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, Annals of Statistics, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, Journal of the Royal Statistical Society Series B 64, 1-34.

**See Also**

[pp.lfr](#) for PP plot and [qq.lfr](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

## Values of AIC, BIC and LogLik for the data(sys2)
abic.lfr(sys2, 1.777673e-03, 2.777640e-06)
```

---

abic.log.gamma	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for log-gamma(LG) distribution</i>
----------------	---

---

**Description**

The function `abic.log.gamma()` gives the loglikelihood, AIC and BIC values assuming an log-gamma(LG) distribution with parameters alpha and lambda.

**Usage**

```
abic.log.gamma(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.log.gamma()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.log.gamma](#) for PP plot and [qq.log.gamma](#) for QQ plot

**Examples**

```
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

## Values of AIC, BIC and LogLik for the data(conductors)
abic.log.gamma(conductors, 0.0088741, 0.6059935)
```

---

abic.logis.exp	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Logistic-Exponential(LE) distribution</i>
----------------	--

---

**Description**

The function `abic.logis.exp()` gives the loglikelihood, AIC and BIC values assuming an Logistic-Exponential(LE) distribution with parameters alpha and lambda.

**Usage**

```
abic.logis.exp(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.logis.exp()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.logis.exp](#) for PP plot and [qq.logis.exp](#) for QQ plot

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

## Values of AIC, BIC and LogLik for the data(bearings)
abic.logis.exp(bearings, 2.36754, 0.01059)
```

---

`abic.logis.rayleigh`     *Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Logistic-Rayleigh(LR) distribution*

---

**Description**

The function `abic.logis.rayleigh()` gives the loglikelihood, AIC and BIC values assuming an Logistic-Rayleigh(LR) distribution with parameters alpha and lambda.

**Usage**

```
abic.logis.rayleigh(x, alpha.est, lambda.est)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda

**Value**

The function `abic.logis.rayleigh()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, Biometrika, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, Annals of Statistics, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, Journal of the Royal Statistical Society Series B 64, 1-34.

**See Also**

[pp.logis.rayleigh](#) for PP plot and [qq.logis.rayleigh](#) for QQ plot

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

## Values of AIC, BIC and LogLik for the data(stress)
abic.logis.rayleigh(stress, 1.4779388, 0.2141343)
```

---

abic.loglog	<i>Akaike information criterion (AIC) and Bayesian/ Schwartz information criterion (BIC)/ (SIC) for a sample from Loglog distribution</i>
-------------	---

---

**Description**

The function `abic.loglog( )` gives the loglikelihood, AIC and BIC values assuming Loglog distribution with parameters alpha and lambda. The function is based on the invariance property of the MLE.

**Usage**

```
abic.loglog(x, alpha.est, lambda.est)
```

**Arguments**

x                    vector of observations  
alpha.est            estimate of the parameter alpha  
lambda.est           estimate of the parameter lambda

**Value**

The function `abic.loglog( )` gives the loglikelihood, AIC and BIC values.

**References**

Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.  
Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.  
Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.  
Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.  
Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[qq.loglog](#) for QQ plot and [ks.loglog](#) function

**Examples**

```
## Load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

## Values of AIC, BIC and LogLik for the data(sys2)
abic.loglog(sys2, 0.9058689, 1.0028228)
```

---

abic.moe	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for the Marshall-Olkin Extended Exponential(MOEE) distribution</i>
----------	---

---

**Description**

The function `abic.moe()` gives the loglikelihood, AIC and BIC values assuming an MOEE distribution with parameters alpha and lambda.

**Usage**

```
abic.moe(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.moew()` gives the loglikelihood, AIC and BIC values.

**References**

Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.

**See Also**

[pp.moew](#) for PP plot and [qq.moew](#) for QQ plot

**Examples**

```
## Load data set
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576
abic.moew(stress, 75.67982, 1.67576)
```

---

abic.moew	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for the Marshall-Olkin Extended Weibull(MOEW) distribution</i>
-----------	---

---

**Description**

The function `abic.moew()` gives the loglikelihood, AIC and BIC values assuming an MOEW distribution with parameters alpha and lambda.

**Usage**

```
abic.moew(x, alpha.est, lambda.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda

**Value**

The function `abic.moew()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.moew](#) for PP plot and [qq.moew](#) for QQ plot

**Examples**

```
## Load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

## Values of AIC, BIC and LogLik for the data(sys2)
abic.moew(sys2, 0.3035937, 279.2177754)
```

---

abic.weibull.ext	<i>Akaike information criterion (AIC) and Bayesian information criterion (BIC) for Weibull Extension(WE) distribution</i>
------------------	---

---

**Description**

The function `abic.weibull.ext()` gives the loglikelihood, AIC and BIC values assuming an Weibull Extension(WE) distribution with parameters alpha and beta.

**Usage**

```
abic.weibull.ext(x, alpha.est, beta.est)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta

**Value**

The function `abic.weibull.ext()` gives the loglikelihood, AIC and BIC values.

**References**

- Akaike, H. (1978). *A new look at the Bayes procedure*, *Biometrika*, 65, 53-59.
- Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge University Press, London.
- Konishi., S. and Kitagawa, G.(2008). *Information Criteria and Statistical Modeling*, Springer Science+Business Media, LLC.
- Schwarz, S. (1978). *Estimating the dimension of the model*, *Annals of Statistics*, 6, 461-464.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and van der Linde, A. (2002). *Bayesian measures of complexity and fit*, *Journal of the Royal Statistical Society Series B* 64, 1-34.

**See Also**

[pp.weibull.ext](#) for PP plot and [qq.weibull.ext](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

## Values of AIC, BIC and LogLik for the data(sys2)
abic.weibull.ext(sys2, 0.00019114, 0.14696242)
```

---

bearings

*bearings*

---

**Description**

Several data sets related to life test are available in the `reliaR` package, which have been taken from the literature.

**Usage**

```
data(bearings)
```

**Format**

A vector containing 23 observations.

**Details**

The data given here arose in tests on endurance of deep groove ball bearings. The data are the number of million revolutions before failure for each of the 23 ball bearings in the life test.

**References**

Lawless, J. F. (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

**Examples**

```
## Load data sets
data(bearings)
## Histogram for bearings
hist(bearings)
```

---

 BurrX

---

*The BurrX (Generalized Rayleigh) distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the BurrX distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
dburrX(x, alpha, lambda, log = FALSE)
pburrX(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qburrX(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rburrX(n, alpha, lambda)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The BurrX distribution has density

$$f(x; \alpha, \lambda) = 2\alpha\lambda^2 x e^{-(\lambda x)^2} \left\{1 - e^{-(\lambda x)^2}\right\}^{\alpha-1}; (\alpha, \lambda) > 0, x > 0.$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dburrX gives the density, pburrX gives the distribution function, qburrX gives the quantile function, and rburrX generates random deviates.

**References**

Kundu, D., and Raqab, M.Z. (2005). *Generalized Rayleigh Distribution: Different Methods of Estimation*, Computational Statistics and Data Analysis, 49, 187-200.

Surles, J.G., and Padgett, W.J. (2005). *Some properties of a scaled Burr type X distribution*, Journal of Statistical Planning and Inference, 128, 271-280.

Raqab, M.Z., and Kundu, D. (2006). *Burr Type X distribution: revisited*, Journal of Probability and Statistical Sciences, 4(2), 179-193.

**See Also**

[.Random.seed](#) about random number; [sburrX](#) for BurrX survival / hazard etc. functions

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

dburrX(bearings, 1.1989515, 0.0130847, log = FALSE)
pburrX(bearings, 1.1989515, 0.0130847, lower.tail = TRUE, log.p = FALSE)
qburrX(0.25, 1.1989515, 0.0130847, lower.tail=TRUE, log.p = FALSE)
rburrX(30, 1.1989515, 0.0130847)
```

---

BurrXsurvival

*Survival related functions for the BurrX distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the BurrX distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
crf.burrX(x, t = 0, alpha, lambda)
h BurrX(x, alpha, lambda)
hra.burrX(x, alpha, lambda)
sburrX(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

**Details**

The hazard function is defined by

$$h(x) = \frac{f(x)}{1 - F(x)}, \quad t > 0, 0 < F(x) < 1,$$

where  $f(\cdot)$  and  $F(\cdot)$  are the pdf and cdf, respectively. The behavior of  $h(x)$  allows one to characterize the aging of the units. For example, if the failure rate is increasing (IFR class), then the units age with time. If  $h(x)$  is decreasing (DFR class), then the units improve in performance with time. Finally, if  $h(x)$  is constant, then the lifetime distribution is necessarily exponential.

There are two more aging indicators which are the following:

The failure rate average (FRA) of X is given by

$$FRA(x) = \frac{H(x)}{x} = \frac{\int_0^x h(x) dx}{x}, \quad x > 0,$$

where  $H(x)$  is the cumulative hazard function. An analysis for  $FRA(x)$  on  $x$  permits to obtain the IFRA and DFRA classes.

The survival/reliability function (s.f.) and the conditional survival of X are defined by

$$R(x) = 1 - F(x) \quad \text{and} \quad R(x|t) = \frac{R(x+t)}{R(x)}, \quad x > 0, t > 0, R(\cdot) > 0,$$

respectively, where  $F(\cdot)$  is the cdf of X. Similarly to  $h(x)$  and  $FRA(x)$ , the distribution of X belongs to the new better than used (NBU), exponential, or new worse than used (NWU) classes, when  $R(x|t) < R(x)$ ,  $R(x|t) = R(x)$ , or  $R(x|t) > R(x)$ , respectively.

**Value**

`crf.burrX` gives the conditional reliability function (crf), `h BurrX` gives the hazard function, `hra.burrX` gives the hazard rate average (HRA) function, and `sburrX` gives the survival function for the BurrX distribution.

## References

Kundu, D., and Raqab, M.Z. (2005). *Generalized Rayleigh Distribution: Different Methods of Estimation*, Computational Statistics and Data Analysis, 49, 187-200.

Lawless, J.F.(2003). *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

## See Also

[dburrX](#) for other BurrX distribution related functions;

## Examples

```
## load data set
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

## Reliability indicators for data(bearings):

## Reliability function
sburrX(bearings, 1.1989515, 0.0130847)

## Hazard function
hburrX(bearings, 1.1989515, 0.0130847)

## hazard rate average(hra)
hra.burrX(bearings, 1.1989515, 0.0130847)

## Conditional reliability function (age component=0)
crf.burrX(bearings, 0.00, 1.1989515, 0.0130847)

## Conditional reliability function (age component=3.0)
crf.burrX(bearings, 3.0, 1.1989515, 0.0130847)
```

## Description

Density, distribution function, quantile function and random generation for the Chen distribution with shape parameter beta and scale parameter lambda.

**Usage**

```
dchen(x, beta, lambda, log = FALSE)
pchen(q, beta, lambda, lower.tail = TRUE, log.p = FALSE)
qchen(p, beta, lambda, lower.tail = TRUE, log.p = FALSE)
rchen(n, beta, lambda)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
beta	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Chen distribution has density

$$f(x; \lambda, \beta) = \lambda \beta x^{\beta-1} \exp(x^\beta) \exp[\lambda \{1 - \exp(x^\beta)\}]; (\lambda, \beta) > 0, x > 0,$$

where  $\beta$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dchen gives the density, pchen gives the distribution function, qchen gives the quantile function, and rchen generates random deviates.

**References**

- Chen, Z. (2000). *A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function*, Statistics & Probability Letters, 49, 155-161.
- Murthy, D.N.P., Xie, M. and Jiang, R. (2004). *Weibull Models*, Wiley, New York.
- Pham, H. (2006). *System Software Reliability*, Springer-Verlag.
- Pham, H. and Lai, C.D. (2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

**See Also**

[.Random.seed](#) about random number; [schen](#) for Chen survival / hazard etc. functions

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

dchen(sys2, 0.262282404, 0.007282371, log = FALSE)
pchen(sys2, 0.262282404, 0.007282371, lower.tail = TRUE,
      log.p = FALSE)
qchen(0.25, 0.262282404, 0.007282371, lower.tail = TRUE, log.p = FALSE)
rchen(10, 0.262282404, 0.007282371)
```

---

Chensurvival

*Survival related functions for the Chen distribution*


---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Chen distribution with shape parameter beta and scale parameter lambda.

**Usage**

```
crf.chen(x, t = 0, beta, lambda)
hchen(x, beta, lambda)
hra.chen(x, beta, lambda)
schen(x, beta, lambda)
```

**Arguments**

x	vector of quantiles.
beta	shape parameter.
lambda	scale parameter.
t	age component.

**Value**

crf.chen gives the conditional reliability function (crf), hchen gives the hazard function, hra.chen gives the hazard rate average (HRA) function, and schen gives the survival function for the Chen distribution.

**References**

Chen, Z.(2000). *A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function*, Statistics and Probability Letters, 49, 155-161.

Pham, H. (2003). *Handbook of Reliability Engineering*, Springer-Verlag.

**See Also**

[dchen](#) for other Chen distribution related functions

**Examples**

```
## Maximum Likelihood(ML) Estimates of beta & lambda
## beta.est = 0.262282404, lambda.est = 0.007282371
## Load data sets
data(sys2)

## Reliability indicators:

## Reliability function
schen(sys2, 0.262282404, 0.007282371)

## Hazard function
hchen(sys2, 0.262282404, 0.007282371)

## hazard rate average(hra)
hra.chen(sys2, 0.262282404, 0.007282371)

## Conditional reliability function (age component=0)
crf.chen(sys2, 0.00, 0.262282404, 0.007282371)

## Conditional reliability function (age component=3.0)
crf.chen(sys2, 3.0, 0.262282404, 0.007282371)
```

---

conductors

*Accelerated life test data*

---

**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

```
data(conductors)
```

**Format**

A vector containing 59 observations.

**Details**

The data is obtained from Lawless(2003, pp. 267) and it represents the failure times of 59 conductors from an accelerated life test. Failure times are in hours, and there are no censored observations.

## References

Lawless, J. F. (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

## Examples

```
## Load data sets
data(conductors)
## Histogram for conductors
hist(conductors)
```

---

dataset2

*Controller Dataset*

---

## Description

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

## Usage

```
data(dataset2)
```

## Format

A vector containing 111 observations.

## Details

The data is obtained from Lyu(1996) and is given in chapter 11 as DATASET2. The data set contains 36 months of defect-discovery times for a release of Controller Software consisting of about 500,000 lines of code installed on over 100,000 controllers.

## References

Lyu, M. R. (1996). *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, <http://www.cse.cuhk.edu.hk/~lyu/book/reliability/>

## Examples

```
## Load data sets
data(dataset2)
## Histogram for dataset2
hist(dataset2)
```

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponential Power distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
crf.exp.power(x, t = 0, alpha, lambda)
hexp.power(x, alpha, lambda)
hra.exp.power(x, alpha, lambda)
sexp.power(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	tilt parameter.
lambda	scale parameter.
t	age component.

**Value**

crf.exp.power gives the conditional reliability function (crf), hexp.power gives the hazard function, hra.exp.power gives the hazard rate average (HRA) function, and sexp.power gives the survival function for the Exponential Power distribution.

**References**

- Chen, Z.(1999). *Statistical inference about the shape parameter of the exponential power distribution*, Journal :Statistical Papers, Vol. 40(4), 459-468.
- Pham, H. and Lai, C.D.(2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.
- Smith, R.M. and Bain, L.J.(1975). *An exponential power life-test distribution*, Communications in Statistics - Simulation and Computation, Vol.4(5), 469 - 481

**See Also**

[dexp.power](#) for other Exponential Power distribution related functions

**Examples**

```

## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

## Reliability indicators:

## Reliability function
sexp.power(sys2, 0.905868898, 0.001531423)

## Hazard function
hexp.power(sys2, 0.905868898, 0.001531423)

## hazard rate average(hra)
hra.exp.power(sys2, 0.905868898, 0.001531423)

## Conditional reliability function (age component=0)
crf.exp.power(sys2, 0.00, 0.905868898, 0.001531423)

## Conditional reliability function (age component=3.0)
crf.exp.power(sys2, 3.0, 0.905868898, 0.001531423)

```

ExpExt

*The Exponential Extension(EE) distribution***Description**

Density, distribution function, quantile function and random generation for the Exponential Extension(EE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```

dexp.ext(x, alpha, lambda, log = FALSE)
pexp.ext(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qexp.ext(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rexp.ext(n, alpha, lambda)

```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Exponential Extension(EE) distribution has density

$$f(x) = \alpha\lambda(1 + \lambda x)^{\alpha-1} \exp\{1 - (1 + \lambda x)^\alpha\}; x \geq 0, \alpha > 0, \lambda > 0.$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dexp.ext gives the density, pexp.ext gives the distribution function, qexp.ext gives the quantile function, and rexp.ext generates random deviates.

**References**

Nikulin, M. and Haghghi, F. (2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

**See Also**

[.Random.seed](#) about random number; [sexp.ext](#) for ExpExt survival / hazard etc. functions

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04
dexp.ext(sys2, 1.012556e+01, 1.5848e-04, log = FALSE)
pexp.ext(sys2, 1.012556e+01, 1.5848e-04, lower.tail = TRUE, log.p = FALSE)
qexp.ext(0.25, 1.012556e+01, 1.5848e-04, lower.tail=TRUE, log.p = FALSE)
rexp.ext(30, 1.012556e+01, 1.5848e-04)
```

---

ExpExtSurvival

*Survival related functions for the Exponential Extension(EE) distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponential Extension(EE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
crf.exp.ext(x, t = 0, alpha, lambda)
hexp.ext(x, alpha, lambda)
hra.exp.ext(x, alpha, lambda)
sexp.ext(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

**Value**

crf.exp.ext gives the conditional reliability function (crf), hexp.ext gives the hazard function, hra.exp.ext gives the hazard rate average (HRA) function, and sexp.ext gives the survival function for the Exponential Extension(EE) distribution.

**References**

Nikulin, M. and Haghghi, F.(2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

**See Also**

[dexp.ext](#) for other Exponential Extension(EE) distribution related functions;

**Examples**

```
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

## Reliability indicators for data(sys2):

## Reliability function
sexp.ext(sys2, 1.0126e+01, 1.5848e-04)

## Hazard function
hexp.ext(sys2, 1.0126e+01, 1.5848e-04)

## hazard rate average(hra)
hra.exp.ext(sys2, 1.0126e+01, 1.5848e-04)

## Conditional reliability function (age component=0)
crf.exp.ext(sys2, 0.00, 1.0126e+01, 1.5848e-04)

## Conditional reliability function (age component=3.0)
crf.exp.ext(sys2, 3.0, 1.0126e+01, 1.5848e-04)
```

ExpoLogistic

*The Exponentiated Logistic(EL) distribution***Description**

Density, distribution function, quantile function and random generation for the Exponentiated Logistic(EL) distribution with shape parameter alpha and scale parameter beta.

**Usage**

```
dexpo.logistic(x, alpha, beta, log = FALSE)
pexpo.logistic(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qexpo.logistic(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rexpo.logistic(n, alpha, beta)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
beta	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Exponentiated Logistic(EL) distribution has density

$$f(x; \alpha, \beta) = \frac{\alpha}{\beta} \exp\left(-\frac{x}{\beta}\right) \left\{1 + \exp\left(-\frac{x}{\beta}\right)\right\}^{-(\alpha+1)}; (\alpha, \beta) > 0, x > 0$$

where  $\alpha$  and  $\beta$  are the shape and scale parameters, respectively.

**Value**

dexpo.logistic gives the density, pexpo.logistic gives the distribution function, qexpo.logistic gives the quantile function, and rexpo.logistic generates random deviates.

**References**

- Ali, M.M., Pal, M. and Woo, J. (2007). *Some Exponentiated Distributions*, The Korean Communications in Statistics, 14(1), 93-109.
- Shirke, D.T., Kumbhar, R.R. and Kundu, D. (2005). *Tolerance intervals for exponentiated scale family of distributions*, Journal of Applied Statistics, 32, 1067-1074

**See Also**

[.Random.seed](#) about random number; [sexpo.logistic](#) for Exponentiated Logistic(EL) survival / hazard etc. functions

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

dexpo.logistic(dataset2, 5.31302, 139.04515, log = FALSE)
pexpo.logistic(dataset2, 5.31302, 139.04515, lower.tail = TRUE, log.p = FALSE)
qexpo.logistic(0.25, 5.31302, 139.04515, lower.tail=TRUE, log.p = FALSE)
rexp.logistic(30, 5.31302, 139.04515)
```

---

ExpoLogisticsurvival *Survival related functions for the Exponentiated Logistic(EL) distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponentiated Logistic(EL) distribution with shape parameter alpha and scale parameter beta.

**Usage**

```
crf.expo.logistic(x, t = 0, alpha, beta)
hexpo.logistic(x, alpha, beta)
hra.expo.logistic(x, alpha, beta)
sexpo.logistic(x, alpha, beta)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
beta	scale parameter.
t	age component.

**Value**

`crf.expo.logistic` gives the conditional reliability function (crf), `hexpo.logistic` gives the hazard function, `hra.expo.logistic` gives the hazard rate average (HRA) function, and `sexpo.logistic` gives the survival function for the Exponentiated Logistic(EL) distribution.

## References

Ali, M.M., Pal, M. and Woo, J. (2007). *Some Exponentiated Distributions*, The Korean Communications in Statistics, 14(1), 93-109.

Shirke, D.T., Kumbhar, R.R. and Kundu, D.(2005). *Tolerance intervals for exponentiated scale family of distributions*, Journal of Applied Statistics, 32, 1067-1074

## See Also

[dexpo.logistic](#) for other Exponentiated Logistic(EL) distribution related functions;

## Examples

```
## load data set
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

## Reliability indicators for data(dataset2):

## Reliability function
sexpo.logistic(dataset2, 5.31302, 139.04515)

## Hazard function
hexpo.logistic(dataset2, 5.31302, 139.04515)

## hazard rate average(hra)
hra.expo.logistic(dataset2, 5.31302, 139.04515)

## Conditional reliability function (age component=0)
crf.expo.logistic(dataset2, 0.00, 5.31302, 139.04515)

## Conditional reliability function (age component=3.0)
crf.expo.logistic(dataset2, 3.0, 5.31302, 139.04515)
```

---

ExpoWeibull

*The Exponentiated Weibull(EW) distribution*

---

## Description

Density, distribution function, quantile function and random generation for the Exponentiated Weibull(EW) distribution with shape parameters alpha and theta.

## Usage

```
dexpo.weibull(x, alpha, theta, log = FALSE)
pexpo.weibull(q, alpha, theta, lower.tail = TRUE, log.p = FALSE)
qexpo.weibull(p, alpha, theta, lower.tail = TRUE, log.p = FALSE)
rexpo.weibull(n, alpha, theta)
```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>alpha</code>	shape parameter.
<code>theta</code>	shape parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Exponentiated Weibull(EW) distribution has density

$$f(x; \alpha, \theta) = \alpha \theta x^{\alpha-1} e^{-x^\alpha} \{1 - \exp(-x^\alpha)\}^{\theta-1}; (\alpha, \theta) > 0, x > 0$$

where  $\alpha$  and  $\theta$  are the shape and scale parameters, respectively.

**Value**

`dexpo.weibull` gives the density, `pexpo.weibull` gives the distribution function, `qexpo.weibull` gives the quantile function, and `rexpo.weibull` generates random deviates.

**References**

Mudholkar, G.S. and Srivastava, D.K. (1993). *Exponentiated Weibull family for analyzing bathtub failure-rate data*, IEEE Transactions on Reliability, 42(2), 299-302.

Murthy, D.N.P., Xie, M. and Jiang, R. (2003). *Weibull Models*, Wiley, New York.

Nassar, M.M., and Eissa, F. H. (2003). *On the Exponentiated Weibull Distribution*, Communications in Statistics - Theory and Methods, 32(7), 1317-1336.

**See Also**

[.Random.seed](#) about random number; [sexpo.weibull](#) for Exponentiated Weibull(EW) survival / hazard etc. functions

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est =1.026465, theta.est = 7.824943

dexpo.weibull(stress, 1.026465, 7.824943, log = FALSE)
pexpo.weibull(stress, 1.026465, 7.824943, lower.tail = TRUE, log.p = FALSE)
qexpo.weibull(0.25, 1.026465, 7.824943, lower.tail=TRUE, log.p = FALSE)
rexpo.weibull(30, 1.026465, 7.824943)
```

---

ExpoWeibullsurvival     *Survival related functions for the Exponentiated Weibull(EW) distribution*

---

## Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Exponentiated Weibull(EW) distribution with shape parameters alpha and theta.

## Usage

```
crf.expo.weibull(x, t = 0, alpha, theta)
hexpo.weibull(x, alpha, theta)
hra.expo.weibull(x, alpha, theta)
sexpo.weibull(x, alpha, theta)
```

## Arguments

x	vector of quantiles.
alpha	shape parameter.
theta	shape parameter.
t	age component.

## Value

crf.expo.weibull gives the conditional reliability function (crf), hexpo.weibull gives the hazard function, hra.expo.weibull gives the hazard rate average (HRA) function, and sexpo.weibull gives the survival function for the Exponentiated Weibull(EW) distribution.

## References

Mudholkar, G.S. and Srivastava, D.K. (1993). *Exponentiated Weibull family for analyzing bathtub failure-rate data*, IEEE Transactions on Reliability, 42(2), 299-302.

Murthy, D.N.P., Xie, M. and Jiang, R. (2003). *Weibull Models*, Wiley, New York.

Nassar, M.M., and Eissa, F. H. (2003). *On the Exponentiated Weibull Distribution*, Communications in Statistics - Theory and Methods, 32(7), 1317-1336.

## See Also

[dexpo.weibull](#) for other Exponentiated Weibull(EW) distribution related functions;

**Examples**

```

## load data set
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est =1.026465, theta.est = 7.824943

## Reliability indicators for data(stress):

## Reliability function
sexpo.weibull(stress, 1.026465, 7.824943)

## Hazard function
hexpo.weibull(stress, 1.026465, 7.824943)

## hazard rate average(hra)
hra.expo.weibull(stress, 1.026465, 7.824943)

## Conditional reliability function (age component=0)
crf.expo.weibull(stress, 0.00, 1.026465, 7.824943)

## Conditional reliability function (age component=3.0)
crf.expo.weibull(stress, 3.0, 1.026465, 7.824943)

```

ExpPower

*The Exponential Power distribution***Description**

Density, distribution function, quantile function and random generation for the Exponential Power distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```

dexp.power(x, alpha, lambda, log = FALSE)
pexp.power(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qexp.power(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rexp.power(n, alpha, lambda)

```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The probability density function of exponential power distribution is

$$f(x; \alpha, \lambda) = \alpha \lambda^\alpha x^{\alpha-1} e^{-(\lambda x)^\alpha} \exp \left\{ 1 - e^{-(\lambda x)^\alpha} \right\}; (\alpha, \lambda) > 0, x > 0.$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dexp.power gives the density, pexp.power gives the distribution function, qexp.power gives the quantile function, and rexp.power generates random deviates.

**References**

Chen, Z.(1999). *Statistical inference about the shape parameter of the exponential power distribution*, Journal :Statistical Papers, Vol. 40(4), 459-468.

Pham, H. and Lai, C.D.(2007). *On Recent Generalizations of the Weibull Distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

Smith, R.M. and Bain, L.J.(1975). *An exponential power life-test distribution*, Communications in Statistics - Simulation and Computation, Vol.4(5), 469 - 481

**See Also**

[.Random.seed](#) about random number; [sexp.power](#) for Exponential Power distribution survival / hazard etc. functions;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

dexp.power(sys2, 0.905868898, 0.001531423, log = FALSE)
pexp.power(sys2, 0.905868898, 0.001531423, lower.tail = TRUE, log.p = FALSE)
qexp.power(0.25, 0.905868898, 0.001531423, lower.tail=TRUE, log.p = FALSE)
rexp.power(30, 0.905868898, 0.001531423)
```

---

FlexWeibull

*The flexible Weibull(FW) distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the flexible Weibull(FW) distribution with parameters alpha and beta.

**Usage**

```
dflex.weibull(x, alpha, beta, log = FALSE)
pflex.weibull(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qflex.weibull(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rflex.weibull(n, alpha, beta)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	parameter.
beta	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The flexible Weibull(FW) distribution has density

$$f(x) = \left( \alpha + \frac{\beta}{x^2} \right) \exp \left( \alpha x - \frac{\beta}{x} \right) \exp \left\{ - \exp \left( \alpha x - \frac{\beta}{x} \right) \right\}; x \geq 0, \alpha > 0, \beta > 0.$$

where  $\alpha$  and  $\beta$  are the shape and scale parameters, respectively.

**Value**

dflex.weibull gives the density, pflex.weibull gives the distribution function, qflex.weibull gives the quantile function, and rflex.weibull generates random deviates.

**References**

Bebbington, M., Lai, C.D. and Zitikis, R. (2007). *A flexible Weibull extension*, Reliability Engineering and System Safety, 92, 719-726.

**See Also**

[.Random.seed](#) about random number; [sflex.weibull](#) for flexible Weibull(FW) survival / hazard etc. functions

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535
```

```
dflex.weibull(repairtimes, 0.07077507, 1.13181535, log = FALSE)
pflex.weibull(repairtimes, 0.07077507, 1.13181535, lower.tail = TRUE, log.p = FALSE)
qflex.weibull(0.25, 0.07077507, 1.13181535, lower.tail=TRUE, log.p = FALSE)
rflex.weibull(30, 0.07077507, 1.13181535)
```

---

FlexWeibullsurvival      *Survival related functions for the flexible Weibull(FW) distribution*

---

### Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the flexible Weibull(FW) distribution with parameters alpha and beta.

### Usage

```
crf.flex.weibull(x, t = 0, alpha, beta)
hflex.weibull(x, alpha, beta)
hra.flex.weibull(x, alpha, beta)
sflex.weibull(x, alpha, beta)
```

### Arguments

x	vector of quantiles.
alpha	parameter.
beta	parameter.
t	age component.

### Value

`crf.flex.weibull` gives the conditional reliability function (crf), `hflex.weibull` gives the hazard function, `hra.flex.weibull` gives the hazard rate average (HRA) function, and `sflex.weibull` gives the survival function for the flexible Weibull(FW) distribution.

### References

Bebbington, M., Lai, C.D. and Zitikis, R. (2007). *A flexible Weibull extension*, Reliability Engineering and System Safety, 92, 719-726.

### See Also

[dflex.weibull](#) for other flexible Weibull(FW) distribution related functions;

**Examples**

```

## load data set
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

## Reliability indicators for data(repairtimes):

## Reliability function
sflex.weibull(repairtimes, 0.07077507, 1.13181535)

## Hazard function
hflex.weibull(repairtimes, 0.07077507, 1.13181535)

## hazard rate average(hra)
hra.flex.weibull(repairtimes, 0.07077507, 1.13181535)

## Conditional reliability function (age component=0)
crf.flex.weibull(repairtimes, 0.00, 0.07077507, 1.13181535)

## Conditional reliability function (age component=3.0)
crf.flex.weibull(repairtimes, 3.0, 0.07077507, 1.13181535)

```

GenExp

*The Generalized Exponential (GE) distribution***Description**

Density, distribution function, quantile function and random generation for the Generalized Exponential (GE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```

dgen.exp(x, alpha, lambda, log = FALSE)
pgen.exp(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qgen.exp(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rgen.exp(n, alpha, lambda)

```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The generalized exponential distribution has density

$$f(x; \alpha, \lambda) = \alpha \lambda x e^{-\lambda x} \{1 - e^{-\lambda x}\}^{\alpha-1}; (\alpha, \lambda) > 0, x > 0.$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dgen.exp gives the density, pgen.exp gives the distribution function, qgen.exp gives the quantile function, and rgen.exp generates random deviates.

**References**

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*. Biometrical Journal, 43(1), 117 - 130.

Gupta, R. D. and Kundu, D. (1999). *Generalized exponential distributions*. Australian and New Zealand Journal of Statistics, 41(2), 173 - 188.

**See Also**

[.Random.seed](#) about random number; [sgen.exp](#) for GE survival / hazard etc. functions

**Examples**

```
## Load data set
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609

dgen.exp(bearings, 5.28321139, 0.03229609, log = FALSE)
pgen.exp(bearings, 5.28321139, 0.03229609, lower.tail = TRUE,
         log.p = FALSE)
qgen.exp(0.25, 5.28321139, 0.03229609, lower.tail = TRUE, log.p = FALSE)
rgen.exp(10, 5.28321139, 0.03229609)
```

---

GenExpsurvival

*Survival related functions for the Generalized Exponential (GE) distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Generalized Exponential (GE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
crf.gen.exp(x, t = 0, alpha, lambda)
hgen.exp(x, alpha, lambda)
hra.gen.exp(x, alpha, lambda)
sgen.exp(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

**Value**

`crf.gen.exp` gives the conditional reliability function (crf), `hgen.exp` gives the hazard function, `hra.gen.exp` gives the hazard rate average (HRA) function, and `sgen.exp` gives the survival function for the GE distribution.

**References**

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*. Biometrical Journal, 43(1), 117 - 130.

Gupta, R. D. and Kundu, D. (1999). *Generalized exponential distributions*. Australian and New Zealand Journal of Statistics, 41(2), 173 - 188.

**See Also**

[dgen.exp](#) for other GE distribution related functions;

**Examples**

```
## load data set
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609
sgen.exp(bearings, 5.28321139, 0.03229609)
hgen.exp(bearings, 5.28321139, 0.03229609)
hra.gen.exp(bearings, 5.28321139, 0.03229609)
crf.gen.exp(bearings, 20.0, 5.28321139, 0.03229609)
```

Gompertz

*The Gompertz distribution***Description**

Density, distribution function, quantile function and random generation for the Gompertz distribution with shape parameter  $\alpha$  and scale parameter  $\theta$ .

**Usage**

```

dgomperz(x, alpha, theta, log = FALSE)
pgomperz(q, alpha, theta, lower.tail = TRUE, log.p = FALSE)
qgomperz(p, alpha, theta, lower.tail = TRUE, log.p = FALSE)
rgomperz(n, alpha, theta)

```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>alpha</code>	shape parameter.
<code>theta</code>	scale parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$ .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Gompertz distribution has density

$$f(x) = \theta e^{\alpha x} \exp \left\{ \frac{\theta}{\alpha} (1 - e^{\alpha x}) \right\}; x \geq 0, \theta > 0, -\infty < \alpha < \infty.$$

where  $\alpha$  and  $\theta$  are the shape and scale parameters, respectively.

**Value**

`dgomperz` gives the density, `pgomperz` gives the distribution function, `qgomperz` gives the quantile function, and `rgomperz` generates random deviates.

**References**

Marshall, A. W., Olkin, I. (2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[.Random.seed](#) about random number; [sgompertz](#) for Gompertz survival / hazard etc. functions

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

dgomperz(sys2, 0.00121307, 0.00173329, log = FALSE)
pgomperz(sys2, 0.00121307, 0.00173329, lower.tail = TRUE, log.p = FALSE)
qgomperz(0.25, 0.00121307, 0.00173329, lower.tail=TRUE, log.p = FALSE)
rgomperz(30, 0.00121307, 0.00173329)
```

---

Gompertzsurvival

*Survival related functions for the Gompertz distribution*


---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Gompertz distribution with shape parameter alpha and scale parameter theta.

**Usage**

```
crf.gompertz(x, t = 0, alpha, theta)
hgompertz(x, alpha, theta)
hra.gompertz(x, alpha, theta)
sgompertz(x, alpha, theta)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
theta	scale parameter.
t	age component.

**Value**

crf.gompertz gives the conditional reliability function (crf), hgompertz gives the hazard function, hra.gompertz gives the hazard rate average (HRA) function, and sgompertz gives the survival function for the Gompertz distribution.

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[dgomPERTZ](#) for other Gompertz distribution related functions;

**Examples**

```
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

## Reliability indicators for data(sys2):

## Reliability function
sgompertz(sys2, 0.00121307, 0.00173329)

## Hazard function
hgomPERTZ(sys2, 0.00121307, 0.00173329)

## hazard rate average(hra)
hra.gompertz(sys2, 0.00121307, 0.00173329)

## Conditional reliability function (age component=0)
crf.gompertz(sys2, 0.00, 0.00121307, 0.00173329)

## Conditional reliability function (age component=3.0)
crf.gompertz(sys2, 3.0, 0.00121307, 0.00173329)
```

---

GPWeibull

*The generalized power Weibull(GPW) distribution*

---

**Description**

Density, distribution function, quantile function and random generation for the generalized power Weibull(GPW) distribution with shape parameters alpha and theta.

**Usage**

```
dgp.weibull(x, alpha, theta, log = FALSE)
pgp.weibull(q, alpha, theta, lower.tail = TRUE, log.p = FALSE)
qgp.weibull(p, alpha, theta, lower.tail = TRUE, log.p = FALSE)
rgp.weibull(n, alpha, theta)
```

**Arguments**

x, q            vector of quantiles.  
p               vector of probabilities.

n	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
alpha	shape parameter.
theta	shape parameter.
log, log.p	logical; if TRUE, probabilities p are given as <code>log(p)</code> .
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

### Details

The generalized power Weibull(GPW) distribution has density

$$f(x) = \alpha \theta x^{\alpha-1} (1+x^\alpha)^{\theta-1} \exp\left\{1 - (1+x^\alpha)^\theta\right\}; x \geq 0, \alpha > 0, \theta > 0.$$

where  $\alpha$  and  $\theta$  are the shape and scale parameters, respectively.

### Value

`dgp.weibull` gives the density, `pgp.weibull` gives the distribution function, `qgp.weibull` gives the quantile function, and `rgp.weibull` generates random deviates.

### References

Nikulin, M. and Haghghi, F. (2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

Pham, H. and Lai, C.D. (2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

### See Also

[.Random.seed](#) about random number; [sgp.weibull](#) for generalized power Weibull(GPW) survival / hazard etc. functions

### Examples

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

dgp.weibull(repairtimes, 1.566093, 0.355321, log = FALSE)
pgp.weibull(repairtimes, 1.566093, 0.355321, lower.tail = TRUE, log.p = FALSE)
qgp.weibull(0.25, 1.566093, 0.355321, lower.tail=TRUE, log.p = FALSE)
rgp.weibull(30, 1.566093, 0.355321)
```

---

GPWeibullsurvival	<i>Survival related functions for the generalized power Weibull(GPW) distribution</i>
-------------------	---

---

### Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the generalized power Weibull(GPW) distribution with shape parameters alpha and theta.

### Usage

```
crf.gp.weibull(x, t = 0, alpha, theta)
hgp.weibull(x, alpha, theta)
hra.gp.weibull(x, alpha, theta)
sgp.weibull(x, alpha, theta)
```

### Arguments

x	vector of quantiles.
alpha	shape parameter.
theta	shape parameter.
t	age component.

### Value

crf.gp.weibull gives the conditional reliability function (crf), hgp.weibull gives the hazard function, hra.gp.weibull gives the hazard rate average (HRA) function, and sgp.weibull gives the survival function for the generalized power Weibull(GPW) distribution.

### References

Nikulin, M. and Haghghi, F.(2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

Pham, H. and Lai, C.D.(2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

### See Also

[dgp.weibull](#) for other generalized power Weibull(GPW) distribution related functions;

**Examples**

```

## load data set
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

## Reliability indicators for data(repairtimes):

## Reliability function
sgp.weibull(repairtimes, 1.566093, 0.355321)

## Hazard function
hgp.weibull(repairtimes, 1.566093, 0.355321)

## hazard rate average(hra)
hra.gp.weibull(repairtimes, 1.566093, 0.355321)

## Conditional reliability function (age component=0)
crf.gp.weibull(repairtimes, 0.00, 1.566093, 0.355321)

## Conditional reliability function (age component=3.0)
crf.gp.weibull(repairtimes, 3.0, 1.566093, 0.355321)

```

Gumbel

*The Gumbel distribution***Description**

Density, distribution function, quantile function and random generation for the Gumbel distribution with location parameter  $\mu$  and scale parameter  $\sigma$ .

**Usage**

```

dgumbel(x, mu, sigma, log = FALSE)
pgumbel(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)
qgumbel(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)
rgumbel(n, mu, sigma)

```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>mu</code>	location parameter.
<code>sigma</code>	scale parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$ .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Gumbel distribution has density

$$f(x) = \frac{1}{\sigma} \exp \left\{ - \left( \frac{x - \mu}{\sigma} \right) \right\} \exp \left[ - \exp \left\{ - \left( \frac{x - \mu}{\sigma} \right) \right\} \right]; -\infty < x < \infty, \sigma > 0.$$

where  $\mu$  and  $\sigma$  are the shape and scale parameters, respectively.

**Value**

dgumbel gives the density, pgumbel gives the distribution function, qgumbel gives the quantile function, and rgumbel generates random deviates.

**References**

Marshall, A. W., Olkin, I. (2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[.Random.seed](#) about random number; [sgumbel](#) for Gumbel survival / hazard etc. functions

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

dgumbel(dataset2, 212.157, 151.768, log = FALSE)
pgumbel(dataset2, 212.157, 151.768, lower.tail = TRUE, log.p = FALSE)
qgumbel(0.25, 212.157, 151.768, lower.tail=TRUE, log.p = FALSE)
rgumbel(30, 212.157, 151.768)
```

---

GumbelSurvival

*Survival related functions for the Gumbel distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Gumbel distribution with location parameter  $\mu$  and scale parameter  $\sigma$ .

**Usage**

```
crf.gumbel(x, t = 0, mu, sigma)
hgumbel(x, mu, sigma)
hra.gumbel(x, mu, sigma)
sgumbel(x, mu, sigma)
```

**Arguments**

x	vector of quantiles.
mu	location parameter.
sigma	scale parameter.
t	age component.

**Value**

crf.gumbel gives the conditional reliability function (crf), hgumbel gives the hazard function, hra.gumbel gives the hazard rate average (HRA) function, and sgumbel gives the survival function for the Gumbel distribution.

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[dgumbel](#) for other Gumbel distribution related functions;

**Examples**

```
## load data set
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

## Reliability indicators for data(dataset2):

## Reliability function
sgumbel(dataset2, 212.157, 151.768)

## Hazard function
hgumbel(dataset2, 212.157, 151.768)

## hazard rate average(hra)
hra.gumbel(dataset2, 212.157, 151.768)

## Conditional reliability function (age component=0)
crf.gumbel(dataset2, 0.00, 212.157, 151.768)

## Conditional reliability function (age component=3.0)
crf.gumbel(dataset2, 3.0, 212.157, 151.768)
```

**Description**

Density, distribution function, quantile function and random generation for the Inverse Generalized Exponential(IGE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
dinv.genexp(x, alpha, lambda, log = FALSE)
pinv.genexp(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qinv.genexp(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rinv.genexp(n, alpha, lambda)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Inverse Generalized Exponential(IGE) distribution has density

$$f(x; \alpha, \lambda) = \frac{\alpha \lambda}{x^2} e^{-\lambda/x} \left\{ 1 - e^{-\lambda/x} \right\}^{\alpha-1}; (\alpha, \lambda) > 0, x > 0$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dinv.genexp gives the density, pinv.genexp gives the distribution function, qinv.genexp gives the quantile function, and rinv.genexp generates random deviates.

**References**

- Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*, Biometrical Journal, 43(1), 117-130.
- Gupta, R.D. and Kundu, D. (2007). *Generalized exponential distribution: Existing results and some recent development*, Journal of Statistical Planning and Inference. 137, 3537-3547.

**See Also**

[.Random.seed](#) about random number; [sinv.genexp](#) for Inverse Generalized Exponential(IGE) survival / hazard etc. functions

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889
dinv.genexp(repairtimes, 1.097807, 1.206889, log = FALSE)
pinv.genexp(repairtimes, 1.097807, 1.206889, lower.tail = TRUE, log.p = FALSE)
qinv.genexp(0.25, 1.097807, 1.206889, lower.tail=TRUE, log.p = FALSE)
rinv.genexp(30, 1.097807, 1.206889)
```

---

InvGenExpEsurvival	<i>Survival related functions for the Inverse Generalized Exponential(IGE) distribution</i>
--------------------	---

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Inverse Generalized Exponential(IGE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
crf.inv.genexp(x, t = 0, alpha, lambda)
hinv.genexp(x, alpha, lambda)
hra.inv.genexp(x, alpha, lambda)
sinv.genexp(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

**Value**

`crf.inv.genexp` gives the conditional reliability function (crf), `hinv.genexp` gives the hazard function, `hra.inv.genexp` gives the hazard rate average (HRA) function, and `sinv.genexp` gives the survival function for the Inverse Generalized Exponential(IGE) distribution.

## References

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*, Biometrical Journal, 43(1), 117-130.

Gupta, R.D. and Kundu, D., (2007). *Generalized exponential distribution: Existing results and some recent development*, Journal of Statistical Planning and Inference. 137, 3537-3547.

## See Also

[dinv.genexp](#) for other Inverse Generalized Exponential(IGE) distribution related functions;

## Examples

```
## load data set
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

## Reliability indicators for data(repairtimes):

## Reliability function
sinv.genexp(repairtimes, 1.097807, 1.206889)

## Hazard function
hinvg.genexp(repairtimes, 1.097807, 1.206889)

## hazard rate average(hra)
hra.invg.genexp(repairtimes, 1.097807, 1.206889)

## Conditional reliability function (age component=0)
crf.invg.genexp(repairtimes, 0.00, 1.097807, 1.206889)

## Conditional reliability function (age component=3.0)
crf.invg.genexp(repairtimes, 3.0, 1.097807, 1.206889)
```

---

ks.burrX

*Test of Kolmogorov-Smirnov for the BurrX distribution*


---

## Description

The function `ks.burrX()` gives the values for the KS test assuming a BurrX with shape parameter  $\alpha$  and scale parameter  $\lambda$ . In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

## Usage

```
ks.burrX(x, alpha.est, lambda.est,
         alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

### Arguments

x	vector of observations.
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

### Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

### Value

The function `ks.burrX()` carries out the KS test for the BurrX

### References

Kundu, D., and Raqab, M.Z. (2005). *Generalized Rayleigh Distribution: Different Methods of Estimation*, Computational Statistics and Data Analysis, 49, 187-200.

Surles, J.G., and Padgett, W.J. (2005). *Some properties of a scaled Burr type X distribution*, Journal of Statistical Planning and Inference, 128, 271-280.

Raqab, M.Z., and Kundu, D. (2006). *Burr Type X distribution: revisited*, Journal of Probability and Statistical Sciences, 4(2), 179-193.

### See Also

[pp.burrX](#) for PP plot and [qq.burrX](#) for QQ plot

### Examples

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

ks.burrX(bearings, 1.1989515, 0.0130847, alternative = "two.sided", plot = TRUE)
```

---

`ks.chen`*Test of Kolmogorov-Smirnov for the Chen distribution*

---

### Description

The function `ks.chen()` gives the values for the KS test assuming the Chen distribution with shape parameter `beta` and scale parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

### Usage

```
ks.chen(x, beta.est, lambda.est,  
        alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

### Arguments

<code>x</code>	vector of observations.
<code>beta.est</code>	estimate of the parameter <code>beta</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

### Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

### Value

The function `ks.chen()` carries out the KS test for the Chen.

### References

Castillo, E., Hadi, A.S., Balakrishnan, N. and Sarabia, J.M.(2004). *Extreme Value and Related Models with Applications in Engineering and Science*, John Wiley and Sons, New York.

Chen, Z.(2000). *A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function*, Statistics and Probability Letters, 49, 155-161.

Pham, H. (2003). *Handbook of Reliability Engineering*, Springer-Verlag.

### See Also

[pp.chen](#) for PP plot and [qq.chen](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Estimates of beta & lambda using 'maxLik' package
## beta.est = 0.262282404, lambda.est = 0.007282371

ks.chen(sys2, 0.262282404, 0.007282371, alternative = "two.sided", plot = TRUE)
```

---

ks.exp.ext	<i>Test of Kolmogorov-Smirnov for the Exponential Extension(EE) distribution</i>
------------	--

---

**Description**

The function `ks.exp.ext()` gives the values for the KS test assuming a Exponential Extension(EE) with shape parameter `alpha` and scale parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.exp.ext(x, alpha.est, lambda.est,
           alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.exp.ext()` carries out the KS test for the Exponential Extension(EE)

**References**

Nikulin, M. and Haghghi, F. (2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

**See Also**

[pp.exp.ext](#) for PP plot and [qq.exp.ext](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

ks.exp.ext(sys2, 1.0126e+01, 1.5848e-04, alternative = "two.sided", plot = TRUE)
```

---

ks.exp.power	<i>Test of Kolmogorov-Smirnov for the Exponential Power(EP) distribution</i>
--------------	--

---

**Description**

The function `ks.exp.power()` gives the values for the KS test assuming an Exponential Power distribution with shape parameter `alpha` and scale parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.exp.power(x, alpha.est, lambda.est,
             alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.exp.power()` carries out the KS test for the EP.

**References**

Smith, R.M. and Bain, L.J. (1975). *An exponential power life-test distribution*, Communications in Statistics - Simulation and Computation, Vol. 4(5), 469-481.

**See Also**

[pp.exp.power](#) for PP plot and [qq.exp.power](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

ks.exp.power(sys2, 0.905868898, 0.001531423, alternative = "two.sided", plot = TRUE)
```

---

ks.expo.logistic	<i>Test of Kolmogorov-Smirnov for the Exponentiated Logistic (EL) distribution</i>
------------------	--

---

**Description**

The function `ks.expo.logistic()` gives the values for the KS test assuming a Exponentiated Logistic(EL) with shape parameter alpha and scale parameter beta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.expo.logistic(x, alpha.est, beta.est,
  alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.expo.logistic()` carries out the KS test for the Exponentiated Logistic(EL)

**References**

Ali, M.M., Pal, M. and Woo, J. (2007). *Some Exponentiated Distributions*, The Korean Communications in Statistics, 14(1), 93-109.

Shirke, D.T., Kumbhar, R.R. and Kundu, D. (2005). *Tolerance intervals for exponentiated scale family of distributions*, Journal of Applied Statistics, 32, 1067-1074

**See Also**

[pp.expo.logistic](#) for PP plot and [qq.expo.logistic](#) for QQ plot

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

ks.expo.logistic(dataset2, 5.31302, 139.04515, alternative = "two.sided", plot = TRUE)
```

---

ks.expo.weibull	<i>Test of Kolmogorov-Smirnov for the Exponentiated Weibull(EW) distribution</i>
-----------------	--

---

**Description**

The function `ks.expo.weibull()` gives the values for the KS test assuming a Exponentiated Weibull(EW) with shape parameter alpha and scale parameter theta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.expo.weibull(x, alpha.est, theta.est,
  alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
alpha.est	estimate of the parameter alpha
theta.est	estimate of the parameter theta
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.expo.weibull()` carries out the KS test for the Exponentiated Weibull(EW)

**References**

Mudholkar, G.S. and Srivastava, D.K. (1993). *Exponentiated Weibull family for analyzing bathtub failure-rate data*, IEEE Transactions on Reliability, 42(2), 299-302.

Murthy, D.N.P., Xie, M. and Jiang, R. (2003). *Weibull Models*, Wiley, New York.

Nassar, M.M., and Eissa, F. H. (2003). *On the Exponentiated Weibull Distribution*, Communications in Statistics - Theory and Methods, 32(7), 1317-1336.

**See Also**

[pp.expo.weibull](#) for PP plot and [qq.expo.weibull](#) for QQ plot

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est =1.026465, theta.est = 7.824943

ks.expo.weibull(stress, 1.026465, 7.824943, alternative = "two.sided", plot = TRUE)
```

---

ks.flex.weibull	<i>Test of Kolmogorov-Smirnov for the flexible Weibull(FW) distribution</i>
-----------------	---

---

### Description

The function `ks.flex.weibull()` gives the values for the KS test assuming a flexible Weibull(FW) with shape parameter `alpha` and scale parameter `beta`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

### Usage

```
ks.flex.weibull(x, alpha.est, beta.est,  
               alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

### Arguments

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>beta.est</code>	estimate of the parameter <code>beta</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

### Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

### Value

The function `ks.flex.weibull()` carries out the KS test for the flexible Weibull(FW)

### References

Bebbington, M., Lai, C.D. and Zitikis, R. (2007). *A flexible Weibull extension*, Reliability Engineering and System Safety, 92, 719-726.

### See Also

[pp.flex.weibull](#) for PP plot and [qq.flex.weibull](#) for QQ plot

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

ks.flex.weibull(repairtimes, 0.07077507, 1.13181535,
  alternative = "two.sided", plot = TRUE)
```

---

ks.gen.exp	<i>Test of Kolmogorov-Smirnov for the Generalized Exponential(GE) distribution</i>
------------	--

---

**Description**

The function `ks.gen.exp()` gives the values for the KS test assuming an GE with shape parameter  $\alpha$  and scale parameter  $\lambda$ . In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.gen.exp(x, alpha.est, lambda.est,
  alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter $\alpha$
<code>lambda.est</code>	estimate of the parameter $\lambda$
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.gen.exp()` carries out the KS test for the GE.

## References

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*. Biometrical Journal, 43(1), 117 - 130.

Gupta, R. D. and Kundu, D. (1999). *Generalized exponential distributions*. Australian and New Zealand Journal of Statistics, 41(2), 173 - 188.

## See Also

[pp.gen.exp](#) for PP plot and [qq.gen.exp](#) for QQ plot

## Examples

```
## Load data sets
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609
ks.gen.exp(bearings, 5.28321139, 0.03229609, alternative = "two.sided", plot = TRUE)
```

---

ks.gompertz

*Test of Kolmogorov-Smirnov for the Gompertz distribution*

---

## Description

The function `ks.gompertz()` gives the values for the KS test assuming a Gompertz with shape parameter  $\alpha$  and scale parameter  $\theta$ . In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

## Usage

```
ks.gompertz(x, alpha.est, theta.est,
            alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter $\alpha$
<code>theta.est</code>	estimate of the parameter $\theta$
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.gompertz()` carries out the KS test for the Gompertz

**References**

Marshall, A. W., Olkin, I. (2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[pp.gompertz](#) for PP plot and [qq.gompertz](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

ks.gompertz(sys2, 0.00121307, 0.00173329, alternative = "two.sided", plot = TRUE)
```

---

ks.gp.weibull	<i>Test of Kolmogorov-Smirnov for the generalized power Weibull(GPW) distribution</i>
---------------	---

---

**Description**

The function `ks.gp.weibull()` gives the values for the KS test assuming a generalized power Weibull(GPW) with shape parameter  $\alpha$  and scale parameter  $\theta$ . In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.gp.weibull(x, alpha.est, theta.est,
             alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
alpha.est	estimate of the parameter $\alpha$
theta.est	estimate of the parameter $\theta$
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

## Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

## Value

The function `ks.gp.weibull()` carries out the KS test for the generalized power Weibull(GPW)

## References

Nikulin, M. and Haghighi, F. (2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

Pham, H. and Lai, C.D. (2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

## See Also

[pp.gp.weibull](#) for PP plot and [qq.gp.weibull](#) for QQ plot

## Examples

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

ks.gp.weibull(repairtimes, 1.566093, 0.355321, alternative = "two.sided", plot = TRUE)
```

---

ks.gumbel

*Test of Kolmogorov-Smirnov for the Gumbel distribution*

---

## Description

The function `ks.gumbel()` gives the values for the KS test assuming a Gumbel with shape parameter  $\mu$  and scale parameter  $\sigma$ . In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

## Usage

```
ks.gumbel(x, mu.est, sigma.est,
          alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
mu.est	estimate of the parameter mu
sigma.est	estimate of the parameter sigma
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.gumbel()` carries out the KS test for the Gumbel

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[pp.gumbel](#) for PP plot and [qq.gumbel](#) for QQ plot

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

ks.gumbel(dataset2, 212.157, 151.768, alternative = "two.sided", plot = TRUE)
```

---

ks.inv.genexp	<i>Test of Kolmogorov-Smirnov for the Inverse Generalized Exponential(IGE) distribution</i>
---------------	---

---

**Description**

The function `ks.inv.genexp()` gives the values for the KS test assuming a Inverse Generalized Exponential(IGE) with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.inv.genexp(x, alpha.est, lambda.est,
             alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.inv.genexp()` carries out the KS test for the Inverse Generalized Exponential (IGE)

**References**

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*, Biometrical Journal, 43(1), 117-130.

Gupta, R.D. and Kundu, D. (2007). *Generalized exponential distribution: Existing results and some recent development*, Journal of Statistical Planning and Inference. 137, 3537-3547.

**See Also**

[pp.inv.genexp](#) for PP plot and [qq.inv.genexp](#) for QQ plot

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood (ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

ks.inv.genexp(repairtimes, 1.097807, 1.206889, alternative = "two.sided", plot = TRUE)
```

---

ks.lfr	<i>Test of Kolmogorov-Smirnov for the linear failure rate(LFR) distribution</i>
--------	---

---

### Description

The function `ks.lfr()` gives the values for the KS test assuming a linear failure rate(LFR) with shape parameter alpha and scale parameter beta. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

### Usage

```
ks.lfr(x, alpha.est, beta.est,
       alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

### Arguments

x	vector of observations.
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

### Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

### Value

The function `ks.lfr()` carries out the KS test for the linear failure rate(LFR)

### References

- Bain, L.J. (1974). *Analysis for the Linear Failure-Rate Life-Testing Distribution*, Technometrics, 16(4), 551 - 559.
- Lawless, J.F. (2003). *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York.
- Sen, A. and Bhattacharya, G.K. (1995). *Inference procedure for the linear failure rate mode*, Journal of Statistical Planning and Inference, 46, 59-76.

### See Also

[pp.lfr](#) for PP plot and [qq.lfr](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

ks.lfr(sys2, 1.777673e-03, 2.777640e-06, alternative = "two.sided", plot = TRUE)
```

---

ks.log.gamma

*Test of Kolmogorov-Smirnov for the log-gamma(LG) distribution*


---

**Description**

The function `ks.log.gamma()` gives the values for the KS test assuming a log-gamma(LG) with shape parameter `alpha` and scale parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.log.gamma(x, alpha.est, lambda.est,
             alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.log.gamma()` carries out the KS test for the log-gamma(LG)

**References**

Klugman, S., Panjer, H. and Willmot, G. (2004). *Loss Models: From Data to Decisions*, 2nd ed., New York, Wiley.

Lawless, J. F., (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

**See Also**

[pp.log.gamma](#) for PP plot and [qq.log.gamma](#) for QQ plot

**Examples**

```
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

ks.log.gamma(conductors, 0.0088741, 0.6059935, alternative = "two.sided", plot = TRUE)
```

---

ks.logis.exp	<i>Test of Kolmogorov-Smirnov for the Logistic-Exponential(LE) distribution</i>
--------------	---

---

**Description**

The function `ks.logis.exp()` gives the values for the KS test assuming a Logistic-Exponential(LE) with shape parameter `alpha` and scale parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.logis.exp(x, alpha.est, lambda.est,
             alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.logis.exp()` carries out the KS test for the Logistic-Exponential(LE)

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[pp.logis.exp](#) for PP plot and [qq.logis.exp](#) for QQ plot

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

ks.logis.exp(bearings, 2.36754, 0.01059, alternative = "two.sided", plot = TRUE)
```

---

ks.logis.rayleigh	<i>Test of Kolmogorov-Smirnov for the Logistic-Rayleigh(LR) distribution</i>
-------------------	--

---

**Description**

The function `ks.logis.rayleigh()` gives the values for the KS test assuming a Logistic-Rayleigh(LR) with shape parameter `alpha` and scale parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.logis.rayleigh(x, alpha.est, lambda.est,
  alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.logis.rayleigh()` carries out the KS test for the Logistic-Rayleigh(LR)

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[pp.logis.rayleigh](#) for PP plot and [qq.logis.rayleigh](#) for QQ plot

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

ks.logis.rayleigh(stress, 1.4779388, 0.2141343,
  alternative = "two.sided", plot = TRUE)
```

---

 ks.loglog

---

*Test of Kolmogorov-Smirnov for the Loglog distribution*


---

**Description**

The function `ks.loglog()` gives the values for the KS test assuming the Loglog distribution with shape parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.loglog(x, alpha.est, lambda.est,
  alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.loglog()` carries out the KS test for the Loglog.

**References**

Pham, H.(2002). *A Vtub-Shaped Hazard Rate Function with Applications to System Safety*, International Journal of Reliability and Applications, Vol. 3, No. 1, pp. 1-16.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

**See Also**

[pp.loglog](#) for PP plot and [qq.loglog](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

ks.loglog(sys2, 0.9058689, 1.0028228, alternative = "two.sided", plot = TRUE)
```

---

ks.moee

*Test of Kolmogorov-Smirnov for the Marshall-Olkin Extended Exponential(MOEE) distribution*

---

**Description**

The function `ks.moee()` gives the values for the KS test assuming an GE with tilt parameter alpha and scale parameter lambda. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.moee(x, alpha.est, lambda.est,  
        alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

x	vector of observations.
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
alternative	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
plot	Logical; if TRUE, the cdf plot is provided.
...	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.moee()` carries out the KS test for the MOEE

**References**

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

**See Also**

[pp.moee](#) for PP plot and [qq.moee](#) for QQ plot

**Examples**

```
## Load dataset  
data(stress)  
## Estimates of alpha & lambda using 'maxLik' package  
## alpha.est = 75.67982, lambda.est = 1.67576  
  
ks.moee(stress, 75.67982, 1.67576, alternative = "two.sided", plot = TRUE)
```

---

ks.moew	<i>Test of Kolmogorov-Smirnov for the Marshall-Olkin Extended Exponential(MOEW) distribution</i>
---------	--

---

## Description

The function `ks.moew()` gives the values for the KS test assuming a MOEW with shape parameter `alpha` and tilt parameter `lambda`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

## Usage

```
ks.moew(x, alpha.est, lambda.est,  
        alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>lambda.est</code>	estimate of the parameter <code>lambda</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Details

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

## Value

The function `ks.moew()` carries out the KS test for the MOEW

## References

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the Weibull and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I. (2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

## See Also

[pp.moew](#) for PP plot and [qq.moew](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

ks.moew(sys2, 0.3035937, 279.2177754, alternative = "two.sided", plot = TRUE)
```

---

ks.weibull.ext	<i>Test of Kolmogorov-Smirnov for the Weibull Extension(WE) distribution</i>
----------------	--

---

**Description**

The function `ks.weibull.ext()` gives the values for the KS test assuming a Weibull Extension(WE) with shape parameter `alpha` and scale parameter `beta`. In addition, optionally, this function allows one to show a comparative graph between the empirical and theoretical cdfs for a specified data set.

**Usage**

```
ks.weibull.ext(x, alpha.est, beta.est,
               alternative = c("less", "two.sided", "greater"), plot = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations.
<code>alpha.est</code>	estimate of the parameter <code>alpha</code>
<code>beta.est</code>	estimate of the parameter <code>beta</code>
<code>alternative</code>	indicates the alternative hypothesis and must be one of "two.sided" (default), "less", or "greater".
<code>plot</code>	Logical; if TRUE, the cdf plot is provided.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Details**

The Kolmogorov-Smirnov test is a goodness-of-fit technique based on the maximum distance between the empirical and theoretical cdfs.

**Value**

The function `ks.weibull.ext()` carries out the KS test for the Weibull Extension(WE)

**References**

Tang, Y., Xie, M. and Goh, T.N., (2003). *Statistical analysis of a Weibull extension model*, Communications in Statistics: Theory & Methods 32(5):913-928.

Zhang, T., and Xie, M.(2007). *Failure Data Analysis with Extended Weibull Distribution*, Communications in Statistics-Simulation and Computation, 36(3), 579-592.

**See Also**

[pp.weibull.ext](#) for PP plot and [qq.weibull.ext](#) for QQ plot

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

ks.weibull.ext(sys2, 0.00019114, 0.14696242, alternative = "two.sided", plot = TRUE)
```

LFR

*The linear failure rate(LFR) distribution***Description**

Density, distribution function, quantile function and random generation for the linear failure rate(LFR) distribution with parameters alpha and beta.

**Usage**

```
dlfr(x, alpha, beta, log = FALSE)
plfr(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qlfr(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rlfr(n, alpha, beta)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	parameter.
beta	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The linear failure rate(LFR) distribution has density

$$f(x) = (\alpha + \beta x) \exp \left\{ - \left( \alpha x + \frac{\beta x^2}{2} \right) \right\}; x \geq 0, \alpha > 0, \beta > 0.$$

where  $\alpha$  and  $\beta$  are the shape and scale parameters, respectively.

**Value**

dlfr gives the density, plfr gives the distribution function, qlfr gives the quantile function, and r1fr generates random deviates.

**References**

- Bain, L.J. (1974). *Analysis for the Linear Failure-Rate Life-Testing Distribution*, Technometrics, 16(4), 551 - 559.
- Lawless, J.F.(2003). *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York.
- Sen, A. and Bhattacharya, G.K.(1995). *Inference procedure for the linear failure rate mode*, Journal of Statistical Planning and Inference, 46, 59-76.

**See Also**

[.Random.seed](#) about random number; [slfr](#) for linear failure rate(LFR) survival / hazard etc. functions

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

dlfr(sys2, 1.777673e-03, 2.777640e-06, log = FALSE)
plfr(sys2, 1.777673e-03, 2.777640e-06, lower.tail = TRUE, log.p = FALSE)
qlfr(0.25, 1.777673e-03, 2.777640e-06, lower.tail=TRUE, log.p = FALSE)
r1fr(30, 1.777673e-03, 2.777640e-06)
```

---

LFRsurvival

*Survival related functions for the linear failure rate(LFR) distribution*


---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the linear failure rate(LFR) distribution with parameters alpha and beta.

**Usage**

```
crf.lfr(x, t = 0, alpha, beta)
hlfr(x, alpha, beta)
hra.lfr(x, alpha, beta)
slfr(x, alpha, beta)
```

**Arguments**

x	vector of quantiles.
alpha	parameter.
beta	parameter.
t	age component.

**Value**

crf.lfr gives the conditional reliability function (crf), hlfr gives the hazard function, hra.lfr gives the hazard rate average (HRA) function, and slfr gives the survival function for the linear failure rate(LFR) distribution.

**References**

- Bain, L.J. (1974). *Analysis for the Linear Failure-Rate Life-Testing Distribution*, Technometrics, 16(4), 551 - 559.
- Lawless, J.F.(2003). *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York.
- Sen, A. and Bhattacharya, G.K.(1995). *Inference procedure for the linear failure rate mode*, Journal of Statistical Planning and Inference, 46, 59-76.

**See Also**

[dlfr](#) for other linear failure rate(LFR) distribution related functions;

**Examples**

```
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

## Reliability indicators for data(sys2):

## Reliability function
slfr(sys2, 1.777673e-03, 2.777640e-06)

## Hazard function
hlfr(sys2, 1.777673e-03, 2.777640e-06)

## hazard rate average(hra)
hra.lfr(sys2, 1.777673e-03, 2.777640e-06)

## Conditional reliability function (age component=0)
crf.lfr(sys2, 0.00, 1.777673e-03, 2.777640e-06)

## Conditional reliability function (age component=3.0)
crf.lfr(sys2, 3.0, 1.777673e-03, 2.777640e-06)
```

Loggamma

*The log-gamma(LG) distribution***Description**

Density, distribution function, quantile function and random generation for the log-gamma(LG) distribution with parameters alpha and lambda.

**Usage**

```
dlog.gamma(x, alpha, lambda, log = FALSE)
plog.gamma(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qlog.gamma(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rlog.gamma(n, alpha, lambda)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	parameter.
lambda	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The log-gamma(LG) distribution has density

$$f(x; \alpha, \lambda) = \alpha \lambda \exp\{\lambda x\} \exp\{-\alpha \exp \lambda x\}; (\alpha, \lambda) > 0, x > 0$$

where  $\alpha$  and  $\lambda$  are the parameters, respectively.

**Value**

dlog.gamma gives the density, plog.gamma gives the distribution function, qlog.gamma gives the quantile function, and rlog.gamma generates random deviates.

**References**

Klugman, S., Panjer, H. and Willmot, G. (2004). *Loss Models: From Data to Decisions*, 2nd ed., New York, Wiley.

Lawless, J. F., (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

**See Also**

[.Random.seed](#) about random number; [slog.gamma](#) for ExpExt survival / hazard etc. functions

**Examples**

```
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935
dlog.gamma(conductors, 0.0088741, 0.6059935, log = FALSE)
plog.gamma(conductors, 0.0088741, 0.6059935, lower.tail = TRUE, log.p = FALSE)
qlog.gamma(0.25, 0.0088741, 0.6059935, lower.tail=TRUE, log.p = FALSE)
rlog.gamma(30, 0.0088741, 0.6059935)
```

---

Loggammasurvival

*Survival related functions for the log-gamma(LG) distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the log-gamma(LG) distribution with shape parameters alpha and lambda.

**Usage**

```
crf.log.gamma(x, t = 0, alpha, lambda)
hlog.gamma(x, alpha, lambda)
hra.log.gamma(x, alpha, lambda)
slog.gamma(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	parameter.
lambda	parameter.
t	age component.

**Value**

crf.log.gamma gives the conditional reliability function (crf), hlog.gamma gives the hazard function, hra.log.gamma gives the hazard rate average (HRA) function, and slog.gamma gives the survival function for the log-gamma(LG) distribution.

**References**

- Klugman, S., Panjer, H. and Willmot, G. (2004). *Loss Models: From Data to Decisions*, 2nd ed., New York, Wiley.
- Lawless, J. F., (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

**See Also**

[dlog.gamma](#) for other log-gamma(LG) distribution related functions;

**Examples**

```
## load data set
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

## Reliability indicators for data(conductors):

## Reliability function
slog.gamma(conductors, 0.0088741, 0.6059935)

## Hazard function
hlog.gamma(conductors, 0.0088741, 0.6059935)

## hazard rate average(hra)
hra.log.gamma(conductors, 0.0088741, 0.6059935)

## Conditional reliability function (age component=0)
crf.log.gamma(conductors, 0.00, 0.0088741, 0.6059935)

## Conditional reliability function (age component=3.0)
crf.log.gamma(conductors, 3.0, 0.0088741, 0.6059935)
```

---

 LogisExp

*The Logistic-Exponential(LE) distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the Logistic-Exponential(LE) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
dlogis.exp(x, alpha, lambda, log = FALSE)
plogis.exp(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qlogis.exp(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rlogis.exp(n, alpha, lambda)
```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>alpha</code>	shape parameter.
<code>lambda</code>	scale parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Logistic-Exponential(LE) distribution has density

$$f(x) = \frac{\lambda \alpha e^{\lambda x} (e^{\lambda x} - 1)^{\alpha-1}}{\{1 + (e^{\lambda x} - 1)^\alpha\}^2}; x \geq 0, \alpha > 0, \lambda > 0.$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

`dlogis.exp` gives the density, `plogis.exp` gives the distribution function, `qlogis.exp` gives the quantile function, and `rlogis.exp` generates random deviates.

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[.Random.seed](#) about random number; [slogis.exp](#) for ExpExt survival / hazard etc. functions

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059
dlogis.exp(bearings, 2.36754, 0.01059, log = FALSE)
plogis.exp(bearings, 2.36754, 0.01059, lower.tail = TRUE, log.p = FALSE)
qlogis.exp(0.25, 2.36754, 0.01059, lower.tail=TRUE, log.p = FALSE)
rlogis.exp(30, 2.36754, 0.01059)
```

---

LogisExpsurvival	<i>Survival related functions for the Logistic-Exponential(LE) distribution</i>
------------------	---

---

### Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Logistic-Exponential(LE) distribution with shape parameter alpha and scale parameter lambda.

### Usage

```
crf.logis.exp(x, t = 0, alpha, lambda)
hlogis.exp(x, alpha, lambda)
hra.logis.exp(x, alpha, lambda)
slogis.exp(x, alpha, lambda)
```

### Arguments

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

### Value

crf.logis.exp gives the conditional reliability function (crf), hlogis.exp gives the hazard function, hra.logis.exp gives the hazard rate average (HRA) function, and slogis.exp gives the survival function for the Logistic-Exponential(LE) distribution.

### References

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

### See Also

[dlogis.exp](#) for other Logistic-Exponential(LE) distribution related functions;

### Examples

```
## load data set
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

## Reliability indicators for data(bearings):
```

```
## Reliability function
slogis.exp(bearings, 2.36754, 0.01059)

## Hazard function
hlogis.exp(bearings, 2.36754, 0.01059)

## hazard rate average(hra)
hra.logis.exp(bearings, 2.36754, 0.01059)

## Conditional reliability function (age component=0)
crf.logis.exp(bearings, 0.00, 2.36754, 0.01059)

## Conditional reliability function (age component=3.0)
crf.logis.exp(bearings, 3.0, 2.36754, 0.01059)
```

---

LogisRayleigh

*The Logistic-Rayleigh(LR) distribution*


---

### Description

Density, distribution function, quantile function and random generation for the Logistic-Rayleigh(LR) distribution with shape parameter alpha and scale parameter lambda.

### Usage

```
dlogis.rayleigh(x, alpha, lambda, log = FALSE)
plogis.rayleigh(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qlogis.rayleigh(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rlogis.rayleigh(n, alpha, lambda)
```

### Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

### Details

The cumulative distribution function(*cdf*) of Logistic-Rayleigh(LR) is given by

$$F(x) = 1 - \frac{1}{1 + (e^{(\lambda x^2/2)} - 1)^\alpha}; x \geq 0, \alpha > 0, \lambda > 0.$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively.

**Value**

dlogis.rayleigh gives the density, plogis.rayleigh gives the distribution function, qllogis.rayleigh gives the quantile function, and rlogis.rayleigh generates random deviates.

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[.Random.seed](#) about random number; [slogis.rayleigh](#) for ExpExt survival / hazard etc. functions

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343
dlogis.rayleigh(stress, 1.4779388, 0.2141343, log = FALSE)
plogis.rayleigh(stress, 1.4779388, 0.2141343, lower.tail = TRUE, log.p = FALSE)
qllogis.rayleigh(0.25, 1.4779388, 0.2141343, lower.tail=TRUE, log.p = FALSE)
rlogis.rayleigh(30, 1.4779388, 0.2141343)
```

---

LogisRayleighsurvival *Survival related functions for the Logistic-Rayleigh(LR) distribution*

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Logistic-Rayleigh(LR) distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
crf.logis.rayleigh(x, t = 0, alpha, lambda)
hlogis.rayleigh(x, alpha, lambda)
hra.logis.rayleigh(x, alpha, lambda)
slogis.rayleigh(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

**Value**

`crf.logis.rayleigh` gives the conditional reliability function (crf), `hlogis.rayleigh` gives the hazard function, `hra.logis.rayleigh` gives the hazard rate average (HRA) function, and `slogis.rayleigh` gives the survival function for the Logistic-Rayleigh(LR) distribution.

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[dlogis.rayleigh](#) for other Logistic-Rayleigh(LR) distribution related functions;

**Examples**

```
## load data set
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

## Reliability indicators for data(stress):

## Reliability function
slogis.rayleigh(stress, 1.4779388, 0.2141343)

## Hazard function
hlogis.rayleigh(stress, 1.4779388, 0.2141343)

## hazard rate average(hra)
hra.logis.rayleigh(stress, 1.4779388, 0.2141343)

## Conditional reliability function (age component=0)
crf.logis.rayleigh(stress, 0.00, 1.4779388, 0.2141343)

## Conditional reliability function (age component=3.0)
crf.logis.rayleigh(stress, 3.0, 1.4779388, 0.2141343)
```

**Description**

Density, distribution function, quantile function and random generation for the Loglog distribution with shape parameter alpha and scale parameter lambda.

**Usage**

```
dloglog(x, alpha, lambda, log = FALSE)
ploglog(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qloglog(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rloglog(n, alpha, lambda)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The loglog(Pham) distribution has density

$$f(x) = \alpha \ln(\lambda) x^{\alpha-1} \lambda^{x^\alpha} \exp\{1 - \lambda^{x^\alpha}\}; x > 0, \lambda > 0, \alpha > 0$$

where  $\alpha$  and  $\lambda$  are the shape and scale parameters, respectively. (Pham, 2002)

**Value**

dloglog gives the density, ploglog gives the distribution function, qloglog gives the quantile function, and rloglog generates random deviates.

**References**

Pham, H.(2002). *A Vtub-Shaped Hazard Rate Function with Applications to System Safety*, International Journal of Reliability and Applications. ,Vol. 3, No. 1, pp. 1-16.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

**See Also**

[.Random.seed](#) about random number; [sloglog](#) for Loglog survival / hazard etc. functions;

**Examples**

```
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

dloglog(sys2, 0.9058689, 1.0028228, log = FALSE)
```

```
ploglog(sys2, 0.9058689, 1.0028228, lower.tail = TRUE, log.p = FALSE)
qloglog(0.25, 0.9058689, 1.0028228, lower.tail=TRUE, log.p = FALSE)
rloglog(30, 0.9058689, 1.0028228)
```

---

Loglogsurvival

*Survival related functions for the Loglog distribution*

---

## Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Loglog distribution with shape parameter alpha and scale parameter lambda.

## Usage

```
crf.loglog(x, t = 0, alpha, lambda)
hloglog(x, alpha, lambda)
hra.loglog(x, alpha, lambda)
sloglog(x, alpha, lambda)
```

## Arguments

x	vector of quantiles.
alpha	shape parameter.
lambda	scale parameter.
t	age component.

## Value

crf.loglog gives the conditional reliability function (crf), hloglog gives the hazard function, hra.loglog gives the hazard rate average (HRA) function, and sloglog gives the survival function for the Loglog distribution.

## References

Pham, H.(2002). *A Vtub-Shaped Hazard Rate Function with Applications to System Safety*, International Journal of Reliability and Applications. ,Vol. 3, No. 1, pp. 1-16.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

## See Also

[dloglog](#) for other Loglog(Pham) distribution related functions;

**Examples**

```

## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

## Reliability indicators for data(sys2):

## Reliability function
sloglog(sys2, 0.9058689, 1.0028228)

## Hazard function
hloglog(sys2, 0.9058689, 1.0028228)

## hazard rate average(hra)
hra.loglog(sys2, 0.9058689, 1.0028228)

## Conditional reliability function (age component=0)
crf.loglog(sys2, 0.00, 0.9058689, 1.0028228)

## Conditional reliability function (age component=3.0)
crf.loglog(sys2, 3.0, 0.9058689, 1.0028228)

```

---

MOEE

---

*The Marshall-Olkin Extended Exponential (MOEE) distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the Marshall-Olkin Extended Exponential (MOEE) distribution with tilt parameter alpha and scale parameter lambda.

**Usage**

```

dmoe(x, alpha, lambda, log = FALSE)
pmoe(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qmoe(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rmoe(n, alpha, lambda)

```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	tilt parameter.
lambda	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

## Details

The Marshall-Olkin extended exponential (MOEE) distribution has density

$$f(x; \alpha, \lambda) = \frac{\alpha \lambda e^{-\lambda x}}{\{1 - (1 - \alpha)e^{-\lambda x}\}^2}; x > 0, \lambda > 0, \alpha > 0$$

where  $\alpha$  and  $\lambda$  are the tilt and scale parameters, respectively.

## Value

dmoe gives the density, pmoe gives the distribution function, qmoe gives the quantile function, and rmoe generates random deviates.

## References

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

## See Also

[.Random.seed](#) about random number; [smoe](#) for MOEE survival / hazard etc. functions

## Examples

```
## Load data sets
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576
dmoe(stress, 75.67982, 1.67576, log = FALSE)
pmoe(stress, 75.67982, 1.67576, lower.tail = TRUE,
     log.p = FALSE)
qmoe(0.25, 0.4, 2.0, lower.tail = TRUE, log.p = FALSE)
rmoe(10, 75.67982, 1.67576)
```

---

MOEEsurvival

*Survival related functions for the Marshall-Olkin Extended Exponential (MOEE) distribution*

---

## Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Marshall-Olkin Extended Exponential (MOEE) distribution with tilt parameter alpha and scale parameter lambda.

## Usage

```
crf.moee(x, t = 0, alpha, lambda)
hmoee(x, alpha, lambda)
hra.moee(x, alpha, lambda)
smoee(x, alpha, lambda)
```

## Arguments

x	vector of quantiles.
alpha	tilt parameter.
lambda	scale parameter.
t	age component.

## Value

`crf.moee` gives the conditional reliability function (crf), `hmoee` gives the hazard function, `hra.moee` gives the hazard rate average (HRA) function, and `smoee` gives the survival function for the MOEE distribution.

## References

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

## See Also

[dmoe](#) for other MOEE distribution related functions;

## Examples

```
## Load data sets
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576
smoee(stress, 75.67982, 1.67576)
hmoee(stress, 75.67982, 1.67576)
hra.moee(stress, 75.67982, 1.67576)
crf.moee(stress, 3.00, 75.67982, 1.67576)
```

**Description**

Density, distribution function, quantile function and random generation for the Marshall-Olkin Extended Weibull (MOEW) distribution with tilt parameter  $\alpha$  and scale parameter  $\lambda$ .

**Usage**

```
dmoew(x, alpha, lambda, log = FALSE)
pmoew(q, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
qmoew(p, alpha, lambda, lower.tail = TRUE, log.p = FALSE)
rmoew(n, alpha, lambda)
```

**Arguments**

<code>x, q</code>	vector of quantiles.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) &gt; 1</code> , the length is taken to be the number required.
<code>alpha</code>	shape parameter.
<code>lambda</code>	tilt parameter.
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as <code>log(p)</code> .
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Details**

The Marshall-Olkin extended Weibull (MOEW) distribution has density

$$f(x) = \frac{\lambda \alpha x^{\alpha-1} \exp(-x^\alpha)}{\{1 - (1 - \lambda) \exp(-x^\alpha)\}^2}; \quad x > 0, \lambda > 0, \alpha > 0$$

where  $\alpha$  and  $\lambda$  are the tilt and scale parameters, respectively.

**Value**

`dmoew` gives the density, `pmoew` gives the distribution function, `qmoew` gives the quantile function, and `rmoew` generates random deviates.

**References**

- Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the Weibull and Weibull families*. *Biometrika*, 84(3):641-652.
- Marshall, A. W., Olkin, I. (2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

**See Also**

[.Random.seed](#) about random number; [smoew](#) for MOEW survival / hazard etc. functions;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

dmoew(sys2, 0.3035937, 279.2177754, log = FALSE)
pmoew(sys2, 0.3035937, 279.2177754, lower.tail = TRUE, log.p = FALSE)
qmoew(0.25, 0.3035937, 279.2177754, lower.tail=TRUE, log.p = FALSE)
rmoew(50, 0.3035937, 279.2177754)
```

---

MOEWsurvival	<i>Survival related functions for the Marshall-Olkin Extended Weibull (MOEW) distribution</i>
--------------	---

---

**Description**

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Marshall-Olkin Extended Weibull (MOEW) distribution with tilt parameter alpha and scale parameter lambda.

**Usage**

```
crf.moew(x, t = 0, alpha, lambda)
hmoew(x, alpha, lambda)
hra.moew(x, alpha, lambda)
smoew(x, alpha, lambda)
```

**Arguments**

x	vector of quantiles.
alpha	tilt parameter.
lambda	scale parameter.
t	age component.

**Value**

`crf.moew` gives the conditional reliability function (crf), `hmoew` gives the hazard function, `hra.moew` gives the hazard rate average (HRA) function, and `smoew` gives the survival function for the MOEW distribution.

## References

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

## See Also

[dmoew](#) for other MOEW distribution related functions;

## Examples

```
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754
## Reliability indicators for data(sys2):

## Reliability function
smoew(sys2, 0.3035937, 279.2177754)

## Hazard function
hmoew(sys2, 0.3035937, 279.2177754)

## hazard rate average(hra)
hra.moew(sys2, 0.3035937, 279.2177754)

## Conditional reliability function (age component=0)
crf.moew(sys2, 0.00, 0.3035937, 279.2177754)

## Conditional reliability function (age component=3.0)
crf.moew(sys2, 3.0, 0.3035937, 279.2177754)
```

---

pp.burrX

*Probability versus Probability (PP) plot for the BurrX distribution*


---

## Description

The function `pp.burrX()` produces a PP plot for the BurrX based on their MLE or any other estimate. Also, a reference line can be sketched.

## Usage

```
pp.burrX(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.burrX()` carries out a PP plot for the BurrX.

**References**

Kundu, D., and Raqab, M.Z. (2005). *Generalized Rayleigh Distribution: Different Methods of Estimation*, Computational Statistics and Data Analysis, 49, 187-200.

Surles, J.G., and Padgett, W.J. (2005). *Some properties of a scaled Burr type X distribution*, Journal of Statistical Planning and Inference, 128, 271-280.

Raqab, M.Z., and Kundu, D. (2006). *Burr Type X distribution: revisited*, Journal of Probability and Statistical Sciences, 4(2), 179-193.

**See Also**

[qq.burrX](#) for QQ plot and [ks.burrX](#) function

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

pp.burrX(bearings, 1.1989515, 0.0130847, main = " ", line = TRUE)
```

---

pp.chen

*Probability versus Probability (PP) plot for the Chen distribution*

---

**Description**

The function `pp.chen()` produces a PP plot for the Chen based on their MLE or any other estimator. Also, a reference line can be sketched.

**Usage**

```
pp.chen(x, beta.est, lambda.est, main = " ", line = TRUE, ...)
```

**Arguments**

x	vector of observations
beta.est	estimate of the parameter beta
lambda.est	estimate of the parameter lambda
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.chen()` carries out a PP plot for the Chen.

**References**

Castillo, E., Hadi, A.S., Balakrishnan, N. and Sarabia, J.M.(2004). *Extreme Value and Related Models with Applications in Engineering and Science*, John Wiley and Sons, New York.

Chen, Z.(2000). *A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function*, Statistics and Probability Letters, 49, 155-161.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

**See Also**

[qq.chen](#) for QQ plot and [ks.chen](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

pp.chen(sys2, 0.262282404, 0.007282371, line = TRUE)
```

---

pp.exp.ext	<i>Probability versus Probability (PP) plot for the Exponential Extension(EE) distribution</i>
------------	--

---

**Description**

The function `pp.exp.ext()` produces a PP plot for the Exponential Extension(EE) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.exp.ext(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.exp.ext()` carries out a PP plot for the Exponential Extension(EE).

**References**

Nikulin, M. and Haghghi, F.(2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

**See Also**

[qq.exp.ext](#) for QQ plot and [ks.exp.ext](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

pp.exp.ext(sys2, 1.0126e+01, 1.5848e-04, main = " ", line = TRUE)
```

---

pp.exp.power	<i>Probability versus Probability (PP) plot for the Exponential Power distribution</i>
--------------	--

---

**Description**

The function `pp.exp.power()` produces a PP plot for the Exponential Power distribution based on their MLE or any other estimator. Also, a reference line can be sketched.

**Usage**

```
pp.exp.power(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.exp.power()` carries out a PP plot for the Exponential Power distribution.

**References**

Smith, R.M. and Bain, L.J.(1975). *An exponential power life-test distribution*, Communications in Statistics - Simulation and Computation, Vol.4(5), 469 - 481

**See Also**

[qq.exp.power](#) for QQ plot and [ks.exp.power](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

pp.exp.power(sys2, 0.905868898, 0.001531423, main = '', line = TRUE)
```

---

pp.expo.logistic	<i>Probability versus Probability (PP) plot for the Exponentiated Logistic(EL) distribution</i>
------------------	---

---

**Description**

The function `pp.expo.logistic()` produces a PP plot for the Exponentiated Logistic(EL) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.expo.logistic(x, alpha.est, beta.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.expo.logistic()` carries out a PP plot for the Exponentiated Logistic(EL).

**References**

Ali, M.M., Pal, M. and Woo, J. (2007). *Some Exponentiated Distributions*, The Korean Communications in Statistics, 14(1), 93-109.

Shirke, D.T., Kumbhar, R.R. and Kundu, D.(2005). *Tolerance intervals for exponentiated scale family of distributions*, Journal of Applied Statistics, 32, 1067-1074

**See Also**

[qq.expo.logistic](#) for QQ plot and [ks.expo.logistic](#) function;

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

pp.expo.logistic(dataset2, 5.31302, 139.04515, main = " ", line = TRUE)
```

---

pp.expo.weibull	<i>Probability versus Probability (PP) plot for the Exponentiated Weibull(EW) distribution</i>
-----------------	--

---

**Description**

The function `pp.expo.weibull()` produces a PP plot for the Exponentiated Weibull(EW) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.expo.weibull(x, alpha.est, theta.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
theta.est	estimate of the parameter theta
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.expo.weibull()` carries out a PP plot for the Exponentiated Weibull(EW).

**References**

- Mudholkar, G.S. and Srivastava, D.K. (1993). *Exponentiated Weibull family for analyzing bathtub failure-rate data*, IEEE Transactions on Reliability, 42(2), 299-302.
- Murthy, D.N.P., Xie, M. and Jiang, R. (2003). *Weibull Models*, Wiley, New York.
- Nassar, M.M., and Eissa, F. H. (2003). *On the Exponentiated Weibull Distribution*, Communications in Statistics - Theory and Methods, 32(7), 1317-1336.

**See Also**

[qq.expo.weibull](#) for QQ plot and [ks.expo.weibull](#) function;

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est =1.026465, theta.est = 7.824943

pp.expo.weibull(stress, 1.026465, 7.824943, main = " ", line = TRUE)
```

---

pp.flex.weibull	<i>Probability versus Probability (PP) plot for the flexible Weibull(FW) distribution</i>
-----------------	---

---

**Description**

The function `pp.flex.weibull()` produces a PP plot for the flexible Weibull(FW) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.flex.weibull(x, alpha.est, beta.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
beta.est	estimate of the parameter beta
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.flex.weibull()` carries out a PP plot for the flexible Weibull(FW).

**References**

Bebbington, M., Lai, C.D. and Zitikis, R. (2007). *A flexible Weibull extension*, Reliability Engineering and System Safety, 92, 719-726.

**See Also**

[qq.flex.weibull](#) for QQ plot and [ks.flex.weibull](#) function;

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

pp.flex.weibull(repairtimes, 0.07077507, 1.13181535, main = " ", line = TRUE)
```

---

pp.gen.exp

*Probability versus Probability (PP) plot for the Generalized Exponential(GE) distribution*

---

**Description**

The function `pp.gen.exp()` produces a PP plot for the GE based on their MLE or any other estimator. Also, a reference line can be sketched.

**Usage**

```
pp.gen.exp(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
lambda.est	estimate of the parameter lambda
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.gen.exp()` carries out a PP plot for the GE.

**See Also**

[qq.gen.exp](#) for QQ plot and [ks.gen.exp](#) functions;

**Examples**

```
## Load dataset
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609

pp.gen.exp(bearings, 5.28321139, 0.03229609, line = TRUE)
```

---

pp.gompertz	<i>Probability versus Probability (PP) plot for the Gompertz distribution</i>
-------------	---

---

**Description**

The function `pp.gompertz()` produces a PP plot for the Gompertz based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.gompertz(x, alpha.est, theta.est, main = " ", line = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
theta.est	estimate of the parameter theta
main	the title for the plot.
line	logical; if TRUE, a 45 degree line is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.gompertz()` carries out a PP plot for the Gompertz.

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[qq.gompertz](#) for QQ plot and [ks.gompertz](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

pp.gompertz(sys2, 0.00121307, 0.00173329, main = " ", line = TRUE)
```

---

<code>pp.gp.weibull</code>	<i>Probability versus Probability (PP) plot for the generalized power Weibull(GPW) distribution</i>
----------------------------	---

---

**Description**

The function `pp.gp.weibull()` produces a PP plot for the generalized power Weibull(GPW) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.gp.weibull(x, alpha.est, theta.est, main = " ", line = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>theta.est</code>	estimate of the parameter theta
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.gp.weibull()` carries out a PP plot for the generalized power Weibull(GPW).

## References

Nikulin, M. and Haghghi, F.(2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

Pham, H. and Lai, C.D.(2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

## See Also

[qq.gp.weibull](#) for QQ plot and [ks.gp.weibull](#) function;

## Examples

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

pp.gp.weibull(repairtimes, 1.566093, 0.355321, main = " ", line = TRUE)
```

---

pp.gumbel

*Probability versus Probability (PP) plot for the Gumbel distribution*

---

## Description

The function `pp.gumbel()` produces a PP plot for the Gumbel based on their MLE or any other estimate. Also, a reference line can be sketched.

## Usage

```
pp.gumbel(x, mu.est, sigma.est, main = " ", line = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>mu.est</code>	estimate of the parameter $\mu$
<code>sigma.est</code>	estimate of the parameter $\sigma$
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `pp.gumbel()` carries out a PP plot for the Gumbel.

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[qq.gumbel](#) for QQ plot and [ks.gumbel](#) function;

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

pp.gumbel(dataset2, 212.157, 151.768, main = " ", line = TRUE)
```

---

pp.inv.genexp	<i>Probability versus Probability (PP) plot for the Inverse Generalized Exponential(IGE) distribution</i>
---------------	---

---

**Description**

The function `pp.inv.genexp()` produces a PP plot for the Inverse Generalized Exponential(IGE) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.inv.genexp(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.inv.genexp()` carries out a PP plot for the Inverse Generalized Exponential(IGE).

## References

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*, Biometrical Journal, 43(1), 117-130.

Gupta, R.D. and Kundu, D., (2007). *Generalized exponential distribution: Existing results and some recent development*, Journal of Statistical Planning and Inference. 137, 3537-3547.

## See Also

[qq.inv.genexp](#) for QQ plot and [ks.inv.genexp](#) function;

## Examples

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

pp.inv.genexp(repairtimes, 1.097807, 1.206889, main = " ", line = TRUE)
```

---

pp.lfr	<i>Probability versus Probability (PP) plot for the linear failure rate(LFR) distribution</i>
--------	---

---

## Description

The function `pp.lfr()` produces a PP plot for the linear failure rate(LFR) based on their MLE or any other estimate. Also, a reference line can be sketched.

## Usage

```
pp.lfr(x, alpha.est, beta.est, main = " ", line = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>beta.est</code>	estimate of the parameter beta
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `pp.lfr()` carries out a PP plot for the linear failure rate(LFR).

## References

- Bain, L.J. (1974). *Analysis for the Linear Failure-Rate Life-Testing Distribution*, Technometrics, 16(4), 551 - 559.
- Lawless, J.F.(2003). *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York.
- Sen, A. and Bhattacharya, G.K.(1995). *Inference procedure for the linear failure rate mode*, Journal of Statistical Planning and Inference, 46, 59-76.

## See Also

[qq.lfr](#) for QQ plot and [ks.lfr](#) function;

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

pp.lfr(sys2, 1.777673e-03, 2.777640e-06, main = " ", line = TRUE)
```

---

pp.log.gamma	<i>Probability versus Probability (PP) plot for the log-gamma(LG) distribution</i>
--------------	--

---

## Description

The function `pp.log.gamma()` produces a PP plot for the log-gamma(LG) based on their MLE or any other estimate. Also, a reference line can be sketched.

## Usage

```
pp.log.gamma(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.log.gamma()` carries out a PP plot for the log-gamma(LG).

**References**

Klugman, S., Panjer, H. and Willmot, G. (2004). *Loss Models: From Data to Decisions*, 2nd ed., New York, Wiley.

Lawless, J. F., (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

**See Also**

`qq.log.gamma` for QQ plot and `ks.log.gamma` function;

**Examples**

```
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

pp.log.gamma(conductors, 0.0088741, 0.6059935, main = " ", line = TRUE)
```

---

pp.logis.exp	<i>Probability versus Probability (PP) plot for the Logistic-Exponential(LE) distribution</i>
--------------	---

---

**Description**

The function `pp.logis.exp()` produces a PP plot for the Logistic-Exponential(LE) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.logis.exp(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.logis.exp()` carries out a PP plot for the Logistic-Exponential(LE).

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[qq.logis.exp](#) for QQ plot and [ks.logis.exp](#) function;

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

pp.logis.exp(bearings, 2.36754, 0.01059, main = " ", line = TRUE)
```

---

<code>pp.logis.rayleigh</code>	<i>Probability versus Probability (PP) plot for the Logistic-Rayleigh(LR) distribution</i>
--------------------------------	--

---

**Description**

The function `pp.logis.rayleigh()` produces a PP plot for the Logistic-Rayleigh(LR) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.logis.rayleigh(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.logis.rayleigh()` carries out a PP plot for the Logistic-Rayleigh(LR).

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[qq.logis.rayleigh](#) for QQ plot and [ks.logis.rayleigh](#) function;

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

pp.logis.rayleigh(stress, 1.4779388, 0.2141343, main = " ", line = TRUE)
```

---

pp.loglog

*Probability versus Probability (PP) plot for the Loglog distribution*


---

**Description**

The function `pp.loglog()` produces a PP plot for the Loglog based on their MLE or any other estimator. Also, a reference line can be sketched.

**Usage**

```
pp.loglog(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.loglog()` carries out a PP plot for the Loglog.

## References

Pham, H.(2002). *A Vtub-Shaped Hazard Rate Function with Applications to System Safety*, International Journal of Reliability and Applications. ,Vol. 3, No. 1, pp. 1-16.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

## See Also

[qq.loglog](#) for QQ plot and [ks.loglog](#) function;

## Examples

```
## Load data sets.
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

pp.loglog(sys2, 0.9058689, 1.0028228, line = TRUE)
```

---

pp.moee

*Probability versus Probability (PP) plot for the Marshall-Olkin Extended Exponential(MOEE) distribution*

---

## Description

The function `pp.moee()` produces a PP plot for the MOEE based on their MLE or any other estimate. Also, a reference line can be sketched.

## Usage

```
pp.moee(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `pp.moee()` carries out a PP plot for the MOEE.

## References

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

## See Also

[qq.moew](#) for QQ plot and [ks.moew](#) functions

## Examples

```
## Load dataset
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576

pp.moew(stress, 75.67982, 1.67576, main = '', line = TRUE)
```

---

pp.moew

*Probability versus Probability (PP) plot for the Marshall-Olkin Extended Weibull(MOEW) distribution*

---

## Description

The function `pp.moew( )` produces a PP plot for the MOEW based on their MLE or any other estimate. Also, a reference line can be sketched.

## Usage

```
pp.moew(x, alpha.est, lambda.est, main = " ", line = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `pp.moew( )` carries out a PP plot for the MOEW.

**References**

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the Weibull and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

**See Also**

[qq.moew](#) for QQ plot and [ks.moew](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

pp.moew(sys2, 0.3035937, 279.2177754, main = " ", line = TRUE)
```

---

pp.weibull.ext	<i>Probability versus Probability (PP) plot for the Weibull Extension(WE) distribution</i>
----------------	--

---

**Description**

The function `pp.weibull.ext()` produces a PP plot for the Weibull Extension(WE) based on their MLE or any other estimate. Also, a reference line can be sketched.

**Usage**

```
pp.weibull.ext(x, alpha.est, beta.est, main = " ", line = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>beta.est</code>	estimate of the parameter beta
<code>main</code>	the title for the plot.
<code>line</code>	logical; if TRUE, a 45 degree line is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `pp.weibull.ext()` carries out a PP plot for the Weibull Extension(WE).

## References

Tang, Y., Xie, M. and Goh, T.N., (2003). *Statistical analysis of a Weibull extension model*, Communications in Statistics: Theory & Methods 32(5):913-928.

Zhang, T., and Xie, M.(2007). *Failure Data Analysis with Extended Weibull Distribution*, Communications in Statistics-Simulation and Computation, 36(3), 579-592.

## See Also

[qq.weibull.ext](#) for QQ plot and [ks.weibull.ext](#) function;

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

pp.weibull.ext(sys2, 0.00019114, 0.14696242, main = " ", line = TRUE)
```

---

qq.burrX

*Quantile versus quantile (QQ) plot for the BurrX distribution*

---

## Description

The function `qq.burrX()` produces a QQ plot for the BurrX based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.burrX(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.burrX()` carries out a QQ plot for the BurrX.

## References

- Kundu, D., and Raqab, M.Z. (2005). *Generalized Rayleigh Distribution: Different Methods of Estimation*, Computational Statistics and Data Analysis, 49, 187-200.
- Surles, J.G., and Padgett, W.J. (2005). *Some properties of a scaled Burr type X distribution*, Journal of Statistical Planning and Inference, 128, 271-280.
- Raqab, M.Z., and Kundu, D. (2006). *Burr Type X distribution: revisited*, Journal of Probability and Statistical Sciences, 4(2), 179-193.

## See Also

[pp.burrX](#) for PP plot and [ks.burrX](#) function

## Examples

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.1989515, lambda.est = 0.0130847

qq.burrX(bearings, 1.1989515, 0.0130847, main = " ", line.qt = FALSE)
```

---

qq.chen

*Quantile versus quantile (QQ) plot for the Chen distribution*

---

## Description

The function `qq.chen()` produces a QQ plot for the Chen based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.chen(x, beta.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>beta.est</code>	estimate of the parameter beta
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.chen()` carries out a QQ plot for the Chen

## References

Castillo, E., Hadi, A.S., Balakrishnan, N. and Sarabia, J.M.(2004). *Extreme Value and Related Models with Applications in Engineering and Science*, John Wiley and Sons, New York.

Chen, Z.(2000). *A new two-parameter lifetime distribution with bathtub shape or increasing failure rate function*, Statistics and Probability Letters, 49, 155-161.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

## See Also

[pp.chen](#) for PP plot and [ks.chen](#) function;

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of beta & lambda for the data(sys2)
## beta.est = 0.262282404, lambda.est = 0.007282371

qq.chen(sys2, 0.262282404, 0.007282371, line.qt = FALSE)
```

---

qq.exp.ext	<i>Quantile versus quantile (QQ) plot for the Exponential Extension(Ee) distribution</i>
------------	--

---

## Description

The function `qq.exp.ext()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.exp.ext(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.exp.ext()` carries out a QQ plot for the Exponential Extension.

## References

Nikulin, M. and Haghghi, F.(2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

## See Also

[pp.exp.ext](#) for PP plot and [ks.exp.ext](#) function;

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.0126e+01, lambda.est = 1.5848e-04

qq.exp.ext(sys2, 1.0126e+01, 1.5848e-04, main = " ", line.qt = FALSE)
```

---

qq.exp.power	<i>Quantile versus quantile (QQ) plot for the Exponential Power distribution</i>
--------------	--

---

## Description

The function `qq.exp.power()` produces a QQ plot for the Exponential Power distribution based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.exp.power(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.exp.power()` carries out a QQ plot for the Exponential Power distribution.

**References**

Castillo, E., Hadi, A.S., Balakrishnan, N. and Sarabia, J.M.(2004). *Extreme Value and Related Models with Applications in Engineering and Science*, John Wiley and Sons, New York.

Smith, R.M. and Bain, L.J.(1975). *An exponential power life-test distribution*, Communications in Statistics - Simulation and Computation, Vol.4(5), 469 - 481

**See Also**

[pp.exp.power](#) for PP plot and [ks.exp.power](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.905868898, lambda.est = 0.001531423

qq.exp.power(sys2, 0.905868898, 0.001531423, line.qt = FALSE)
```

---

qq.expo.logistic	<i>Quantile versus quantile (QQ) plot for the Exponentiated Logistic(EL) distribution</i>
------------------	---

---

**Description**

The function `qq.expo.logistic()` produces a QQ plot for the Exponentiated Logistic(EL) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.expo.logistic(x, alpha.est, beta.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>beta.est</code>	estimate of the parameter beta
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.expo.logistic()` carries out a QQ plot for the Exponentiated Logistic(EL).

**References**

Ali, M.M., Pal, M. and Woo, J. (2007). *Some Exponentiated Distributions*, The Korean Communications in Statistics, 14(1), 93-109.

Shirke, D.T., Kumbhar, R.R. and Kundu, D.(2005). *Tolerance intervals for exponentiated scale family of distributions*, Journal of Applied Statistics, 32, 1067-1074

**See Also**

[pp.expo.logistic](#) for PP plot and [ks.expo.logistic](#) function;

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(dataset2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 5.31302, beta.est = 139.04515

qq.expo.logistic(dataset2, 5.31302, 139.04515, main = " ", line.qt = FALSE)
```

---

qq.expo.weibull	<i>Quantile versus quantile (QQ) plot for the Exponentiated Weibull(EW) distribution</i>
-----------------	--

---

**Description**

The function `qq.expo.weibull()` produces a QQ plot for the Exponentiated Weibull(EW) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.expo.weibull(x, alpha.est, theta.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

x	vector of observations
alpha.est	estimate of the parameter alpha
theta.est	estimate of the parameter theta
main	the title for the plot
line.qt	logical; if TRUE, a line going by the first and third quartile is sketched.
...	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.expo.weibull()` carries out a QQ plot for the Exponentiated Weibull(EW).

## References

- Mudholkar, G.S. and Srivastava, D.K. (1993). *Exponentiated Weibull family for analyzing bathtub failure-rate data*, IEEE Transactions on Reliability, 42(2), 299-302.
- Murthy, D.N.P., Xie, M. and Jiang, R. (2003). *Weibull Models*, Wiley, New York.
- Nassar, M.M., and Eissa, F. H. (2003). *On the Exponentiated Weibull Distribution*, Communications in Statistics - Theory and Methods, 32(7), 1317-1336.

## See Also

[pp.expo.weibull](#) for PP plot and [ks.expo.weibull](#) function;

## Examples

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(stress)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est =1.026465, theta.est = 7.824943

qq.expo.weibull(stress, 1.026465, 7.824943, main = " ", line.qt = FALSE)
```

---

qq.flex.weibull	<i>Quantile versus quantile (QQ) plot for the flexible Weibull(FW) distribution</i>
-----------------	---

---

## Description

The function `qq.flex.weibull()` produces a QQ plot for the flexible Weibull(FW) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.flex.weibull(x, alpha.est, beta.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>beta.est</code>	estimate of the parameter beta
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.flex.weibull()` carries out a QQ plot for the flexible Weibull(FW).

**References**

Bebbington, M., Lai, C.D. and Zitikis, R. (2007). *A flexible Weibull extension*, Reliability Engineering and System Safety, 92, 719-726.

**See Also**

`pp.flex.weibull` for PP plot and `ks.flex.weibull` function;

**Examples**

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(repairtimes)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.07077507, beta.est = 1.13181535

qq.flex.weibull(repairtimes, 0.07077507, 1.13181535, main = " ", line.qt = FALSE)
```

---

qq.gen.exp	<i>Quantile versus quantile (QQ) plot for the Generalized Exponential(GE) distribution</i>
------------	--

---

**Description**

The function `qq.gen.exp()` produces a QQ plot for the GE based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.gen.exp(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.gen.exp()` carries out a QQ plot for the GE

**References**

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*. Biometrical Journal, 43(1), 117 - 130.

Gupta, R. D. and Kundu, D. (1999). *Generalized exponential distributions*. Australian and New Zealand Journal of Statistics, 41(2), 173 - 188.

**See Also**

[pp.gen.exp](#) for PP plot and [ks.gen.exp](#) function

**Examples**

```
## Load data
data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 5.28321139, lambda.est = 0.03229609

qq.gen.exp(bearings, 5.28321139, 0.03229609, line.qt = FALSE)
```

---

 qq.gompertz

---

*Quantile versus quantile (QQ) plot for the Gompertz distribution*


---

**Description**

The function `qq.gompertz()` produces a QQ plot for the Gompertz based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.gompertz(x, alpha.est, theta.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>theta.est</code>	estimate of the parameter theta
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.gompertz()` carries out a QQ plot for the Gompertz.

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[pp.gompertz](#) for PP plot and [ks.gompertz](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(sys2)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 0.00121307, theta.est = 0.00173329

qq.gompertz(sys2, 0.00121307, 0.00173329, main = " ", line.qt = FALSE)
```

---

qq.gp.weibull	<i>Quantile versus quantile (QQ) plot for the generalized power Weibull(GPW) distribution</i>
---------------	---

---

**Description**

The function `qq.gp.weibull()` produces a QQ plot for the generalized power Weibull(GPW) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.gp.weibull(x, alpha.est, theta.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>theta.est</code>	estimate of the parameter theta
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.gp.weibull()` carries out a QQ plot for the generalized power Weibull(GPW).

## References

Nikulin, M. and Haghghi, F.(2006). *A Chi-squared test for the generalized power Weibull family for the head-and-neck cancer censored data*, Journal of Mathematical Sciences, Vol. 133(3), 1333-1341.

Pham, H. and Lai, C.D.(2007). *On recent generalizations of the Weibull distribution*, IEEE Trans. on Reliability, Vol. 56(3), 454-458.

## See Also

[pp.gp.weibull](#) for PP plot and [ks.gp.weibull](#) function;

## Examples

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & theta for the data(repairtimes)
## Estimates of alpha & theta using 'maxLik' package
## alpha.est = 1.566093, theta.est = 0.355321

qq.gp.weibull(repairtimes, 1.566093, 0.355321, main = " ", line.qt = FALSE)
```

---

qq.gumbel

*Quantile versus quantile (QQ) plot for the Gumbel distribution*

---

## Description

The function `qq.gumbel()` produces a QQ plot for the Gumbel based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.gumbel(x, mu.est, sigma.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>mu.est</code>	estimate of the parameter $\mu$
<code>sigma.est</code>	estimate of the parameter $\sigma$
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.gumbel()` carries out a QQ plot for the Gumbel.

**References**

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*, Springer, New York.

**See Also**

[pp.gumbel](#) for PP plot and [ks.gumbel](#) function;

**Examples**

```
## Load data sets
data(dataset2)
## Maximum Likelihood(ML) Estimates of mu & sigma for the data(dataset2)
## Estimates of mu & sigma using 'maxLik' package
## mu.est = 212.157, sigma.est = 151.768

qq.gumbel(dataset2, 212.157, 151.768, main = " ", line.qt = FALSE)
```

---

qq.inv.genexp	<i>Quantile versus quantile (QQ) plot for the Inverse Generalized Exponential(IGE) distribution</i>
---------------	---

---

**Description**

The function `qq.inv.genexp()` produces a QQ plot for the Inverse Generalized Exponential(IGE) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.inv.genexp(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.inv.genexp()` carries out a QQ plot for the Exponential Extension.

## References

Gupta, R. D. and Kundu, D. (2001). *Exponentiated exponential family; an alternative to gamma and Weibull distributions*, Biometrical Journal, 43(1), 117-130.

Gupta, R.D. and Kundu, D., (2007). *Generalized exponential distribution: Existing results and some recent development*, Journal of Statistical Planning and Inference. 137, 3537-3547.

## See Also

[pp.inv.genexp](#) for PP plot and [ks.inv.genexp](#) function;

## Examples

```
## Load data sets
data(repairtimes)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(repairtimes)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.097807, lambda.est = 1.206889

qq.inv.genexp(repairtimes, 1.097807, 1.206889, main = " ", line.qt = FALSE)
```

---

qq.lfr	<i>Quantile versus quantile (QQ) plot for the linear failure rate(LFR) distribution</i>
--------	---

---

## Description

The function `qq.lfr()` produces a QQ plot for the linear failure rate(LFR) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.lfr(x, alpha.est, beta.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>beta.est</code>	estimate of the parameter beta
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.lfr()` carries out a QQ plot for the linear failure rate(LFR).

**References**

- Bain, L.J. (1974). *Analysis for the Linear Failure-Rate Life-Testing Distribution*, Technometrics, 16(4), 551 - 559.
- Lawless, J.F.(2003). *Statistical Models and Methods for Lifetime Data*, John Wiley and Sons, New York.
- Sen, A. and Bhattacharya, G.K.(1995). *Inference procedure for the linear failure rate mode*, Journal of Statistical Planning and Inference, 46, 59-76.

**See Also**

[pp.lfr](#) for PP plot and [ks.lfr](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 1.77773e-03, beta.est = 2.77764e-06

qq.lfr(sys2, 1.777673e-03, 2.777640e-06, main = " ", line.qt = FALSE)
```

---

qq.log.gamma	<i>Quantile versus quantile (QQ) plot for the log-gamma(LG) distribution</i>
--------------	--

---

**Description**

The function `qq.log.gamma()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.log.gamma(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.log.gamma()` carries out a QQ plot for the log-gamma(LG).

**References**

Klugman, S., Panjer, H. and Willmot, G. (2004). *Loss Models: From Data to Decisions*, 2nd ed., New York, Wiley.

Lawless, J. F., (2003). *Statistical Models and Methods for Lifetime Data*, 2nd ed., John Wiley and Sons, New York.

**See Also**

[pp.log.gamma](#) for PP plot and [ks.log.gamma](#) function;

**Examples**

```
## Load data sets
data(conductors)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(conductors)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 0.0088741, lambda.est = 0.6059935

qq.log.gamma(conductors, 0.0088741, 0.6059935, main = " ", line.qt = FALSE)
```

---

qq.logis.exp	<i>Quantile versus quantile (QQ) plot for the Logistic-Exponential(LE) distribution</i>
--------------	---

---

**Description**

The function `qq.logis.exp()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.logis.exp(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.logis.exp()` carries out a QQ plot for the Exponential Extension.

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[pp.logis.exp](#) for PP plot and [ks.logis.exp](#) function;

**Examples**

```
## Load data sets
data(bearings)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(bearings)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 2.36754, lambda.est = 0.01059

qq.logis.exp(bearings, 2.36754, 0.01059, main = " ", line.qt = FALSE)
```

---

<code>qq.logis.rayleigh</code>	<i>Quantile versus quantile (QQ) plot for the Logistic-Rayleigh(LR) distribution</i>
--------------------------------	--

---

**Description**

The function `qq.logis.rayleigh()` produces a QQ plot for the ExpExt based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.logis.rayleigh(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.logis.rayleigh()` carries out a QQ plot for the Exponential Extension.

**References**

Lan, Y. and Leemis, L. M. (2008). *The Logistic-Exponential Survival Distribution*, Naval Research Logistics, 55, 252-264.

**See Also**

[pp.logis.rayleigh](#) for PP plot and [ks.logis.rayleigh](#) function;

**Examples**

```
## Load data sets
data(stress)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 1.4779388, lambda.est = 0.2141343

qq.logis.rayleigh(stress, 1.4779388, 0.2141343, main = " ", line.qt = FALSE)
```

---

qq.loglog

---

*Quantile versus quantile (QQ) plot for the Loglog distribution*


---

**Description**

The function `qq.loglog()` produces a QQ plot for the Loglog based on their MLE or any other estimator. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.loglog(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.loglog()` carries out a QQ plot for the Loglog

## References

Pham, H.(2002). *A Vtub-Shaped Hazard Rate Function with Applications to System Safety*, International Journal of Reliability and Applications. ,Vol. 3, No. 1, pp. 1-16.

Pham, H.(2006). *System Software Reliability*, Springer-Verlag.

## See Also

[pp.loglog](#) for PP plot and [ks.loglog](#) function;

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.9058689 lambda.est = 1.0028228

qq.loglog(sys2, 0.9058689, 1.0028228, line.qt = FALSE)
```

---

qq.moee	<i>Quantile versus quantile (QQ) plot for the Marshall-Olkin Extended Exponential(MOEE) distribution</i>
---------	--

---

## Description

The function `qq.moee()` produces a QQ plot for the MOEE based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.moee(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.moee()` carries out a QQ plot for the MOEE.

## References

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the exponential and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

## See Also

[pp.moew](#) for PP plot and [ks.moew](#) function

## Examples

```
## Load dataset
data(stress)
## Estimates of alpha & lambda using 'maxLik' package
## alpha.est = 75.67982, lambda.est = 1.67576

qq.moew(stress, 75.67982, 1.67576, main = '', line.qt = FALSE)
```

---

qq.moew

*Quantile versus quantile (QQ) plot for the Marshall-Olkin Extended Weibull(MOEW) distribution*

---

## Description

The function `qq.moew( )` produces a QQ plot for the MOEW based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

## Usage

```
qq.moew(x, alpha.est, lambda.est, main = " ", line.qt = FALSE, ...)
```

## Arguments

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>lambda.est</code>	estimate of the parameter lambda
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

## Value

The function `qq.moew( )` carries out a QQ plot for the MOEW.

**References**

Marshall, A. W., Olkin, I. (1997). *A new method for adding a parameter to a family of distributions with application to the Weibull and Weibull families*. *Biometrika*,84(3):641-652.

Marshall, A. W., Olkin, I.(2007). *Life Distributions: Structure of Nonparametric, Semiparametric, and Parametric Families*. Springer, New York.

**See Also**

[pp.moew](#) for PP plot and [ks.moew](#) function;

**Examples**

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & lambda for the data(sys2)
## alpha.est = 0.3035937, lambda.est = 279.2177754

qq.moew(sys2, 0.3035937, 279.2177754, main = " ", line.qt = FALSE)
```

---

qq.weibull.ext	<i>Quantile versus quantile (QQ) plot for the Weibull Extension(WE) distribution</i>
----------------	--

---

**Description**

The function `qq.weibull.ext()` produces a QQ plot for the Weibull Extension(WE) based on their MLE or any other estimate. Also, a line going through the first and the third quartile can be sketched.

**Usage**

```
qq.weibull.ext(x, alpha.est, beta.est, main = " ", line.qt = FALSE, ...)
```

**Arguments**

<code>x</code>	vector of observations
<code>alpha.est</code>	estimate of the parameter alpha
<code>beta.est</code>	estimate of the parameter beta
<code>main</code>	the title for the plot
<code>line.qt</code>	logical; if TRUE, a line going by the first and third quartile is sketched.
<code>...</code>	additional arguments to be passed to the underlying plot function.

**Value**

The function `qq.weibull.ext()` carries out a QQ plot for the Weibull Extension(WE).

## References

Tang, Y., Xie, M. and Goh, T.N., (2003). *Statistical analysis of a Weibull extension model*, Communications in Statistics: Theory & Methods 32(5):913-928.

Zhang, T., and Xie, M.(2007). *Failure Data Analysis with Extended Weibull Distribution*, Communications in Statistics-Simulation and Computation, 36(3), 579-592.

## See Also

[pp.weibull.ext](#) for PP plot and [ks.weibull.ext](#) function;

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

qq.weibull.ext(sys2, 0.00019114, 0.14696242, main = " ", line.qt = FALSE)
```

---

reactorpump

*Reactor pump*

---

## Description

Several data sets related to life test are available in the `reliaR` package, which have been taken from the literature.

## Usage

```
data(reactorpump)
```

## Format

A vector containing 23 observations.

## Details

The data is based on total time on test plot analysis for mechanical components of the RSG-GAS reactor. The data are the time between failures of secondary reactor pumps.

## References

Bebbington, M., Lai, C.D. and Zitikis, R.(2007). *A flexible Weibull extension*. Reliability Engineering and System Safety, 92, 719-726.

Salman Suprawhardana M, Prayoto, Sangadji. *Total time on test plot analysis for mechanical components of the RSG-GAS reactor*. Atom Indones (1999), 25(2).

**Examples**

```
## Load data sets
data(reactorpump)
## Histogram for reactorpump
hist(reactorpump)
```

---

repairtimes

*Maintenance Data*

---

**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

```
data(repairtimes)
```

**Format**

A vector containing 46 observations.

**Details**

repairtimes correspond to maintenance data on active repair times (in hours) for an airborne communications transceiver.

**References**

Chhikara, R. S. and Folks, J. L. (1989). *The Inverse Gaussian Distribution*. Marcel Dekker, New York.

**Examples**

```
## Load data sets
data(repairtimes)
## Histogram for repairtimes
hist(repairtimes)
```

---

stress	<i>Breaking stress</i>
--------	------------------------

---

**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

```
data(stress)
```

**Format**

A vector containing 100 observations.

**Details**

The data is obtained from Nichols and Padgett (2006) and it represents the breaking stress of carbon fibres (in Gba).

**References**

Nichols, M.D. and Padgett, W.J. (2006). *A bootstrap control chart for Weibull percentiles*. Quality and Reliability Engineering International, 22, 141-151.

**Examples**

```
## Load data sets
data(stress)
## Histogram for stress
hist(stress)
```

---

sys2	<i>Software Reliability Dataset</i>
------	-------------------------------------

---

**Description**

Several data sets related to life test are available in the reliaR package, which have been taken from the literature.

**Usage**

```
data(sys2)
```

**Format**

A vector containing 86 observations.

**Details**

The data is obtained from DACS Software Reliability Dataset, Lyu (1996). The data represents the time-between-failures (time unit in miliseconds) of a software. The data given here is transformed from time-between-failures to failure times.

**References**

Lyu, M. R. (1996). *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, <http://www.cse.cuhk.edu.hk/~lyu/book/reliability/>

**Examples**

```
## Load data sets
data(sys2)
## Histogram for sys2
hist(sys2)
```

---

WeibullExt

*The Weibull Extension(WE) distribution*

---

**Description**

Density, distribution function, quantile function and random generation for the Weibull Extension(WE) distribution with shape parameter alpha and scale parameter beta.

**Usage**

```
dweibull.ext(x, alpha, beta, log = FALSE)
pweibull.ext(q, alpha, beta, lower.tail = TRUE, log.p = FALSE)
qweibull.ext(p, alpha, beta, lower.tail = TRUE, log.p = FALSE)
rweibull.ext(n, alpha, beta)
```

**Arguments**

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
alpha	shape parameter.
beta	scale parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

## Details

The Weibull Extension(WE) distribution has density

$$f(x; \alpha, \beta) = \beta \left(\frac{x}{\alpha}\right)^{\beta-1} \exp\left(\frac{x}{\alpha}\right)^{\beta} \exp\left\{-\alpha \left(\exp\left(\frac{x}{\alpha}\right)^{\beta} - 1\right)\right\}; (\alpha, c\beta) > 0, x > 0$$

where  $\alpha$  and  $\beta$  are the shape and scale parameters, respectively.

## Value

`dweibull.ext` gives the density, `pweibull.ext` gives the distribution function, `qweibull.ext` gives the quantile function, and `rweibull.ext` generates random deviates.

## References

Murthy, D.N.P., Xie, M. and Jiang, R. (2003). *Weibull Models*, Wiley, New York

Tang, Y., Xie, M. and Goh, T.N., (2003). *Statistical analysis of a Weibull extension model*, Communications in Statistics: Theory & Methods 32(5):913-928.

Xie, M., Tang, Y., Goh, T.N., (2002). *A modified Weibull extension with bathtub-shaped failure rate function*, Reliability Engineering System Safety 76(3):279-285.

Zhang, T., and Xie, M.(2007). *Failure Data Analysis with Extended Weibull Distribution*, Communications in Statistics-Simulation and Computation, 36(3), 579-592.

## See Also

[.Random.seed](#) about random number; [sweibull.ext](#) for Weibull Extension(WE) survival / hazard etc. functions

## Examples

```
## Load data sets
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242

dweibull.ext(sys2, 0.00019114, 0.14696242, log = FALSE)
pweibull.ext(sys2, 0.00019114, 0.14696242, lower.tail = TRUE, log.p = FALSE)
qweibull.ext(0.25, 0.00019114, 0.14696242, lower.tail=TRUE, log.p = FALSE)
rweibull.ext(30, 0.00019114, 0.14696242)
```

---

WeibullExtSurvival      *Survival related functions for the Weibull Extension(WE) distribution*

---

### Description

Conditional reliability function (crf), hazard function, hazard rate average (HRA) and survival function for the Weibull Extension(WE) distribution with shape parameter alpha and scale parameter beta.

### Usage

```
crf.weibull.ext(x, t = 0, alpha, beta)
hweibull.ext(x, alpha, beta)
hra.weibull.ext(x, alpha, beta)
sweibull.ext(x, alpha, beta)
```

### Arguments

x	vector of quantiles.
alpha	shape parameter.
beta	scale parameter.
t	age component.

### Value

crf.weibull.ext gives the conditional reliability function (crf), hweibull.ext gives the hazard function, hra.weibull.ext gives the hazard rate average (HRA) function, and sweibull.ext gives the survival function for the Weibull Extension(WE) distribution.

### References

Tang, Y., Xie, M. and Goh, T.N., (2003). *Statistical analysis of a Weibull extension model*, Communications in Statistics: Theory & Methods 32(5):913-928.

Zhang, T., and Xie, M.(2007). *Failure Data Analysis with Extended Weibull Distribution*, Communications in Statistics-Simulation and Computation, 36(3), 579-592.

### See Also

[dweibull.ext](#) for other c distribution related functions;

### Examples

```
## load data set
data(sys2)
## Maximum Likelihood(ML) Estimates of alpha & beta for the data(sys2)
## Estimates of alpha & beta using 'maxLik' package
## alpha.est = 0.00019114, beta.est = 0.14696242
```

```
## Reliability indicators for data(sys2):  
  
## Reliability function  
sweibull.ext(sys2, 0.00019114, 0.14696242)  
  
## Hazard function  
hweibull.ext(sys2, 0.00019114, 0.14696242)  
  
## hazard rate average(hra)  
hra.weibull.ext(sys2, 0.00019114, 0.14696242)  
  
## Conditional reliability function (age component=0)  
crf.weibull.ext(sys2, 0.00, 0.00019114, 0.14696242)  
  
## Conditional reliability function (age component=3.0)  
crf.weibull.ext(sys2, 3.0, 0.00019114, 0.14696242)
```

# Index

## \*Topic **datasets**

- bearings, [24](#)
- conductors, [31](#)
- dataset2, [32](#)
- reactorpump, [141](#)
- repairtimes, [142](#)
- stress, [143](#)
- sys2, [143](#)

## \*Topic **distribution**

- BurrX, [25](#)
- Chen, [28](#)
- ExpExt, [34](#)
- ExpoLogistic, [37](#)
- ExpoWeibull, [39](#)
- ExpPower, [42](#)
- FlexWeibull, [43](#)
- GenExp, [46](#)
- Gompertz, [49](#)
- GPWeibull, [51](#)
- Gumbel, [54](#)
- InvGenExp, [57](#)
- LFR, [83](#)
- Loggamma, [86](#)
- LogisExp, [88](#)
- LogisRayleigh, [91](#)
- Loglog, [93](#)
- MOEE, [96](#)
- MOEW, [99](#)
- WeibullExt, [144](#)

## \*Topic **hplot**

- pp.burrX, [101](#)
- pp.chen, [102](#)
- pp.exp.ext, [103](#)
- pp.exp.power, [104](#)
- pp.expo.logistic, [105](#)
- pp.expo.weibull, [106](#)
- pp.flex.weibull, [107](#)
- pp.gen.exp, [108](#)
- pp.gompertz, [109](#)

- pp.gp.weibull, [110](#)
- pp.gumbel, [111](#)
- pp.inv.genexp, [112](#)
- pp.lfr, [113](#)
- pp.log.gamma, [114](#)
- pp.logis.exp, [115](#)
- pp.logis.rayleigh, [116](#)
- pp.loglog, [117](#)
- pp.moee, [118](#)
- pp.moew, [119](#)
- pp.weibull.ext, [120](#)
- qq.burrX, [121](#)
- qq.chen, [122](#)
- qq.exp.ext, [123](#)
- qq.exp.power, [124](#)
- qq.expo.logistic, [125](#)
- qq.expo.weibull, [126](#)
- qq.flex.weibull, [127](#)
- qq.gen.exp, [128](#)
- qq.gompertz, [129](#)
- qq.gp.weibull, [130](#)
- qq.gumbel, [131](#)
- qq.inv.genexp, [132](#)
- qq.lfr, [133](#)
- qq.log.gamma, [134](#)
- qq.logis.exp, [135](#)
- qq.logis.rayleigh, [136](#)
- qq.loglog, [137](#)
- qq.moee, [138](#)
- qq.moew, [139](#)
- qq.weibull.ext, [140](#)

## \*Topic **htest**

- ks.burrX, [59](#)
- ks.chen, [61](#)
- ks.exp.ext, [62](#)
- ks.exp.power, [63](#)
- ks.expo.logistic, [64](#)
- ks.expo.weibull, [65](#)
- ks.flex.weibull, [67](#)

- ks.gen.exp, 68
- ks.gompertz, 69
- ks.gp.weibull, 70
- ks.gumbel, 71
- ks.inv.genexp, 72
- ks.lfr, 74
- ks.log.gamma, 75
- ks.logis.exp, 76
- ks.logis.rayleigh, 77
- ks.loglog, 78
- ks.moe, 79
- ks.moew, 81
- ks.weibull.ext, 82
- \*Topic **models**
  - abic.burrX, 4
  - abic.chen, 5
  - abic.exp.ext, 6
  - abic.exp.power, 7
  - abic.expo.logistic, 8
  - abic.expo.weibull, 9
  - abic.flex.weibull, 10
  - abic.gen.exp, 11
  - abic.gompertz, 12
  - abic.gp.weibull, 13
  - abic.gumbel, 14
  - abic.inv.genexp, 15
  - abic.lfr, 16
  - abic.log.gamma, 17
  - abic.logis.exp, 18
  - abic.logis.rayleigh, 19
  - abic.loglog, 20
  - abic.moe, 21
  - abic.moew, 22
  - abic.weibull.ext, 23
- \*Topic **survival**
  - BurrXsurvival, 26
  - Chensurvival, 30
  - EPsurvival, 33
  - ExpExtSurvival, 35
  - ExpoLogisticsSurvival, 38
  - ExpoWeibullSurvival, 41
  - FlexWeibullSurvival, 45
  - GenExpsurvival, 47
  - GompertzSurvival, 50
  - GPWeibullSurvival, 53
  - GumbelSurvival, 55
  - InvGenExpESurvival, 58
  - LFRSurvival, 84
  - LoggammaSurvival, 87
  - LogisExpsurvival, 90
  - LogisRayleighSurvival, 92
  - LoglogSurvival, 95
  - MOESurvival, 97
  - MOEWSurvival, 100
  - WeibullExtSurvival, 146
- .Random.seed, 26, 29, 35, 38, 40, 43, 44, 47, 50, 52, 55, 58, 84, 87, 89, 92, 94, 97, 100, 145
- abic.burrX, 4
- abic.chen, 5
- abic.exp.ext, 6
- abic.exp.power, 7
- abic.expo.logistic, 8
- abic.expo.weibull, 9
- abic.flex.weibull, 10
- abic.gen.exp, 11
- abic.gompertz, 12
- abic.gp.weibull, 13
- abic.gumbel, 14
- abic.inv.genexp, 15
- abic.lfr, 16
- abic.log.gamma, 17
- abic.logis.exp, 18
- abic.logis.rayleigh, 19
- abic.loglog, 20
- abic.moe, 21
- abic.moew, 22
- abic.weibull.ext, 23
- bearings, 24
- BurrX, 25
- BurrXsurvival, 26
- Chen, 28
- Chensurvival, 30
- conductors, 31
- crf.burrX (BurrXsurvival), 26
- crf.chen (Chensurvival), 30
- crf.exp.ext (ExpExtSurvival), 35
- crf.exp.power (EPsurvival), 33
- crf.expo.logistic (ExpoLogisticsSurvival), 38
- crf.expo.weibull (ExpoWeibullSurvival), 41
- crf.flex.weibull (FlexWeibullSurvival), 45

- crf.gen.exp (GenExpsurvival), 47
- crf.gompertz (Gompertzsurvival), 50
- crf.gp.weibull (GPWeibullsurvival), 53
- crf.gumbel (Gumbelurvival), 55
- crf.inv.genexp (InvGenExpEsurvival), 58
- crf.lfr (LFRsurvival), 84
- crf.log.gamma (Loggamasurvival), 87
- crf.logis.exp (LogisExpsurvival), 90
- crf.logis.rayleigh  
(LogisRayleighsurvival), 92
- crf.loglog (Loglogsurvival), 95
- crf.moe (MOEsurvival), 97
- crf.moew (MOEWsurvival), 100
- crf.weibull.ext (WeibullExtsurvival),  
146
  
- dataset2, 32
- dburrX, 28
- dburrX (BurrX), 25
- dchen, 31
- dchen (Chen), 28
- dexp.ext, 36
- dexp.ext (ExpExt), 34
- dexp.power, 33
- dexp.power (ExpPower), 42
- dexpo.logistic, 39
- dexpo.logistic (ExpoLogistic), 37
- dexpo.weibull, 41
- dexpo.weibull (ExpoWeibull), 39
- dflex.weibull, 45
- dflex.weibull (FlexWeibull), 43
- dgen.exp, 48
- dgen.exp (GenExp), 46
- dgompertz, 51
- dgompertz (Gompertz), 49
- dgp.weibull, 53
- dgp.weibull (GPWeibull), 51
- dgumbel, 56
- dgumbel (Gumbel), 54
- dinv.genexp, 59
- dinv.genexp (InvGenExp), 57
- dlfr, 85
- dlfr (LFR), 83
- dlog.gamma, 88
- dlog.gamma (Loggamma), 86
- dlogis.exp, 90
- dlogis.exp (LogisExp), 88
- dlogis.rayleigh, 93
- dlogis.rayleigh (LogisRayleigh), 91
  
- dloglog, 95
- dloglog (Loglog), 93
- dmoe, 98
- dmoe (MOEE), 96
- dmoe, 101
- dmoe (MOEW), 99
- dweibull.ext, 146
- dweibull.ext (WeibullExt), 144
  
- EPsurvival, 33
- ExpExt, 34
- ExpExtsurvival, 35
- ExpoLogistic, 37
- ExpoLogisticsurvival, 38
- ExpoWeibull, 39
- ExpoWeibullsurvival, 41
- ExpPower, 42
  
- FlexWeibull, 43
- FlexWeibullsurvival, 45
  
- GenExp, 46
- GenExpsurvival, 47
- Gompertz, 49
- Gompertzsurvival, 50
- GPWeibull, 51
- GPWeibullsurvival, 53
- Gumbel, 54
- Gumbelurvival, 55
  
- hburrX (BurrXsurvival), 26
- hchen (Chensurvival), 30
- hexp.ext (ExpExtsurvival), 35
- hexp.power (EPsurvival), 33
- hexpo.logistic (ExpoLogisticsurvival),  
38
- hexpo.weibull (ExpoWeibullsurvival), 41
- hflex.weibull (FlexWeibullsurvival), 45
- hgen.exp (GenExpsurvival), 47
- hgompertz (Gompertzsurvival), 50
- hgp.weibull (GPWeibullsurvival), 53
- hgumbel (Gumbelurvival), 55
- hinv.genexp (InvGenExpEsurvival), 58
- hlfr (LFRsurvival), 84
- hlog.gamma (Loggamasurvival), 87
- hlogis.exp (LogisExpsurvival), 90
- hlogis.rayleigh  
(LogisRayleighsurvival), 92
- hloglog (Loglogsurvival), 95

- hmoe (MOEESurvival), 97  
 hmoew (MOEWSurvival), 100  
 hra.burrX (BurrXSurvival), 26  
 hra.chen (ChenSurvival), 30  
 hra.exp.ext (ExpExtSurvival), 35  
 hra.exp.power (ExpPowerSurvival), 33  
 hra.expo.logistic  
     (ExpoLogisticSurvival), 38  
 hra.expo.weibull (ExpoWeibullSurvival),  
     41  
 hra.flex.weibull (FlexWeibullSurvival),  
     45  
 hra.gen.exp (GenExpSurvival), 47  
 hra.gompertz (GompertzSurvival), 50  
 hra.gp.weibull (GPWeibullSurvival), 53  
 hra.gumbel (GumbelSurvival), 55  
 hra.inv.genexp (InvGenExpSurvival), 58  
 hra.lfr (LFRSurvival), 84  
 hra.log.gamma (LogGammaSurvival), 87  
 hra.logis.exp (LogisExpSurvival), 90  
 hra.logis.rayleigh  
     (LogisRayleighSurvival), 92  
 hra.loglog (LogLogSurvival), 95  
 hra.moe (MOEESurvival), 97  
 hra.moew (MOEWSurvival), 100  
 hra.weibull.ext (WeibullExtSurvival),  
     146  
 hweibull.ext (WeibullExtSurvival), 146  
  
 InvGenExp, 57  
 InvGenExpSurvival, 58  
 InvGenExpSurvival (InvGenExpSurvival),  
     58  
  
 ks.burrX, 59, 102, 122  
 ks.chen, 61, 103, 123  
 ks.exp.ext, 62, 104, 124  
 ks.exp.power, 63, 105, 125  
 ks.expo.logistic, 64, 106, 126  
 ks.expo.weibull, 65, 107, 127  
 ks.flex.weibull, 67, 108, 128  
 ks.gen.exp, 68, 109, 129  
 ks.gompertz, 69, 110, 130  
 ks.gp.weibull, 70, 111, 131  
 ks.gumbel, 71, 112, 132  
 ks.inv.genexp, 72, 113, 133  
 ks.lfr, 74, 114, 134  
 ks.log.gamma, 75, 115, 135  
 ks.logis.exp, 76, 116, 136  
  
 ks.logis.rayleigh, 77, 117, 137  
 ks.loglog, 21, 78, 118, 138  
 ks.moe, 79, 119, 139  
 ks.moew, 81, 120, 140  
 ks.weibull.ext, 82, 121, 141  
  
 LFR, 83  
 LFRSurvival, 84  
 Loggamma, 86  
 LogGammaSurvival, 87  
 LogisExp, 88  
 LogisExpSurvival, 90  
 LogisRayleigh, 91  
 LogisRayleighSurvival, 92  
 Loglog, 93  
 LogLogSurvival, 95  
  
 MOEE, 96  
 MOEESurvival, 97  
 MOEW, 99  
 MOEWSurvival, 100  
  
 pburrX (BurrX), 25  
 pchen (Chen), 28  
 pexp.ext (ExpExt), 34  
 pexp.power (ExpPower), 42  
 pexpo.logistic (ExpoLogistic), 37  
 pexpo.weibull (ExpoWeibull), 39  
 pflex.weibull (FlexWeibull), 43  
 pgen.exp (GenExp), 46  
 pgompertz (Gompertz), 49  
 pgp.weibull (GPWeibull), 51  
 pgumbel (Gumbel), 54  
 pinv.genexp (InvGenExp), 57  
 plfr (LFR), 83  
 plog.gamma (LogGamma), 86  
 plogis.exp (LogisExp), 88  
 plogis.rayleigh (LogisRayleigh), 91  
 ploglog (Loglog), 93  
 pmoe (MOEE), 96  
 pmoew (MOEW), 99  
 pp.burrX, 5, 60, 101, 122  
 pp.chen, 6, 61, 102, 123  
 pp.exp.ext, 7, 63, 103, 124  
 pp.exp.power, 8, 64, 104, 125  
 pp.expo.logistic, 9, 65, 105, 126  
 pp.expo.weibull, 10, 66, 106, 127  
 pp.flex.weibull, 11, 67, 107, 128  
 pp.gen.exp, 12, 69, 108, 129

- pp.gompertz, [13](#), [70](#), [109](#), [130](#)
- pp.gp.weibull, [14](#), [71](#), [110](#), [131](#)
- pp.gumbel, [15](#), [72](#), [111](#), [132](#)
- pp.inv.genexp, [16](#), [73](#), [112](#), [133](#)
- pp.lfr, [17](#), [74](#), [113](#), [134](#)
- pp.log.gamma, [18](#), [76](#), [114](#), [135](#)
- pp.logis.exp, [19](#), [77](#), [115](#), [136](#)
- pp.logis.rayleigh, [20](#), [78](#), [116](#), [137](#)
- pp.loglog, [79](#), [117](#), [138](#)
- pp.moee, [22](#), [80](#), [118](#), [139](#)
- pp.moew, [23](#), [81](#), [119](#), [140](#)
- pp.weibull.ext, [24](#), [83](#), [120](#), [141](#)
- pweibull.ext (WeibullExt), [144](#)
  
- qburrX (BurrX), [25](#)
- qchen (Chen), [28](#)
- qexp.ext (ExpExt), [34](#)
- qexp.power (ExpPower), [42](#)
- qexpo.logistic (ExpoLogistic), [37](#)
- qexpo.weibull (ExpoWeibull), [39](#)
- qflex.weibull (FlexWeibull), [43](#)
- qgen.exp (GenExp), [46](#)
- qgompertz (Gompertz), [49](#)
- qgp.weibull (GPWeibull), [51](#)
- qgumbel (Gumbel), [54](#)
- qinv.genexp (InvGenExp), [57](#)
- qlfr (LFR), [83](#)
- qlog.gamma (Loggamma), [86](#)
- qlogis.exp (LogisExp), [88](#)
- qlogis.rayleigh (LogisRayleigh), [91](#)
- qloglog (Loglog), [93](#)
- qmoee (MOEE), [96](#)
- qmoew (MOEW), [99](#)
- qq.burrX, [5](#), [60](#), [102](#), [121](#)
- qq.chen, [6](#), [61](#), [103](#), [122](#)
- qq.exp.ext, [7](#), [63](#), [104](#), [123](#)
- qq.exp.power, [8](#), [64](#), [105](#), [124](#)
- qq.expo.logistic, [9](#), [65](#), [106](#), [125](#)
- qq.expo.weibull, [10](#), [66](#), [107](#), [126](#)
- qq.flex.weibull, [11](#), [67](#), [108](#), [127](#)
- qq.gen.exp, [12](#), [69](#), [109](#), [128](#)
- qq.gompertz, [13](#), [70](#), [110](#), [129](#)
- qq.gp.weibull, [14](#), [71](#), [111](#), [130](#)
- qq.gumbel, [15](#), [72](#), [112](#), [131](#)
- qq.inv.genexp, [16](#), [73](#), [113](#), [132](#)
- qq.lfr, [17](#), [74](#), [114](#), [133](#)
- qq.log.gamma, [18](#), [76](#), [115](#), [134](#)
- qq.logis.exp, [19](#), [77](#), [116](#), [135](#)
- qq.logis.rayleigh, [20](#), [78](#), [117](#), [136](#)
  
- qq.loglog, [21](#), [79](#), [118](#), [137](#)
- qq.moee, [22](#), [80](#), [119](#), [138](#)
- qq.moew, [23](#), [81](#), [120](#), [139](#)
- qq.weibull.ext, [24](#), [83](#), [121](#), [140](#)
- qweibull.ext (WeibullExt), [144](#)
  
- rburrX (BurrX), [25](#)
- rchen (Chen), [28](#)
- reactorpump, [141](#)
- repairtimes, [142](#)
- rexp.ext (ExpExt), [34](#)
- rexp.power (ExpPower), [42](#)
- rexp.logistic (ExpoLogistic), [37](#)
- rexp.weibull (ExpoWeibull), [39](#)
- rflex.weibull (FlexWeibull), [43](#)
- rgen.exp (GenExp), [46](#)
- rgompertz (Gompertz), [49](#)
- rgp.weibull (GPWeibull), [51](#)
- rgumbel (Gumbel), [54](#)
- rinv.genexp (InvGenExp), [57](#)
- rlfr (LFR), [83](#)
- rlog.gamma (Loggamma), [86](#)
- rlogis.exp (LogisExp), [88](#)
- rlogis.rayleigh (LogisRayleigh), [91](#)
- rloglog (Loglog), [93](#)
- rmoee (MOEE), [96](#)
- rmoew (MOEW), [99](#)
- rweibull.ext (WeibullExt), [144](#)
  
- sburrX, [26](#)
- sburrX (BurrXsurvival), [26](#)
- schen, [29](#)
- schen (Chensurvival), [30](#)
- sexp.ext, [35](#)
- sexp.ext (ExpExtsurvival), [35](#)
- sexp.power, [43](#)
- sexp.power (EPsurvival), [33](#)
- sexpo.logistic, [38](#)
- sexpo.logistic (ExpoLogisticsurvival), [38](#)
- sexpo.weibull, [40](#)
- sexpo.weibull (ExpoWeibullsurvival), [41](#)
- sflex.weibull, [44](#)
- sflex.weibull (FlexWeibullsurvival), [45](#)
- sgen.exp, [47](#)
- sgen.exp (GenExpsurvival), [47](#)
- sgompertz, [50](#)
- sgompertz (Gompertzsurvival), [50](#)
- sgp.weibull, [52](#)

sgp.weibull (GPWeibullsurvival), 53  
sgumbel, 55  
sgumbel (GumbelSurvival), 55  
sinv.genexp, 58  
sinv.genexp (InvGenExpESurvival), 58  
slfr, 84  
slfr (LFRSurvival), 84  
slog.gamma, 87  
slog.gamma (LogGammaSurvival), 87  
slogis.exp, 89  
slogis.exp (LogisExpSurvival), 90  
slogis.rayleigh, 92  
slogis.rayleigh  
    (LogisRayleighSurvival), 92  
sloglog, 94  
sloglog (LogLogSurvival), 95  
smoe, 97  
smoe (MOEESurvival), 97  
smoew, 100  
smoew (MOEWSurvival), 100  
stress, 143  
sweibull.ext, 145  
sweibull.ext (WeibullExtSurvival), 146  
sys2, 143  
  
WeibullExt, 144  
WeibullExtSurvival, 146